PA2

May 21, 2019

1 Homework #2 Programming Assignment

CSCI567, Spring 2019Victor Adamchik**Due: 11:59 pm, Feb. 08, 2019**

1.0.1 Before you start:

There is a known issue with vocareum that caused by vocareum website, not us. If you submit your homework twice in a row without refresh the page, the grade book score might not be updated. So I suggest you to refresh your page before doing another submission. Also, this project have two part. You need to submit each part seperately on vocareum.

1.1 Office Hour for Project Assignment 2

Question for Part 1, we suggest you go to: 2nd Floor-201A: 1:00am - 3:00am, Thursday, January 31, 2019 for Yang, Fang Question for Part 2, we suggest you go to: 2nd Floor-201A: 3:00am - 5:00am, Thursday, January 31, 2019 for Cheng-Ju, Lin We will also hold office hour for the first week of February, we will announce the time shortly. Also, you can post your question on Piazza under pa-2 folder. We will try our best to answer all questions as soon as possible. Everyone in the class is welcomed to answer these questions. We really appreciate that. Please make sure you read previous posts before creating a new post in case your question has been answered before. We prefer Piazza than emails. However, if you have any urgent issue, feel free to send an email to both of us. Yang, Fang: yangfang@usc.edu Cheng-Ju, Lin: chengjul@usc.edu

1.2 Problem 1 Regression (40 points)

For this Assignment you are asked to implement linear regression. You are asked to implement 6 python functions for linear regression in linear_regression.py. The input and output of the functions are specified in linear_regression.py. Note that we have already appended the column of 1's to the feature matrix, so that you do not need to modify the data yourself. We have provide a linear_regression_test.py for you to run some of your code and do some testing. Read the code, run it and check the result during your implementation. Submission: All you need to submit is linear_regression.py

1.2.1 Q1.1 Mean Square Error (5 points)

First, You need to implement the mean sqaure error. This is very similar to the RSS you learned from the lecture note page 15 except you need take the arithmetic mean of the error. numpy already have a mean function for that, feel free to use it. There should be simple as one or two lines of code and there is only one thing you need to be careful: please check the dimension of the input w, X and y before you do matrix dot operation. You might need to take a transpose of some matrix in order to get the right shape. Report the mean square error of the model on the give n test set, no rounding for the result. The type of the error is <class 'numpy.float64'>, you can do type(err) to check yours. * (5 points) `TODO 1` You need to complete `def mean_square_error(w, X, y)` in `linear_regression.py`

1.2.2 Q1.2 Linear Regression (5 points)

Now, you need to train your model. In linear regression model, it means find the weight for each feature. Check the lecture note page 21 and find the expression of general least square solution. Again, be careful with your matrix shape. Also, use numpy inverse funciton so you don't need to create your own. The shape of your return w should be (12,) if you train your model with the data we provide you Implement linear regression with no regularization and return the model parameters. Again, the implementation should be as simple as one or two lines of code. Use numpy instead of crazy nested for loop in your implementation. You don't need to worry about non-invertible matrix. We will take care of it in Q1.3. * (5 points) `TODO 2` You need to complete `def linear_regression_noreg(X, y)` in `linear_regression.py`

Once you finish Q1.1 and Q1.2, you should be able to run your linear_regression_test.py. Read the output, check your dimension of w, and MSE for training, evaluation and testing dataset. The MSE for all of them should between 0.5~0.6. Otherewise, there must be something wrong with your implementation.

1.2.3 Q1.3 Handle Non-invertible Matrix (5 points)

There are cases that during the calculation, the matrix are non-invertible. We manually created that situation in the data_loader.py line 40. We simply manully set one row and one column in the dataset to be zero. Thus, we will get similar result as letcure note page 29. Now, you need to implement the solution on lecture note page 31. Here is the rul: in this assignment, if the smallest absolute value of all eigenvalue of a matrix is smaller than 10^{-5}, the matrix is non-invertible. If the matrix is non-invertible, keep adding $10^{-1} * I$ until it is invertible. You can use numpy functions to get eigen value for a matrix, get the absolute value ,find the minimum and create identity matrix, just search for them. * (10 points) `TODO 3` You need to complete `def linear_regression_invertible(X, y)` in `linear_regression.py`

Once you finish Q1.3, run linear_regression_test.py again, the MSE should be between 0.5~0.6. Otherewise, there must be something wrong with your implementation.

1.2.4 Q1.4 Regularized Linear Regression (5 points)

To prevent overfitting, we usually add regularization. For now, we will focus on L2 regularization, same as the one in lecture note page 50. Implement the regularized_linear_regression(X,y,lambda) * (10 points) `TODO 4` You need to complete `def regularized_linear_regression(X, y, lambd)` in `linear_regression.py`

Once you finish Q1.4, run linear_regression_test.py again. Compare your result with Q1.3, did you find something interesting? Think about why or check the statement in red on lecture note page 50.