Homework #1

RELEASE DATE: 09/25/2020

RED BUG FIX: 09/26/2020 11:20

BLUE BUG FIX: 09/28/2020 11:00

GREEN BUG FIX: 10/06/2020 17:00

DUE DATE: 10/16/2020, BEFORE 13:00 on NTU COOL

QUESTIONS ARE WELCOMED ON THE NTU COOL FORUM.

We will instruct you on how to use NTU COOL (or other platforms) to upload your choices and your scanned/printed solutions later. For problems marked with (*), please follow the guidelines on the course website and upload your source code to NTU COOL. 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 400 points. For each problem, there is one correct choice. For most of the problems, if you choose the correct answer, you get 20 points; if you choose an incorrect answer, you get -10 points. That is, the expected value of random guessing is -20 per problem, and if you can eliminate two of the choices accurately, the expected value of random guessing on the remaining three choices would be 0 per problem. For other problems, the TAs will check your solution in terms of the written explanations and/or code. The solution will be given points between $[-20, 20]$ based on how logical your solution is.

The Learning Problem

1. Which of the following problem is suited for machine learning if there is assumed to be enough associated data? Choose the correct answer; explain how you can possibly use machine learning to solve it.

[a] predicting the winning number of the next invoice lottery

[b] calculating the average score of 500 students

[c] identifying the exact minimal spanning tree of a graph

[d] ranking mango images by the quality of the mangoes

[e] none of the other choices

2. Which of the following describes an machine learning approach to build a system for spam detection? Choose the correct answer; explain briefly why you think other choices are *not* machine learning.

[a] flip 3 fair coins; classify the email as a spam iff at least 2 of them are heads

[b] forward the email to 3 humans; classify the email as a spam iff at least 2 of them believe so

[c] produce a list of words for spams by 3 humans; classify the email as a spam iff the email contains more than 10 words from the list

[d] get a data set that contains spams and non-spams, for all words in the data set, let the machine calculate the ratio of spams per word; produce a list of words that appear more than 5 times and are of the highest 20% ratio; classify the email as a spam iff the email contains more than 10 words from the list

[e] get a data set that contains spams and non-spams, for all words in the data set, let the machine decide its "spam score"; sum the score up for each email; let the machine optimize a threshold that achieves the best precision of spam detection; classify the email as a spam iff the email is of score more than the threshold

Perceptron Learning Algorithm

Next, we will play with multiple variations to the Perceptron Learning Algorithm (PLA).

3. Dr. Short scales down all $\mathbf{x}_n$ (including the x_0 within) linearly by a factor of 4 before running PLA. How does the worst-case speed of PLA (in terms of the bound on page 16 of lecture 2) change after scaling? Choose the correct answer; explain your answer.

[a] 4 times smaller (i.e. faster)

[b] 2 times smaller

[c] $\sqrt{2}$ times smaller

[d] unchanged

[e] $\sqrt{2}$ times larger (i.e. slower)

4. The scaling in the previous problem is equivalent to inserting a "learning rate" η to the PLA update rule

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

with $\eta = \frac{1}{4}$. In fact, we do not need to use a fixed η. Let η_t denote the learning rate in the t-th iteration; that is, let PLA update $\mathbf{w}_t$ by

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

whenever $(\mathbf{x}_{n(t)}, y_{n(t)})$ is not correctly classified by $\mathbf{w}_t$. Dr. Adaptive decides to set $\eta_t = \frac{1}{\|\mathbf{x}_{n(t)}\|}$ so "longer" $\mathbf{x}_{n(t)}$ will not affect $\mathbf{w}_t$ too much. Let

$$\hat{\rho} = \min_{n \in \{1,2,\cdots,N\}} \frac{|\mathbf{w}_f^T \mathbf{x}_n|}{\|\mathbf{w}_f\|\|\mathbf{x}_n\|},$$

which can be viewed as a "normalized" version of the ρ on page 16 of lecture 2. The bound on the same page then becomes $\hat{\rho}^{-p}$ after using this adaptive η_t. What is p? Choose the correct answer; explain your answer.

[a] 0

[b] 1

[c] 2

[d] 4

[e] 8

5. Another possibility of setting η_t is to consider how negative $y_{n(t)}\mathbf{w}_t^T\mathbf{x}_{n(t)}$ is, and try to make $y_{n(t)}\mathbf{w}_{t+1}^T\mathbf{x}_{n(t)} > 0$; that is, let $\mathbf{w}_{t+1}$ correctly classify $(\mathbf{x}_{n(t)}, y_{n(t)})$. Which of the following update rules make $y_{n(t)}\mathbf{w}_{t+1}^T\mathbf{x}_{n(t)} > 0$? Choose the correct answer; explain your answer.

