CS 189 Introduction to Machine Learning
Fall 2019 Jennifer Listgarten & Stella Yu HW0

This homework is due **Tuesday, September 3 at 10 p.m.**

1 Getting Started

Read through this page carefully. You may typeset your homework in latex or submit neatly handwritten/scanned solutions. Please start each question on a new page. Deliverables:

1. Submit a PDF of your writeup, **with an appendix for your code**, to assignment on Gradescope, "HW? Write-Up". If there are graphs, include those graphs in the correct sections. Do not simply reference your appendix.
2. If there is code, submit all code needed to reproduce your results, "HW? Code".
3. If there is a test set, submit your test set evaluation results, "HW? Test Set".

2 Sample Submission

Please submit a plain text file to the Gradescope programming assignment "Homework 0 Test Set":

1. Containing 5 rows, where each row has only one value "1".
2. No spaces or miscellaneous characters.
3. Name it "submission.txt".

3 Linear Algebra Review

Consider the vectors $\mathbf{u} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$. Define the matrix $\mathbf{M} = \mathbf{u}\, \mathbf{v}^\top$.

(a) Compute the eigenvalues and eigenvectors of the matrix $\mathbf{M}$.

(b) Compute the rank and the determinant of the matrix $\mathbf{M}$. What is the dimension of the nullspace of the matrix $\mathbf{M}$?

(c) Now consider two non-zero vectors $\mathbf{p}$ and $\mathbf{q}$ in $\mathbb{R}^d$ and the matrix $\mathbf{N} = \mathbf{p}\, \mathbf{q}^\top$. Repeat the computations for parts (a) and (b) for the matrix $\mathbf{N}$.

Please explain your computations/arguments precisely.

4 Linear Regression and Adversarial Noise

In this question, we will investigate how the presence of noise in the data can adversely affect the model that we learn from it.

Suppose we obtain a training dataset consisting of n points (x_i, y_i) where $n \geq 2$. In case of no noise in the system, these set of points lie on a line given by $y = w_1 x + w_2$, i.e, for each i, $y_i = w_1 x_i + w_2$. The variable x is commonly referred to as the covariate [1] and y's are referred to as the observation. Suppose that all x_i's are distinct and non-zero. Our task is to estimate the slope w_1 and the y-intercept w_2 from the training data. We call the pair (w_1, w_2) as the true model.

Suppose that an adversary modifies our data by corrupting the observations and we now have the training data $(x_i, \tilde{y}_i)$ where $\tilde{y}_i = y_i + \epsilon_i$ and the noise ϵ_i is chosen by the adversary. Note that the adversary has access to the features x_i but *can not* modify them. Its goal is to trick us into learning a wrong model $(\hat{w}_1, \hat{w}_2)$ from the dataset $\{(x_i, \tilde{y}_i), i = 1, \ldots, n\}$. We denote by $(\hat{w}_1, \hat{w}_2)$ the model that we learn from this dataset $\{(x_i, \tilde{y}_i), i = 1, \ldots, n\}$ using the standard ordinary least-squares regression.

(a) Suppose that the adversary wants us to learn a particular wrong model (w_1^*, w_2^*). If we use standard ordinary least-squares regression, can the adversary *always* (for any choice of w_1^* and w_2^*) fool us by setting a particular value for exactly one ϵ_i (and leaving other observations as it is, i.e., $\tilde{y}_j = y_j, j \neq i$), so that we obtain $\hat{w}_1 = w_1^*$ and $\hat{w}_2 = w_2^*$? If yes, justify by providing a mathematical mechanism for the adversary to set the value of the noise term as a function of the dataset $\{(x_i, y_i), i = 1, \ldots, n\}$ and (w_1^*, w_2^*)? If no, provide a counter example.

(b) Repeat part (a) for the case when the adversary can corrupt two observations, i.e., for the case when the adversary can set up at most two of the ϵ_i's to any non-zero values of its choice.

(c) In the context of machine learning and applications, what lessons do you take-away after working through this problem?

5 Background Review

Please describe the coursework that you have undertaken on the following topics:

(a) Linear Algebra, e.g., EE 16A/B

(b) Optimization, e.g., EECS 127

(c) Probability and stochastic processes, e.g., EECS 126

(d) Vector Calculus, e.g., EECS 127, Math 53.

(e) Also describe your experience with programming, in particular with python, e.g., like in your coursework in CS 61A/B and EE 16A/B.

[1] Besides covariate, some other names for x include feature, regressor, predictor.

6 Your Own Question

Write your own question, and provide a thorough solution.

Writing your own problems is a very important way to really learn the material. The famous "Bloom's Taxonomy" that lists the levels of learning is: Remember, Understand, Apply, Analyze, Evaluate, and Create. Using what you know to create is the top-level. We rarely ask you any HW questions about the lowest level of straight-up remembering, expecting you to be able to do that yourself. (e.g. make yourself flashcards) But we don't want the same to be true about the highest level.

As a practical matter, having some practice at trying to create problems helps you study for exams much better than simply counting on solving existing practice problems. This is because thinking about how to create an interesting problem forces you to really look at the material from the perspective of those who are going to create the exams.

Besides, this is fun. If you want to make a boring problem, go ahead. That is your prerogative. But it is more fun to really engage with the material, discover something interesting, and then come up with a problem that walks others down a journey that lets them share your discovery. You don't have to achieve this every week. But unless you try every week, it probably won't happen ever.