\CS 596 Machine Learning

**Homework Assignment 2**

**(10 Credits)**

Due: 11:59 pm, Friday, September 21st, 2018

The goal of this homework assignment is to practice the Python implementation of two methods for training Linear Regression models: *normal equations* and gradient descent. See detailed instructions below.

**Overview**

The task is to build a linear regression model that predicts the GPAs of university students from two features, Math SAT and Verb SAT.

* Sub-task 1) Train the model using normal equation method.
* Sub-task 2) Train the model using gradient descent.
* Sub-task 3) Evaluate the two models on a testing set, and compare their performance.

The Python and data files included are: ha2.py, GD.py, sat\_gpa.csv

**Datasets**

The file sat\_gpa.csv contains all training and testing data. It has 105 rows and 3 columns. Each row is the record of a student. The three columns are Math SAT score, Verb SAT score, and University GPA. The first two columns are the features, and the third is the output. The first 60 rows are used for training and the rest 45 rows are used for testing.

**Sub-task 1)**

1 credit.

Implement the normal equation method for linear regression in ha2.py:



Complete the missing code between the “### START TODO ###” and “### END TODO ###” blocks from line 48 to 51.

**Sub-task 2)**

4 credits

Implement the gradientDescent function defined in GD.py.

Complete four lines of missing code between the TODO blocks from line 16 to 37 in GD.py, and each line is worth 1 credit.

**Sub-task 3)**

1 credit

Evaluate the two models on the same testing set, and compute their Mean Square Errors (MSE) respectively.

Complete the missing code between the TODO blocks from line 84 to 94.

**Sub-task 4)**

1 credit

Try at least two different learning rate other than 0.1 (default value), by changing the variable ALPHA in line 21.

**Report**

3 credit

Write up a report that includes these items:

* Show the parameters () of both methods, normal equation and gradient descent.
* Tell which model is a better fit on the training data, by comparing their residual sum of squares (RSS), which will be printed by the program.
* Compare the models’ performance (MSE) on the testing data. Are they different or similar? Briefly discuss why.
* Show at least 3 plots of cost vs. iteration (from gradient descent method), using 3 different learning rates. The default one () should be included. Discuss which case converges fastest, and whether changing the learning rate will influence the model’s performance on both training and testing data.

Feel free to add any additional information you find relevant.

**How to run your code**

Once you are done coding, go to the terminal and enter your directory where ha2.py and GD.py are located, and type “python ha2.py” or “python3 ha2.py”.

After that, you can see the terminal printing out the model parameters, RSS on training data, and MSE on testing data.

Also, you will see a PDF file “cost\_iter\_alpha\*\*.pdf” automatically saved to your directory, which contains the cost vs. iteration plot of the gradient descent method, and the number “\*\*” in file name is your current learning rate.

**Submission requirements**

1. Submit two modified Python files, ha2.py and GD.py. If running your code gives error, -3 credits penalty will be placed.
2. Submit the report.
3. No hard copy allowed.