Linear model selection and regularization

Zhongling Jinag, Xinyu Zhang

October 28, 2016

Abstract

This report is about reporducing linear regression based on four linear models in order to improve the simple linear model by replcing plain lesat squares fitting with some alternative fitting procedures. All data and sources are from Chapter 6 *Linear model selection and regularization* (pages 203-264), from the book "An Introduction to Statistical Learning" (by James et al)

Introduction

In this report, in order to yield better prediction accuracy and model interpretability, we want to explore-approaches for extending the linear model framework. Hence, we use four approaches for automatically performing feature selection or vairable selection–that is, for excluding irrelevant wariables from a multiple regression model:

- Ridge regression (RR)
- Lasso regression (LR)
- Principal Components regression (PCR)
- Partial Least Squares regression (PLSR)

Data

The data set is a CSV file avaiable in http://www-bcf.usc.edu/~gareth/ISL/Credit.csv. It contains `balance` (average credit card debt for a number of individuals) as well as 10 quantitative predictors: `age`, `cards`(number of credit cards), `rating`(credit rating), `education` (year of education), `income`(in thousands of dollars), `limit`(credit limit), `gender`, `student`(student status), `status`(marital status), and `ethnicity`(Caucasian, African American or Asian).

The `Credit` data set has 400 observations. Based on this data set, we transform it into another data set into `scaled_credit.csv` in `data/` folder (We will explain in Analysis section). In `scaled_credit` data set, there are a random sample of size 300 with no replacement in `train` set and the rest of 100 values in `test` set. For reproducibility purposes, we set a random seed `set.seed(200)`. Then we will use train and test vectors (in scaled data), to build models.

Methods

Ordinary Least Square Method

In statistics, ordinary least squares (OLS) or linear least squares is a method for estimating the unknown parameters in a linear regression model, with the goal of minimizing the sum of the squares of the differences between the observed responses in the given dataset and those predicted by a linear function of a set of explanatory variables (visually this is seen as the sum of the vertical distances between each data point in the set and the corresponding point on the regression line – the smaller the differences, the better the model fits the data).

$$RSS = \sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} (\hat{\beta}_j x_{ij}) \right)^2$$

or equivalently as

$$RSS = (y_1 - \hat{\beta}_0 - \sum_{j=1}^{p} (\hat{\beta}_j x_{1j}))^2 + (y_2 - \hat{\beta}_0 - \sum_{j=1}^{p} (\hat{\beta}_j x_{2j}))^2 + ... + (y_n - \hat{\beta}_0 - \sum_{j=1}^{p} (\hat{\beta}_j x_{nj}))^2$$

Shrinkage methods

This approach involves fitting a model involving all `p` predictors. However, the estimated coefficients are shrunken towards zero relative to the least squares estimates. This shrinkage (also known as *regularization*) has the effect of reducing variance. Depending on what type of shrinkage is performed, some of the coefficients may be estimated to be exactly zero. Hence, shrinkage methods can also perform variable selection. It may not be immediately obvious why such a constraint should improve the fit, but it turns out that shrinking the coefficient estimates can significantly reduce their variance. The two best-known techniques for shrinking the regression coefficients towards zero are *ridge regression* the the *lasso*.

Ridge Regression

Ridge regression is very similar to least squares, except that the coefficients are estimated by minimizing a slightly different quantity. In particular, the ridge regression coefficinet estiamtes $\hat{\beta}^R$ are the values that minimize

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

where $\lambda \geq 0$ is a *tunig parameter*, to be determined separately. Equation trades off two different criteria. as with least squares, ridge regression seeks coefficinet estimates that fit the data well, by making the RSS small. however, the second term, $\lambda \sum_{j=1}^{p} \beta_j^2$, called a *shrinkage penalty*, is small when $\beta_1, ..., \beta_p$ are close to zero, and so it has the effect of *shrinking* the estimates of β_j towards zero. The tunning parameter λ serves to control the relative impact of these two terms on the regression coefficient estiamtes.