[a] $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + 2 \cdot y_{n(t)}\mathbf{x}_{n(t)}$

[b] $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + 0.1126 \cdot y_{n(t)}\mathbf{x}_{n(t)}$

[c] $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_{n(t)}\mathbf{x}_{n(t)} \cdot \left(\frac{-y_{n(t)}\mathbf{w}_t^T\mathbf{x}_{n(t)}}{\|\mathbf{x}_{n(t)}\|^2}\right)$

[d] $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_{n(t)}\mathbf{x}_{n(t)} \cdot \left\lfloor \frac{-y_{n(t)}\mathbf{w}_t^T\mathbf{x}_{n(t)}}{\|\mathbf{x}_{n(t)}\|^2} + 1 \right\rfloor$

[e] $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - y_{n(t)}\mathbf{x}_{n(t)} \cdot \left\lfloor \frac{-y_{n(t)}\mathbf{w}_t^T\mathbf{x}_{n(t)}}{\|\mathbf{x}_{n(t)}\|^2} + 1 \right\rfloor$

6. Dr. Separate decides to use one of the update rules in the previous problem for PLA. When the data set is linear separable, how many choices in the previous problem ensures halting with a "perfect line"? Choose the correct answer; explain the reason behind each halting case.

[a] 1

[b] 2

[c] 3

[d] 4

[e] 5

Types of Learning

7. One shared technique between the famous AlphaGo, AlphaGo Zero, and AlphaStar is called self-practicing: learning to play the game by practicing with itself and getting the feedback from the "judge" environment. What best describes the learning problem behind self-practicing? Choose the correct answer; explain your answer.

[a] human learning

[b] unsupervised learning

[c] semi-supervised learning

[d] supervised learning

[e] reinforcement learning

8. Consider formulating a learning problem for building a self-driving car. First, we gather a training data set that consists of 100 hours of video that contains the view in front of a car, and records about how the human behind the wheel acted with physically constrained choices like steering, braking, and signaling-before-turning. We also gather another 100 hours of videos from 1126 more cars without the human records. The learning algorithm is expected to learn from all the videos to obtain a hypothesis that imitates the human actions well. What learning problem best matches the description above? Choose the correct answer; explain your answer.

[a] regression, unsupervised learning, active learning, concrete features

[b] structured learning, semi-supervised learning, batch learning, raw features

[c] structured learning, supervised learning, batch learning, concrete features

[d] regression, reinforcement learning, batch learning, concrete features

[e] structured learning, supervised learning, online learning, concrete features

(*We are definitely not hinting that you should build a self-driving car this way. :-D*)

Off-Training-Set Error

As discussed on page 5 of lecture 4, what we really care about is whether $g \approx f$ *outside* $\mathcal{D}$. For a set of "universe" examples $\mathcal{U}$ with $\mathcal{D} \subset \mathcal{U}$, the error *outside* $\mathcal{D}$ is typically called the Off-Training-Set (OTS) error

$$E_{\text{ots}}(h) = \frac{1}{|\mathcal{U} \setminus \mathcal{D}|} \sum_{(\mathbf{x},y) \in \mathcal{U} \setminus \mathcal{D}} [\![h(\mathbf{x}) \neq y]\!] .$$

9. Consider $\mathcal{U}$ with 6 examples

$\mathbf{x}$	y
$(1, 0)$,	$+1$
$(3, 2)$,	$+1$
$(0, 2)$,	$+1$
$(2, 3)$,	-1
$(2, 4)$,	-1
$(3, 5)$,	-1

Run the process of choosing any three examples from $\mathcal{U}$ as $\mathcal{D}$, and learn a perceptron hypothesis (say, with PLA, or any of your "human learning" algorithm) to achieve $E_{\text{in}}(g) = 0$ on $\mathcal{D}$. Then, evaluate g outside $\mathcal{D}$. What is the smallest and largest $E_{\text{ots}}(g)$? Choose the correct answer; explain your answer.

[a] $(0, \frac{1}{3})$
[b] $(0, \frac{2}{3})$
[c] $(\frac{2}{3}, 1)$
[d] $(\frac{1}{3}, 1)$
[e] $(0, 1)$

Hoeffding Inequality

10. Suppose you are given a biased coin with one side coming up with probability $\frac{1}{2} + \epsilon$. How many times do you need to toss the coin to find out the more probable side with probability at least $1 - \delta$ using the Hoeffding's Inequality mentioned in page 10 of lecture 4? Choose the correct answer; explain your answer. (*Hint: There are multiple versions of Hoeffding's inequality. Please use the version in the lecture, albeit slightly loose, for answering this question. The* log *here is* $\log_e$.)

[a] $\frac{1}{2\epsilon^2\delta} \log 2$
[b] $\frac{1}{2\epsilon^2} \log \frac{2}{\delta}$
[c] $\frac{1}{2\epsilon} \log \frac{2}{\epsilon\delta}$
[d] $\frac{1}{2} \log \frac{2}{\epsilon^2\delta}$
[e] $\log \frac{1}{\epsilon^2\delta}$

Bad Data

11. Consider $\mathbf{x} = [x_1, x_2]^T \in \mathbb{R}^2$, a target function $f(\mathbf{x}) = \text{sign}(x_1)$, a hypothesis $h_1(\mathbf{x}) = \text{sign}(2x_1 - x_2)$, and another hypothesis $h_2(\mathbf{x}) = \text{sign}(x_2)$. When drawing 5 examples independently and uniformly within $[-1, +1] \times [-1, +1]$ as $\mathcal{D}$, what is the probability that we get 5 examples $(\mathbf{x}_n, f(\mathbf{x}_n))$ such that $E_{\text{in}}(h_2) = 0$? Choose the correct answer; explain your answer. (*Note: This is one of the BAD-data cases for h_2 where $E_{\text{in}}(h_2)$ is far from $E_{\text{out}}(h_2)$.*)

