Take Home Part

Jinze Gu SID:24968967

March 14, 2014

Introduction

Concrete is one of the most widely used artificial construction materials nowadays. Among all of the benchmarks evaluating the quality of a kind of concrete, people are more concerned about the compressive strength, which is determined by the indegredients and age of concretes. Thus, we want to build a regression model which can help us to predict the compressive strength of a concrete based on its composition and age. Specifically, we have 1030 records of lab data of conrete including the concrete's composition as well as age. The response variable is conceret compressive strength (MPa), and predictors includes cement, fly ash, water, coarse aggregate and so on.

Methods

First of all, it is indicated by the abstract of data that compressive strength is a highly non linear response, so I think it is pretty clear that if we use all of the predictors, we are not likely to have a perfect linear model to fit the data. Thus, I considered to use a sparse model, the intention is that high dimensional non linear points can be better fitted using linear method in a low dimension. For example, any non linear two dimensional data once projected to one dimension is just a line. The first thing I considered is lasso, but if I use the whole data set to construct a lasso regression model, none of the predictors will be dropped.Then I tried to conduct PCA and then use linear method on the projected data without dropping predictor. However, the bad screeplot as well as the complicated interepretation pushed me to turn to other methods. Hence, I decided to drop the predictor mannually before lasso since I notice there are three predictors with many 0 entries and then conduct lasso.

Secondly, the non linear relationship reminds me of the polynomial regression; the polynomial fit may have a better fit as well as prediction than normal linear method. Thus the second method I tried is to fit a polynomial regression on the data. But I am afraid that we may potentially add more noise to our model by using normal polynomial regression, then I decided to use a sparse

polynomial regressionby combining polynomial regression with lasso.

Sparse Linear Model

Reasoning

After we read in all of the data, the first impression is that some of ingredients have a number of 0 input while the others does not. Thus we made a table called "zeros" to summarize this infromation. It turns out that Blast Furnace Slag, Fly Ash and Superplasticizer are the three ingredients containing many 0 values; thus, these three ingredients are more likely to be flexible in our model construction. In other words, we usually do not produce concrete without the other components but we could or possibly can build better concrete without Blast Furnace Slag, Fly Ash and Superplasticize. Hence, my intention is to sort of classify the concrete into 5 categories and use different sparse models to predict the compressive stength of different concerets.

Procedure

By the intention I mentioned before, I need to select different categories of data in order to conduct my sparse method.
By the correlation plot, we do not have very correlated predictors, so I did not go throught different combination of three selected predictors(the low correlation indicates that we should treat their effects independently) and only classfied the data into 5 categories:
Non zero: Eliminate the 0 values in predictors Blast Furnace Slag, Fly Ash and Superplasticizer and leave only non zero entries. I have 209 obervations, and I denote it as wo.zero(without zero).
All Three: I selected the data that does not contain any 0 values of Blast Furnace Slag, Fly Ash and Superplasticizer. I denoted it as wo.three.
With Blast Furnace Slag non zero: I selected the part containing all of the nonzero values of Blast Furnace Slag and dropped the Fly Ash and Superplasticizer. I denote it as wo.BFS.
Similarly, I selected the data wo.FA(Fly Ash) and wo.S(superplasticizer).
Now, I use the 5 data sets to do lasso regression respetively and figure out their fitted errors. Specifically, I wrote a function called sparse model that help me to fit a lasso model with the best lambda(based on leave one out cross validation)Then, I use the model constructed by one data set to predict the compressive strength of the other data set and calculate the prediction error. Finally, I plotted the error in the figure "Summary of Prediction and Fitted Errors" where the row name represents the training data set I used to construct the model and column name indicates the test data set I used to calculate prediction error.

Conclusion From Sparse Model

Basically, since I can not fit a consistently good sparse model, I used 5 sparse models based on different data set and compare their prediction error on the different kinds of data set I constructed. By my model construction, I manually drop predictors before I use lasso to do model construction.
Based on the plot, the diagnal entries are just fitted error since we use the training data set as the test data set.
If we want to predict the compressive strength of a concrete which includes all three components: Blast Furnace Slag, Fly Ash and Superplasticizer. I would use the model constructed with all three predictors by lasso since it has the samllest prediction error of 7.96 among all 5 models. If we want to predict the compressive strength of a concrete without component Superplasticizer, then I would use either the model without the three components or with all of the three components since they give prediction errors 4.35 and 4.37 respectively.
In conclusion, the linear sparse model can be generally

Polynomial Model With Lasso

Reasoning

Since the orginal data is said to have non linear relationship, I believe polynomial would be better than straight line in fitting a non linear relationship. Besides, I would like to see if I can find a generally better model so that I do not need to separate the data into different parts and use different models to fit the data.

Procedure

Basically, I constructed a function called make poly which helps me to design the data with specific n degree of design matrix required for polynomial regression. Then I put it into a lasso model and use the cross validated lambda to fit a polynomial sparse model. However, it is hard to decide which degree I should use. So I did a for loop to run over all degrees from 1 to 12(which is the maximum number I can use based on the warning output). Then I plotted the figure "Cross Validate Error Over Different Degree of Polynomials". It turns out degree of ten hits almost the bottom of the error(degree 12 is slightly better but it does not worth to add 16 predictors for a slightly better model).

Conclusion of Sparse Polynomial

After I fit a twn degree sparse polynomial regression, I have kept all of my predictors. It seems the model has not been over fitted yet even if I have 81 predictors in the model. However, I feel this is possible because I may need more

predictors to linearly represent the data better. No matter sparse polynomial regression or lasso regression, all the method we used is limited in linear scenario. Although the polynomial fitting is better, I may need to find some other non linear methods for the data set from the perspective of better prediction.

Final Conclusion

There are several conclusions I have drawn by doing the project:
One: Linear method is not enough for us to deal with many practical problems where there are no obvious indications that linear method would be a good choice. If we used linear methods to solve non linear problem, we either need to reduce the dimension of our design matrix or increase the dimension of our design matrix in order to have a better fit.
Two: we need to consider prediction error and fitted error together in order to compare different methods. Some time a better fit could be an indicator that we are not able to have better prediction. For example, I tried the k nearest neighbor and did not propose it because we are able to have a great fit if k equals to the number of data points but the prediction is just horrible.
Three: Lasso can help us to do model selection but it is not able to find the best combination of predictors, especially from the perspective of prediction. If we can not drop any predictor from a linear model, then lasso is no better than a linear model fit.
Four: even if for the same data set, we may be able to have a better model fitting by classifying data into different parts and fit each part with slightly different model.