ES 647 - Pattern Recognition & Machine Learning

Assignment 1

Release Date : January 20, 2018
Due Date: January 31, 2018

0.1 Instructions

1. Follow the honor code of the institute while doing any assignment. Any violation in that would be taken quite seriously.

2. You can consult/discuss with any of your friend to develop the solution strategy. You can also take help of your friend in setting up your machine. However, the final solution and code should be written by you from scratch and you should not copy even a single bit of it from others. You should acknowledge the help taken from your friend(s) in your code at the top part (in comments section).

3. You will be required to submit one single **.py** file for the entire Assignment 1. The submission needs to be done via Canvas only.

4. You should name the file as follows: *RollNumber_assignment1.py* . Files not following this naming convention will not be evaluated.

5. The submission should be done by 11:59 PM on the due date. Late submissions will be penalized.

6. All the plots should be properly titled. The axes should have proper title and markers. In any plot, the width of curves and markers (if any) should be chosen sufficiently so that the plot is visible properly. Further, highlight gridlines or additional lines wherever it make sense and wherever it adds more value to the plot.

7. For all the plots in the Assignment 1, test error should be plotted in **Red** color, training error should be plotted in **Blue** color and R^2 score in **Black** color.

8. For any kind of clarification on the problem definition and what you need to do in this assignment, you can contact our TAs (Rachit and Harsha) via email communication in Canvas. You can also post your queries in the announcement section of Canvas and let your friends or TAs answer that eventually. You also feel free to answer the queries of others on canvas (but don't provide the solution).

0.2 Setting up Your System

Setup your system for this as well all the future assignments

- Install Anaconda on your machine. You can get more details from the Assignment0 document already uploaded on canvas.
- Install any IDE of your choice (recommended: Sublime Text)

0.3 Familiarity with Python

All the assignments should be done in Python only. You may want to watch the videos from the following play lists (or any other list of your choice) to acquaint yourself with the Python prerequisites.

1. Corey Schafer
2. Sentdex - Python 3
3. Sentdex - Machine Learning with Python

Specifically, you will be needing the knowledge of following Python topics more often in order to complete all the assignments: List, Numpy and Scipy. It is advisable that you familiarize yourself with these topics properly eventually. Finally, you may read the Sklearn documentation for help on Python's inbuilt APIs for machine learning tasks.

0.4 Problem Set for Assignment 1

Load the Boston housing dataset from the Sklearn datasets and write a Python code to accomplish the following tasks:

1. Plot a curve with number of training examples on X-axis and *Training* and *Test Error* (both on the same plot in specific colors) for the Least Square Regression (without regularization or say $\lambda = 0$ in regularization) on the Y-axis. You can use inbuilt function to fit the Least Square Regression Model. Use Data Normalization. You should use $50\%, 60\%, 70\%$, 80%, $90\%, 95\%$, and 99% as the different sizes for the training set while plotting this curve. Test set is the remaining part of the dataset. Further, you also need to make another plot having R^2 score on the Y-axis and number of training examples on the X-axis for the same set of experiments. Also repeat the same experiment with l_2-regularization with values of λ as $0.01, 0.1, 1$ using the inbuilt function for l_2-regularization. You should use a gridplot in matplotlib in order to show all these plots. This grid will have 2 rows and 4 columns displaying the Training and Test error plots in row 1 (different plots for different values of $\lambda = 0, 0.01, 0.1, 1$) and R^2 score plots in row 2 (different plots for different values of $\lambda = 0, 0.01, 0.1, 1$). **Keep the scale of X and Y axis same on all plots**. Also every time you take a certain number of training examples, shuffle the data.

2. Plot the curve with the value of λ in Ridge regression on the X-axis and Training and Test Error (on the same plot) on the Y axis. Fit the ridge regression model using Sklearn inbuilt function **Ridge** for l_2-regularization. Use normalization. You should vary the values of λ as $[0, 0.0001, 0.001, 0.01, 0.1, 1, 1.5, 2, 3, 4, 5]$. You need to make such a plot separately for each of the following training set size – 99%, 90%,80% and 70%. **Keep the scale of X and Y axis same on all plots**. Furthermore, on each of these plots, you should also plot the validation error by using the inbuilt cross validation function in SKlearn (namely, **RidgeCV** - you are encouraged to read the documentation for RidgeCV function). Use 5-folds for the cross validation. You should use the same set of λ values for this validation error plots also. The validation error plot should also be in the same color as test error but use dotted lines. So there will be 3 curves on same plot. Remember that test error will be calculated on the test set whereas validation error has nothing to do with the test set. Also make separate plots with value of λ on X- axis and R^2 score on Y-axis for each of the percentages of training examples mentioned above and value of λ between 0 and 5 varying them as shown above. Again use the concept of matplotlib grid for showing all these plots. Your grid will have 2 rows and 4 columns displaying the Training,Test and Validation error plots in row 1 for different number of training examples and R^2 score plots in row 2.

3. Repeat the process in question 2 with l_1-regularization by using inbuilt functions for LASSO regression and LASSO regression with cross validation.

4. Repeat the process in questions 1 and 2 but now use your own code for data normalization, fitting least square regression model, ridge regression model, calculating training error, test error and R^2 score. You do not need to do cross validation for this question.

5. Repeat the tasks in questions 1 through 4 for the diabetes dataset.