Lasso Regression

Ridge regression does have one obvious disadvantage. It includes all p predictors in the final model. The penalty $\lambda \sum \beta_j^2$ will shrink all of the coefficients towards zero. but it will not set any of them exactly to zero (unless $\lambda = \infty$). It can create a challenge in model interpretation in settings in which the number of vairables p is quite large. The *lasso* is a relatively recent alternative to ridge regression that overcomes this disadvantage. The lasso coefficients, $\hat{\beta_\lambda^L}$, minimize the quantity

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

The lasso shrinks the coefficient estiamtes towards zero. However, in the case of the lasso, the l_1 penalty (In statistical parlance, the lasso uses l_1 penalty instead of an l_2 penalty. The l_1 nrom of a coefficient vector β is given by $||\beta||_1 = \sum |\beta_j|$) has the effect of forcing some of the coefficient estiamtes to be exactly equal to zero when the tuning parameter λ is sufficiently large. Hence, much like best subset selection, the lasso performs *variable selection.* As a result, models generated from the lasso are generally much easier to interpret than those produced by ridge regression. We say that the lasso yields *sparse* models – that is, models that involve only a subset of the vairbales.

Dimension Reduction Method

This approach involves *prejecting* the p predictors into a M-dimensional subsapce, where M < p. This is achieved by computing M different *linear combinations*, or *projections*, of the variables. Then these M projections are used as predictors to fit a linear regression model by least squares.

We now explore a class of approaches that *transform* the predictors and then fit a least squares model using the transformed variabels. We will refer to these techniques as *dimension reduction methods.*

Let $Z_1, Z_2, ..., Z_M$ represent $M < p$ linear combinations of our original p predictors. That is,

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j$$

for some constants $\phi_{1m}, \phi_{2m}, .., \phi_{pm}, m = 1, .., M$. We can then fit the linear regression model using least squares.

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \epsilon_i, i = 1, ..., n$$

All dimension reduction methods work in two steps. First, the transformed predictors $Z_1, Z_2, ..., Z_M$ are obtained. Second, the model is fit using these M predictors. however, the choice of $Z_1, Z_2, ..., Z_M$, or equivalently, the selection of the ϕ_{jm}'s, can be achieved in different ways. So we will consider two approaches for this talk: *principal componets* and *partial least squares.*

Principal Components Regression (PCR)

Principal components analysis (PCA) is a popular approach for deriving a low-dimensional set of features from a large set of variables. It's a dimension reduction technique for regression of a $n \times p$ data matrix X. The *first principal component* direction of the data is that along which the observations *vary the most.*

The first principal component vector defines the line taht is *as close as possible* to the data. The first principal component line minimizes the sum of the squared perpendicular distances between each point and the line.

In general, one can construct up to p distinct principal components. The second principal component Z_2 is linear combination of the variables that is uncorrelated with Z_1, and has largest variance subject to this constraint. With more predictors, then more additional componetns could be constructed. They would successively maximize variance, subject to the constraint of being uncorrelated with the preceding components.

PCR approach involves constructing the first M principal components, $Z_1, Z_2, ..., Z_M$, and then using these components as the predictors in a linear regression model that is fit using least squares. The key idea is that often a small number of principal components suffice to explain most of the vairability in the data, as well as the relationship with the response. In other words, we assuem that *the directions in which* $X_1, ..., X_p$ *show the most variation are the directions that are associated with* Y. While this assumption is not guaranteed to be true, it often turns out to be a reasonable enough approximation to give good results.

Even though PCR provides a simple way to perform regression using $M < p$ predictors, it is not a feature selection method. This is because each of the M principal components used in the regression is a linear combination of all p of the original features. In PCR, the number of principal components, M, is typically chosen by cross-validation.

Partial Least Square Regression (PLSR)

We now present partial least squares (PLS), a *supervised* alternative to PCR. Like PCR, PLS is a dimension reduction method, which first identifies a new set of features $Z_1, ..., Z_M$ that are linear combinations of the original features, and then fits a linear model via least squares using these M new features. But unlike PCR, PLS identifies these new features in a supervised way–that is, it makes use of the response `Y` in order to identify new features that not only approximate the old features well, but also that are related to the response. Roughly speaking, the PLS approach attempts to find directions that help explain both the response and the predictors.

Compute the first PLS direction: after standardizing the p predictors, PLS computes the first direction Z_1 by setting each ϕ_{j1} in equal to the coefficient from the simple linear regression of Y onto X_j. One can show that this coefficient is proportional to the correlation between Y and X_j. Hence, in computing $Z_1 = \sum_{j=1}^{p} \phi_{j1} X_j$, PLS places the highest weight on the vairables that are most strongly related to the response.

Compute the second PLS direction: we first *adjust* each of the variables for Z_1, by regressing each variable on Z_1 and taking *residuals.* These residuals can be interpreted as the remaining information that has not been explained by the first PLS direction. We then compute Z_2 using this *orthogonalized* data in exactly the same fashion as Z_1 was computed based on the original data. This iterative approach can be repoeated M times to identify multiple PLS cmoponents $Z_1, ..., Z_M$. Finally, at the end of this procedure, we use least squares to fit a linear model to predict Y using $Z_1, ..., Z_M$ in exactly the same fashion as for PCR.

