Assignment 1

August 2021

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, Ordinary Least Squares regression, and Bayesian Linear Regression.
- This Assignment carries a total of 10 marks for CS337 Theory and 15 marks for CS335 Lab
- The code for all the questions should be written in provided ipython notebook only. Don't modify the name/directory structure of the provided `.zip` file.
- Please make sure that your code runs in **python 3.x**. You shuld not import any new python libraries.
- Code should be written in the provided `assignment_1.ipynb` file only.
- All CS 335 questions will be auto graded with private testcases.

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 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. However, for this assignment, we initialize them with zeros. 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.

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 $= \frac{1}{N}\sum_{i=1}^{N}((wx_i + b) - y_i)^2$

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.

Data Description

For this assignment, we assume a ground truth function $y = f(x) = 1 - 3x - 2x^2 + 2.5x^3$. However, as is typical of machine learning, we observe only noisy samples in the datasets that are provided. i.e. The dataset comprises of samples (x, y) such that $y = f(x) + N(0, \sigma^2)$ where σ is a hidden parameter that is not revealed to you. You have to use these samples to learn linear regression models using different algorithms.

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()` (0.5 marks)

(b) Complete the function `mse_single_var()` (0.5 marks)

(c) Complete the `singlevar_grad()` and `singlevar_closedform()` functions. You can modify `lr` and `max_iter` if needed for gradient descent based solution. Add the generated figures in the report. (1.5 + 1.5 marks)

(d) Is it possible to obtain a solution using `singlevar_grad()` such that its training loss is strictly less than that of the solution obtained by `singlevar_closedform()`? If yes, mention the parameters that you obtain in the report. Else, contradict and argue why it is not possible. (1 marks)

2 OLS and Ridge Regression

2.1 CS337: Theory

Let N be the number of samples each having d features. Given 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 $mse = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$. Derive vectorized formula for $\frac{\partial mse}{\partial W}$ (1 Mark)

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

(d) 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)

2.2 CS335: Lab

Complete the following functions

(a) Complete the functions `mse_multi_var` and `mse_regularized` (1 marks)

(b) Complete the functions `multivar_grad` and `multivar_closedform`. Include the generated plots in your report. (1.5 + 1.5 marks)

(c) Complete the functions `multivar_reg_grad` and `multivar_reg_closedform`. Include the generated plots in your report. (1.5 + 1.5 marks)

Bayesian Linear Regression

Now, we will move to Bayesian Linear Regression. The fundamental difference here is that we model (w, b) as a Random Variable unlike the previous approaches that give point estimates of them.
For theory question, you will derive formulae for single variable regression. For programming part, we deal with multi-variable regression. We give you the formulae and you have to complete the code.

3 Bayesian Linear Regression

3.1 CS337: Theory

Let us consider the dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$ where $x_i \in \mathcal{R}$ and $y_i \in \mathcal{R}$. We need to learn the parameter $w \in \mathcal{R}$. For convenience, let us assume the bias $b = 0$. (Note: For the programming part, you need to learn bias as well).

(a) Let us assume the parameter w has a prior given by Gaussian distribution $\mathcal{N}(\mu_0, 1)$. Mention the complete expression for $p(w)$ (0.5 marks)

(b) Because we need a conjugate prior, we also assume that the data likelihood is Gaussian. i.e. $p(y|x; w) = \mathcal{N}(wx, 1)$

(c) Assuming the dataset $\mathcal{D}$ is obtained `IID`, give the expression for $p(D|w)$ (0.5 marks)

(d) Now that we have the expressions for prior and Data likelihood, give expression for the posterior $p(w|\mathcal{D})$ using the Bayes theorem. (0.5 marks)

(e) From the expression above, we immediately realize that the denominator involves a daunting integral that is difficult to evaluate. Here is where we can leverage the conjugate prior assumption and find out $p(w)$ analytically (ignoring the denominator).

(f) We can simplify the numerator of $p(w|\mathcal{D})$ and thus deduce that $p(w|\mathcal{D}) \propto exp\big(-\frac{1}{2}\big[w^2(\sum_i x_i^2 + 1) - 2w(\sum_i y_i x_i + \mu_0) + \sum_i y_i^2 + \mu_0^2\ \big]\ \big)$ (Equation 1)

(g) We know that posterior is also Gaussian by virtue of conjugate prior. Thus posterior has the form $p(w|\mathcal{D}) = N(\mu_N, \sigma_N^2) \propto exp(-\frac{1}{2\sigma_N^2}(w - \mu_N)^2)$ (Equation 2)

(h) Compare Equation (1) and (2) above and thus deduce the values of μ_N, σ_N^2 (2 marks)

(i) What will happen to the estimates above when we observe lots of data. Give an expression for μ_N, σ_N^2 as $N \to \infty$ (0.5 marks)

(j) Briefly explain in one or two lines what you intuitively understand from the estimates in the limit $N \to \infty$ (0.5 marks)

3.2 MLE Estimate

(a) As we studied in class, in MLE estimate we maximize the (log) data likelihood i.e., $w^* = argmax_w\ p(D|w)$. For the above problem find the MLE estimate w^* (1 mark)

(b) compare and comment on the MLE and Bayesian estimate (in the limit of ∞ data) that you obtain (0.5 mark)

3.3 CS335: Lab

(a) Complete the function `bayesion_lr()`. Include the generated plots in your report. (3 marks)

(b) For this program, you will not encounter the problem mentioned in the question 2.1(d). Briefly explain why? (1 mark)

4 Conclusion

We saw multiple methods to learn the linear regression weights (w, b) for fitting the function $f(x)$. Compare different methods and opine which one you think is superior. This is a open-ended question and we encourage you to compare for example running time, convergece, # iterations etc and explain briefly in your report with plots where applicable. (bonus 1 mark)