Assignment 1

August 2020

Instructions

- This assignment should be completed individually.
- Do not look at solutions to this assignment or related ones on the Internet.
- The files related to the assignment are present in `lab1-rollno.zip` folder. Extract it and upload it on moodle in the same .zip format after completion and after replacing the string "rollno" with your actual roll number. For example, if you roll number is 00405036, then single zip folder that you will upload will be named "lab1-00405036.zip". Also collate all the CS337 based theory solutions into ONE pdf file named `answers.pdf`. Include `answers.pdf` inside the zip folder mentioned above and submit the zip folder.
- Answers to all subjective questions need to be placed in single pdf `answers.pdf` including all plots and figures and uploaded.
- Only add/modify code between `TODO` and `END TODO` unless specified otherwise
- In this assignment, you will perform Ridge, Lasso Regression and Ordinary Least Squares using the old school method of gradient descent.
- This Assignment carries a total of 7 marks for CS337 Theory and 12 marks for CS335 Lab

1 Ordinary Least Squares (OLS) Regression in one variable

1. Gradient descent is an optimization algorithm used to find the values of parameters (coefficients) of a function (f) that minimizes a cost function.
2. As for linear regression for the single variable case (that for 1 dimensional input x), we will use the standard $y = wx + b$ slope equation where[1] w is the line's slope and b is the line's y-intercept. To find the best line for our data, we need to find the best set of slope w and y-intercept in the form of b's value.
3. To get started with, w and b are randomly initialized. Hence, we get a random line in the beginning. Our goal is to update these values so that the resulting line gives the least error.

[1] Adopting the notation from 'single_var_reg.py', we try to maintain the same notation across the theory and lab problems in Sections 1.1 and 1.2 respective

4. We will use mean squared error 'mse' as the cost function, that calculates the error between the actual value output, and prediction from the hypothesis. mse $= (1/2N)((wx + b) - y)^2$
5. This error function is typically convex - that is, cup-shaped. In simple terms, we can say that the result of convexity is that this error function typically has just one minimum (the global minimum).
6. When we start with random values of w and b, we get some value of y correspondingly. Error is minimum at the lowest point on this graph, so our goal is to move down the slope to finally reach the bottom-most point.
7. The slope of the tangent at any point on a graph is equal to the derivative of the graph w.r.t. input variables.
8. The slope of tangent at the bottom-most point on the graph is 0, *i.e.*, the partial derivatives of mse at the bottom-most point are 0. To get to the bottom-most point, we have to move in the direction of the slope. That is, we will update values of w and b, such that we eventually get to the optimum values, where error function is minimum.
9. The update equations are

$$\begin{pmatrix} w^{new} \\ b^{new} \end{pmatrix} = \begin{pmatrix} w^{old} \\ b^{old} \end{pmatrix} - \eta \nabla mse(w^{old}, b^{old})$$

10. Here, $\nabla mse(w^{old}, b^{old})$ denotes the 'gradient' vector $\nabla mse(w, b)$ evaluated at w^{old}, b^{old}. Further, $\nabla mse(w, b)$ is defined as

$$\nabla mse(w, b) = \begin{pmatrix} \frac{\partial mse(w,b)}{\partial w} \\ \frac{mse(w,b)}{\partial b} \end{pmatrix}$$

and η is called the 'learning rate' that determines how large the steps should be in the direction of the gradient. If the value of η is set to be very small, reaching the optimum value is guaranteed, but it will take a lot of time to converge. If η is very large, the values of w and b might overshoot the optimal values, and then the error will start to increase instead of decreasing. Hence, learning rate plays an important part in convex optimization.

1.1 CS337: Theory

Based on the directions specified above, write down the specific expression for $\nabla mse(w, b)$. (0.5 marks)

1.2 CS335: Lab

(a) Complete the function `split_data()` in `utils.py` (0.5 marks)

(b) Complete the function `mse()` in single_var_reg.py (0.5 marks)

(c) Complete the `ordinary_least_squares()` function in single_var_reg.py. You can modify `lr` and `max_iter` if needed. Your code will be tested using submitted values. (3 marks)

(d) Add the generated figure in the report. Interpret and explain the right figure (1 mark)

2 OLS and Ridge Regression

2.1 CS337: Theory

Let N be the number of samples each having D features. Given[2] the feature matrix X ($N \times D$ dimensional matrix) the outputs Y (vector of size N) and W the weights to be learnt, solve following

(a) The predicted outputs $\hat{Y}$ for all the N samples (0.5 marks)

(b) For the minimum squared error loss function $mse = \frac{1}{2N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$. Derive vectorized formula for $\frac{\partial mse}{\partial W}$ (1 Mark)

(c) For Ridge regression loss function $mse = \frac{1}{2N}\sum_{i=1}^{N}(\hat{y}_i - y_i) + \lambda\|W\|^2$. Derive vectorized formula for $\frac{\partial mse}{\partial W}$ (0.5 marks)

2.2 CS335: Lab

Complete the following functions

(a) `normalize()` and `preprocess()` in `utils.py` (0.5 + 1 marks)

(b) `mse()` in `multi_var_reg.py`

(c) `ordinary_least_squares()` and `ridge_regression()` in `multi_var_reg.py`. You can modify `lr` and `max_iter` if needed. Your code will be tested using submitted values. (1.5 + 1.5 marks)

3 Weighted Linear Regression

3.1 CS337: Theory

Consider a dataset of n points where each point x_i is weighted by a factor r_i such that the sum of squares error function becomes-

$$E(w) = \frac{1}{2n}\sum_{i=1}^{n} r_i^2(y_i - w^T x_i)^2$$

Find a closed form solution w^* which can minimize this error.
Hint : First convert the above expression into a matrix expression containing X, Y, W and R, where R is some amalgamation of r_i. (3 marks)

3.2 CS335: Lab

Complete the function `weighted_regression()` using the closed form solution obtained above. (1 mark)

[2]Adopting the notation from 'multi_var_reg.py', we try to maintain the same notation across the theory and lab problems in Sections 1.1 and 1.2 respectively

4 Failure cases of linear regression

4.1 CS335: Lab

This problem uses the code you developed in problem 3 to fit a linear regression to the dataset in file `data_4.csv`. The program throws an error. The command to run the program is

```
$ python3 problem_4.py
```

Rectify this error by modifying the dataset and writing it in the file `modified_data_4.csv`. The command

```
$ python3 problem_4.py --data modified_data_4.csv
```

should now run without errors. Submit the dataset. The modified dataset should contain atleast 3 features. Briefly describe the modification and the reasoning in the answers file as well. (1.5 marks)

4.2 CS337: Theory

Under what condition on the columns of data matrix X (denoted by Φ in our notes) does there exist no solution for the closed form of OLS. Does gradient descent converge to a solution in that case? (1.5 marks)