PLS is popular in the field of chemometrics, where many variables arise from digitized spectrometry signals. In practice it often performs no better than ridge regression or PCR. While the supervised dimension reduction of PLS can reduce bias, it also has the potential to increase variance, so that the overall benefit of PLS relative to PCR is a wash.

Analysis

Exploratory Data Analysis (EDA)

In this section, the first step is to understand the data. The `Credit.csv` data we have quantitative variables and qualitative (categorical) variables. We obtain descriptive statistics and summaries of all variables and get statistics information. You can find summary output `eda-output.txt` in `data/` folder, and plots in `images/` folder.

For the quantitative variables, there are `balance`, `age`, `cards`, `rating`, `education` , `income`, `limit`. We compute:

- Minimum, Maximum, Range
- Median, First and Third quartiles, and interquartile range (IQR)
- Mean and Standard Deviation
- histograms and boxplots

For the qualitative (categorical) variables, there are `gender`, `student`, `status`, and `ethnicity`. We compute

- a table of frequencies with both the frequency and the relative frequency
- barcharts of such frequencies

Because we are interested in studying the association between `Balance` and the rest of predictors, we also obtain:

- matrix of correlations among all quantitative variables.

- scatterplot matrix.
- anova's between `Balance` and all the qualitative variables
- conditional boxplots between `Balance` and the qualitative variables, that is, boxplots of `Balance` conditioned to each of `Gender`, `Ethnicity`, `Student`, and `Married`.

Pre-modeling Data Processing

There are two major processing steps are needed before we fit any model:

- Convert factors into dummy variables
- Mean centering and standardization

The first processing step involves transforming each categorical variable (`Gender`, `Student`, `Married`, and `Ethnicity`) into dummy variables (binary indicators).For a given factor, the number of binary indicators will be one less than the number of levels. The main reason to do this is because both ridge and lasso regressions will not work if the input data contains factors.

The second processing step involves mean-centering and standardizing all the variables. This means that each variable will have mean zero, and standard deviation one. One reason to standardize variables is to have comparable scales. In a regression analysis, the value of the computed coefficients will depend on the measurement scale of the associated predictors. A β coefficient will be different if the variable is measured in grams or in kilos. To avoid favoring a certain coefficient, it is recommended to mean-center and standardize the variables.

> For the scaled data, it's saved in `scaled_credit.csv` in `data/` folder. And it is the actual data that we will use for the model building process.

Regression Models

As we mentioned in Data section, The `scaled_credit` data set has 400 observations which has a random sample of size 300 with no replacement in `training` set and the rest of 100 values in `testing` set. For reproducibility purposes, we set a random seed `set.seed(200)`. Then we will use train and test vectors to build 5 models: - Ordinary Least Squares (OLS) - Ridge Regression - Lasso Regression - Principal Components regression (PCR) - Partial Least Squares regression (PLSR) The multiple linear regression model via OLS serves as a benchmark to other 4 models. Ridge and Lasso regressions are shrinkage moethods. Functions to fit models are in package `"glmnet"`. Functions to fit with PCR and PLSR are available in the package `"pls"`.

Model Building Process

First, we set seed to 200 and run the corresponding fitting function on the train set using ten-fold cross-validation whih works by resampling data. `cv.glmnet()` performs 10-fold cross-validation by default and computes an intercept term and standardizes the variables by default. Because we already mean-centered and standardized all the variables in `scaled_credit`, we use the arguments `intercept = FALSE`, and `standardize = FALSE`. The grid for the argument `lambda` of: `grid <- 10^seq(10, -2, length = 100)`, with functions `pcr()` and `plsr()` use the argument `validation = "CV"` to perform 10-fold cross-validation.

Second, the output from the fitting function give us a list of models. In order to select the best model: in RR and LR, we look for `$lambda.min` from the output of `cv.glmnet()`; in PCR and PLSR, we look for the minimum `$validation$PRESS` from the outputs of `pcr()` and `plsr()`.

Third, after we identify the "best" model, we use the **test set** to compute the test Mean Square Error (test MSE). We will use these values to compare the performance of all the models in next `results` section.

Last but not least, we refit the model on the **full data set** using the parameter chosen by cross-validation. This fit gives the "official" coefficient estimates.

All output we saved and plots we got will be anaylysis in next section – `Results`