

Machine Learning

COMP 5630/ COMP 6630/ COMP 6630 - D01

Instructor: Dr. Shubhra (“Santu”) Karmaker

TA 1: Dongji Feng

TA 2: Souvika Sarkar

Department of Computer Science and Software Engineering

Auburn University

Fall, 2022

September 28, 2022

Assignment #5

Linear Regression and Regularization

Submission Instructions

This assignment is due Tuesday, October 11, 2022, at 11:59 pm. Please submit your solutions via Canvas (<https://auburn.instructure.com/>). You should submit your assignment as a PDF file. Please do not include blurry scanned/photographed equations as they are difficult for us to grade.

Late Submission Policy

The late submission policy for assignments will be as follows unless otherwise specified:

1. 75% credit within 0-48 hours after the submission deadline.
2. 50% credit within 48-96 hours after the submission deadline.
3. 0% credit beyond 96 hours after the submission deadline.

Tasks

1 Linear Regression [30 pts]

Suppose that, $y = w_0 + w_1x_1 + w_2x_2 + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$

- a) [10 pts] Write down an expression for $P(y|x_1, x_2)$.
- b) [10 pts] Assume you are given a set of training observations $(x_1^{(i)}, x_2^{(i)}, y^{(i)})$ for $i = 1, \dots, n$. Write down the conditional log-likelihood of this training data. Drop any constants that do not depend on the parameters w_0 , w_1 , or w_2 .
- c) [10 pts] Based on your answer, show that finding the MLE of that conditional log-likelihood is equivalent to minimizing least squares,

$$\frac{1}{2} \sum_{i=1}^n \left\{ y^{(i)} - (w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)}) \right\}^2$$

2 Regularization [30 pts]

- a) [15 pts] Find the partial derivative of the regularized least squares problem:

$$\frac{1}{2} \sum_{i=1}^n \left\{ y^{(i)} - (w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)}) \right\}^2 + \frac{\lambda}{2} \|w\|^2$$

with respect to w_0 , w_1 , and w_2 . Although there is a closed-form solution to this problem, there are situations in practice where we solve this via gradient descent. Write down the gradient descent update rules for w_0 , w_1 , and w_2 .

- b) [15 pts] Suppose that $w_1, w_2 \sim \mathcal{N}(0, \tau^2)$. Prove that, the MAP estimate of w_0 , w_1 , and w_2 with this prior is equivalent to minimizing the above regularized least squares problem with $\lambda = \frac{\sigma^2}{\tau^2}$.

3 Programming: Implement Linear Regression [40 pts]

In this assignment, you will implement Linear Regression for a small dataset. Note that you will implement *Linear Regression* from scratch, and no external libraries like Tensorflow, Sklearn, or Pytorch can be used. The data set (“numpydataset.csv”) has been provided on canvas as part of this assignment. We have also provided a template code (“Linear Regression.py”) to simplify the input, output, and plot functions.

You will complete the following tasks for this programming assignment:

- a) [10 pts] Implement a method `MSE()` that will compute the loss incurred at each epoch. We aim to minimize the distance between the data and the line after each epoch. An epoch is a single iteration over a dataset of features. The higher the cost (or Error), the more the parameter values need to be changed in order to bring it down. The Error function is represented as $E = \frac{1}{n} \sum_{i=0}^n (y_i - \hat{y})^2$, where y_i is the actual value, \hat{y} is the predicted value, n is the total number of data points or samples. Invoke the function to calculate the initial loss when $m = 0$ and $b = 0$. We are using m and w interchangeably because they both refer to the weight or the slope.

- b) **[20 pts]** Your aim in Linear Regression is to minimize the MSE by using the GRADIENT DESCENT Algorithm. Extensively used in Machine Learning, this process involves finding the local minimum of a function. The sole idea behind this algorithm is, given a certain learning rate γ , to take iterative steps (slowly) in the direction of $-\nabla E$ (negative gradient) in order to minimize the error rate. Each iteration of gradient descent updates the w and the b (collectively denoted by θ) according to

$$\theta^{t+1} = \theta^t - \gamma \frac{\partial E(X, \theta)}{\partial \theta}$$

where θ^t denotes the weights and the bias at iteration t .

- c) **[10 pts]** Using the template code, compute the MSE after 100 iterations for learning rate 0.001 and 0.01 separately. The code will also create two plots (corresponding to different learning rates) for the fitted line along with the data points for you. Which learning rate do you think is working better? Why? Explain your answer with relevant data and observations.

Disclaimers: This assignment re-uses some materials from the publicly available website: [CMU Introduction to Machine Learning Course, 10-315, Spring 2019](#). I personally thank Prof. Maria-Florina Balcan for sharing her teaching materials publicly. This assignment is exclusively used for instructional purposes.