Homework #1

RELEASE DATE: 10/02/2018

DUE DATE: 10/30/2018, BEFORE 14:00

QUESTIONS ABOUT HOMEWORK MATERIALS ARE WELCOMED ON THE FACEBOOK FORUM.

Unless granted by the instructor in advance, you must turn in a printed/written copy of your solutions (without the source code) for all problems.

For problems marked with (*), please follow the guidelines on the course website and upload your source code to designated places. You are encouraged to (but not required to) include a README to help the TAs check your source code. Any programming language/platform is allowed.

Any form of cheating, lying, or plagiarism will not be tolerated. Students can get zero scores and/or fail the class and/or be kicked out of school and/or receive other punishments for those kinds of misconducts.

Discussions on course materials and homework solutions are encouraged. But you should write the final solutions alone and understand them fully. Books, notes, and Internet resources can be consulted, but not copied from.

Since everyone needs to write the final solutions alone, there is absolutely no need to lend your homework solutions and/or source codes to your classmates at any time. In order to maximize the level of fairness in this class, lending and borrowing homework solutions are both regarded as dishonest behaviors and will be punished according to the honesty policy.

You should write your solutions in English or Chinese with the common math notations introduced in class or in the problems. We do not accept solutions written in any other languages.

This homework set comes with 200 points and 20 bonus points. In general, every homework set would come with a full credit of 200 points, with some possible bonus points.

1. (80 points) Go register for the Coursera version of the first part of the class (`https://www.coursera.org/teach/ntumlone-mathematicalfoundations/`) and solve its homework 1. The registration should be totally free. Then, record the highest score that you get within up to 3 trials. Please print out a snapshot of your score as an evidence. (*Hint: The problems below are simple extensions of the Coursera problems.*)

Problems 2-3 are about *Off-Training-Set error*.

Let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N, \mathbf{x}_{N+1}, \ldots, \mathbf{x}_{N+L}\}$ and $\mathcal{Y} = \{-1, +1\}$ (binary classification). Here the set of training examples is $\mathcal{D} = \Big\{(\mathbf{x}_n, y_n)\Big\}_{n=1}^{N}$, where $y_n \in \mathcal{Y}$, and the set of test inputs is $\Big\{\mathbf{x}_{N+\ell}\Big\}_{\ell=1}^{L}$. The *Off-Training-Set error* (OTS) with respect to an underlying target f and a hypothesis g is

$$E_{OTS}(g, f) = \frac{1}{L} \sum_{\ell=1}^{L} [\![g(\mathbf{x}_{N+\ell}) \neq f(\mathbf{x}_{N+\ell})]\!] .$$

2. (20 points) Consider $f(\mathbf{x}) = +1$ for all $\mathbf{x}$ and

$$g(\mathbf{x}) = \begin{cases} +1, & \text{for } \mathbf{x} = \mathbf{x}_k \text{ and } k \text{ is odd and } 1 \leq k \leq N+L \\ -1, & \text{otherwise} \end{cases} .$$

What is the value of $E_{OTS}(g, f)$? Please provide proof of your answer.

3. (20 points) A determistic algorithm $\mathcal{A}$ is defined as a procedure that takes $\mathcal{D}$ as an input, and outputs a hypothesis g. For any two deterministic algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$, if all those f that can "generate" $\mathcal{D}$ in a noiseless setting are equally likely in probability, please prove or disprove that

$$\mathbb{E}_f \Big\{ E_{OTS}\big(\mathcal{A}_1(\mathcal{D}), f\big) \Big\} = \mathbb{E}_f \Big\{ E_{OTS}\big(\mathcal{A}_2(\mathcal{D}), f\big) \Big\}.$$

For Problem 4, consider the bin model introduced in class.

Consider a bin with infinitely many marbles, and let μ be the fraction of orange marbles in the bin, and ν is the fraction of orange marbles in a sample of 10 marbles.

4. (20 points) If $\mu = 0.8$, what is the probability of $\nu \leq 0.1$? What is the probability of $\nu \geq 0.9$? Please provide calculating steps of your answer.

Problems 5-6 illustrate what happens with multiple bins. Please note that the dice is not meant to be thrown for random experiments in this problem. They are just used to bind the six faces together. The probability below only refers to drawing from the bag.

Consider four kinds of dice in a bag, with the same (super large) quantity for each kind.

- A: all even numbers are colored orange, all odd numbers are colored green
- B: all even numbers are colored green, all odd numbers are colored orange
- C: all small (1-3) are colored orange, all large numbers (4-6) are colored green
- D: all small (1-3) are colored green, all large numbers (4-6) are colored orange

5. (20 points) If we pick 5 dice from the bag, what is the probability that we get five green 1's? Please provide calculating steps of your answer.

6. (20 points) If we pick 5 dice from the bag, what is the probability that we get "some number" that is purely green? Please provide calculating steps of your answer. Compare your answer to the previous problem and describe your findings.

For Problem 7, you will play with the PLA algorithm.

First, we use an artificial data set to study PLA. The data set is in

`http://www.csie.ntu.edu.tw/~htlin/course/mlfound18fall/hw1/hw1_7_train.dat`

Note that the file is exactly the same as the one for Cousera Homework 1, Problem 15.

`https://www.csie.ntu.edu.tw/~htlin/mooc/datasets/mlfound_math/hw1_15_train.dat`

Each line of the data set contains one $(\mathbf{x}_n, y_n)$ with $\mathbf{x}_n \in \mathbb{R}^4$. The first 4 numbers of the line contains the components of $\mathbf{x}_n$ orderly, the last number is y_n. Please initialize your algorithm with $\mathbf{w} = \mathbf{0}$ and take sign(0) as -1. As a friendly reminder, remember to add $x_0 = 1$ as always!

7. (*, 20 points) Implement a version of PLA by visiting examples in fixed, pre-determined random cycles throughout the algorithm. Run the algorithm on the data set. Please repeat your experiment for 1126 times, each with a different random seed. What is the average number of updates before the algorithm halts? Plot a histogram (`https://en.wikipedia.org/wiki/Histogram`) to show the number of updates versus the frequency of the number.

Bonus: Another Perceptron Learning Algorithm

The original perceptron learning algorithm does not take the "seriousness" of the prediction error into account. That is, regardless of whether $y_{n(t)}\mathbf{w}_t^T\mathbf{x}_{n(t)}$ is very negative or just slightly negative, the update rule is always

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_{n(t)}\mathbf{x}_{n(t)}.$$

Dr. Learn decides to use a different update rule. Namely, if $y_{n(t)} \neq \text{sign}(\mathbf{w}_t^T\mathbf{x}_{n(t)})$, the doctor will use

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_{n(t)}\mathbf{x}_{n(t)} \cdot M.$$

where M is the smallest integer such that $y_{n(t)}\mathbf{w}_{t+1}^T\mathbf{x}_{n(t)} > 0$.

8. (20 bonus points) Prove that M exists by deriving its formula. Then, prove our disprove the following claim: *the new update rull still ensures halting with a "perfect line" when the data set is linear separable.*