EE219 Project 1

Regression Analysis

Winter 2017

Introduction: Regression analysis is a statistical procedure for estimating the relationship between a target variable and a set of potentially relevant variables. Usually, in a stochastic setting, regression analysis estimates the conditional expectation of the response variable given the other variables; roughly speaking, the average value of the response variable for a realization of the other variables. Such an analysis is highly dependent on the underlying data generating process, the assumptions on which guide the choice of the regression function and the constraints we impose on the relationship that we want to estimate. If the assumed model is excessively complex, over-fitting occurs, which diminishes the predictive performance of the model.

In this project, we explore basic regression models on two datasets, along with basic techniques to handle over-fitting; namely cross-validation, and regularization. With cross-validation, we test how the model generalizes to unseen data by evaluating its performance on a set of data not used for training, while with regularization we penalize overly complex models.

Network backup Dataset

1) Load the dataset. You can download the dataset from this link. The dataset is comprised of simulated traffic data on a backup system in a network. The system monitors the files residing in a destination machine and copies their changes in four hours cycles. At the end of each backup process, the size of the data moved to the destination as well as the duration it took are logged, to be used for developing prediction models. We define a workflow as a task that backs up data from a group of files, which have similar patterns of change in terms of size over time. In other words, how the files are changing varies among different workflows and it depends on different factors like the day of the week it happens and the time of the day. The dataset has around 18000 data points with the following columns:

- Week index
- Day of the week at which the file back up has started
- Backup start time-Hour of the day: the exact time that the backup process is completed
- Workflow ID
- File name
- Backup size: the size of the file that is backed up in that cycle in GB
- Backup time: the duration of the backup procedure in hour

Given this dataset, we want to develop prediction models for predicting the size of the data being backed up as well as the time a backup process may take (refer to “Size of Backup” and “Backup Time” columns in the dataset). To get an idea on the type of relationships in your dataset, for each workflow, plot the actual copy sizes of all the files on a time period of 20 days. Can you identify any repeating patterns?

2) Let us now predict the copy size of a file given the other attributes.

a) Fit a linear regression model with copy size as the target variable and the other attributes as the features. We use ordinary least square as the penalty function. That is

$$\min \| Y - X\beta \|_2,$$

where the minimization is on the coefficient vector β.

Perform a 10-fold cross validation. That is, split the data randomly into 10 parts and each time take 90% of the data for training and intentionally regard the other 10% to have an unknown response variable for testing. After training the model compare the predicted value of the 10% testing data with their actual values. If we split the data into 10 equally sized parts and test 10 times, each time testing for one of these 10 parts while training on the other 9 parts, we would achieve "10-fold Cross-validation".

Analyze the significance of different variables with the statistics obtained from the model you have trained (e.g. p-value with complete description) and report your obtained Root Mean Squared Error (RMSE). Evaluate how well your model fits the data by providing “Fitted values and actual values scattered plot over time”, and “residuals versus fitted values plot”.

b) Use a random forest regression model for this same task. Set the parameters of your model with the following initial values
   - Number of trees: 20
   - Depth of each tree: 4

And you can initialize the maximum number of features at each node to be the number of features you have. By tuning the parameters you can improve the performance of the model. Using more trees reduces the variance. Tune the parameters of your model and report the best RMSE you can get. Compare the performance in RMSE with the linear regression model developed earlier.

The output of the random forest algorithm gives a lot of insight about the data. ***Interpret*** the output of your random forest model. Which features are more important? Can you identify the patterns you observed in part 1 in your fitted model?

c) Now use a neural network regression model. ***Explain*** the major parameters of your model and how they affect the performance in RMSE.

3) Predict the Backup size for each of the workflows separately. ***Explain*** if the fit is improved? Note that in this case, you are fitting a piece-wise linear regression model.

Now, try fitting a more complex regression function to your data. You can try a polynomial function of your variables? Try increasing the degree of the polynomial to improve your fit. Again, use a 10 fold cross validation to evaluate your results. Plot the RMSE of the trained model against the degree of the polynomial you fit first for a fixed training and test set, and then for the average RMSE using cross validation. Can you find a threshold on the degree of the fitted polynomial beyond which the generalization error of your model gets worse?

Can you explain how cross validation helps controlling the complexity of your model?

Boston Housing Dataset

Load the dataset. You can download the dataset from this link. This dataset concerns housing values in the suburbs of the greater Boston area and is taken from the StatLib library which is maintained at Carnegie Mellon University. There are around 500 data points with the following features

- CRIM: per capita crime rate by town
- ZN: proportion of residential land zoned for lots over 25,000 sq. ft.
- INDUS: proportion of non-retail business acres per town
- CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- NOX: nitric oxides concentration (parts per 10 million)
- RM: average number of rooms per dwelling
- AGE: proportion of owner-occupied units built prior to 1940
- DIS: weighted distances to five Boston employment centers
- RAD: index of accessibility to radial highways
- TAX: full-value property-tax rate per $10,000
- PTRATIO: pupil-teacher ratio by town
- B: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
- LSTAT: % lower status of the population
- MEDV: Median value of owner-occupied homes in $1000's

4) Fit a linear regression model with MEDV as the target variable and the other attributes as the features and ordinary least square as the penalty function. Perform a 10 fold cross validation, ***analyze*** the significance of different variables with the statistics obtained from the model you have trained, and the averaged Root Mean Squared Error (RMSE), and plot the same curves as in part 2. Repeat the same steps for a polynomial regression function and find the optimal degree of fit as in part 3.

5) In this part, we try to control over fitting via regularization of the parameters. The idea behind regularization is to constrain the coefficient vector to lie in a less complex manifold rather than R^p, with p being the number of features. In this part we explore common regularization techniques that impose a further penalty on the size of the regression coefficients along with the sum of residuals. Namely we consider ridge and lasso regression techniques, which correspond to ℓ_1 and ℓ_2 regularizations respectively.

a) Tune the complexity parameter α of the ridge regression below in the range $\{1, 0.1, 0.01, 0.001\}$ and report the best RMSE obtained via 10-fold cross validation.

$$\min \| Y - X\beta \|_2^2 + \alpha \| \beta \|_2^2$$

Compare the value of the optimal coefficients obtained this way with the un-regularized model and the ℓ_1 regularized model below.

b) Repeat the previous part for Lasso regularization as formulated below

$$\min \| Y - X\beta \|_2^2 + \alpha \| \beta \|_1.$$

Use an appropriate normalization for the range of α, if needed.

Submission: Please submit a zip file containing your report, and your codes with a readme file on how to run your code to ee219.winter2017@gmail.com. The zip file should be named as "Project1_UID1_UID2_..._UIDn.zip" where UIDx are student ID numbers of the team members. If you had any questions you can send an email to the same address.