[a] 0
[b] $\frac{1}{5}$
[c] $\frac{1}{32}$
[d] $\frac{1}{1024}$
[e] 1

12. Following the setting of the previous problem, what is the probability that we get 5 examples such that $E_{\text{in}}(h_2) = E_{\text{in}}(h_1)$, including both the zero and non-zero E_{in} cases? Choose the correct answer; explain your answer. (*Note: This is one of the BAD-data cases where we cannot distinguish the better-E_{out} hypothesis h_1 from the worse hypothesis h_2.*)

[a] $\frac{243}{32768}$
[b] $\frac{1440}{32768}$
[c] $\frac{2160}{32768}$
[d] $\frac{3843}{32768}$
[e] $\frac{7776}{32768}$

13. According to page 22 of lecture 4, for a hypothesis set $\mathcal{H}$,

$$\text{BAD } \mathcal{D} \text{ for } \mathcal{H} \iff \exists h \in \mathcal{H} \text{ s.t. } |E_{\text{out}}(h) - E_{\text{in}}(h)| > \epsilon.$$

Let $\mathbf{x} = [x_1, x_2, \cdots, x_d]^T \in \mathbb{R}^d$ with $d > 1$. Consider a binary classification target with $\mathcal{Y} = \{+1, -1\}$ and a hypothesis set $\mathcal{H}$ with $2d$ hypotheses $h_1, \cdots, h_{2d}$.

- For $i = 1, \cdots, d$, $h_i(\mathbf{x}) = \text{sign}(x_i)$.
- For $i = d+1, \cdots, 2d$, $h_i(\mathbf{x}) = -\text{sign}(x_{i-d})$.

Extend the Hoeffding's Inequality mentioned in page 10 of lecture 4 with a proper union bound. Then, for any given N and ϵ, what is the smallest C that makes this inequality true?

$$\mathbb{P}[\text{BAD } \mathcal{D} \text{ for } \mathcal{H}] \leq C \cdot 2\exp(-2\epsilon^2 N).$$

Choose the correct answer; explain your answer.

[a] $C = 1$
[b] $C = d$
[c] $C = 2d$
[d] $C = 4d$
[e] $C = \infty$

Multiple-Bin Sampling

14. We then illustrate what happens with multiple-bin sampling with an experiment that use a dice (instead of a marble) to bind the six faces together. Please note that the dice is not meant to be thrown for random experiments. The probability below only refers to drawing the dices from the bag. Try to view each number as a *hypothesis*, and each dice as an *example* in our multiple-bin scenario. You can see that no single number is always green—that is, E_{out} of each hypothesis is always non-zero. In the next two problems, we are essentially asking you to calculate the probability of getting $E_{\text{in}}(h_3) = 0$, and the probability of the minimum $E_{\text{in}}(h_i) = 0$.

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

- A: all even numbers are colored green, all odd numbers are colored orange
- B: $(2, 3, 4)$ are colored green, others are colored orange
- C: the number 6 is colored green, all other numbers are colored orange
- D: all primes are colored green, others are colored orange

If we draw 5 dices independently from the bag, which combination is of the same probability as getting five green 3's? Choose the correct answer; explain your answer.

[a] five green 1's

[b] five orange 2's

[c] five green 2's

[d] five green 4's

[e] five green 5's

15. Following the previous problem, if we draw 5 dices independently from the bag, what is the probability that we get *some number* that is purely green? Choose the correct answer; explain your answer.

[a] $\frac{512}{1024}$

[b] $\frac{333}{1024}$

[c] $\frac{274}{1024}$

[d] $\frac{243}{1024}$

[e] $\frac{32}{1024}$

Experiments with Perceptron Learning Algorithm

Next, we use an artificial data set to study PLA. The data set with $N = 100$ examples is in

`http://www.csie.ntu.edu.tw/~htlin/course/ml20fall/hw1/hw1_train.dat`

Each line of the data set contains one $(\mathbf{x}_n, y_n)$ with $\mathbf{x}_n \in \mathbb{R}^{10}$. The first 10 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.

16. (*) Please first follow page 4 of lecture 2, and add $x_0 = 1$ to every $\mathbf{x}_n$. Implement a version of PLA that randomly picks an example $(\mathbf{x}_n, y_n)$ in every iteration, and updates $\mathbf{w}_t$ if and only if $\mathbf{w}_t$ is incorrect on the example. Note that the random picking can be simply implemented *with replacement*—that is, the same example can be picked multiple times, even consecutively. Stop updating and return $\mathbf{w}_t$ as $\mathbf{w}_{\text{PLA}}$ if $\mathbf{w}_t$ is correct consecutively after checking $5N$ randomly-picked examples.

Hint: (1) The update procedure described above is equivalent to the procedure of gathering all the incorrect examples first and then randomly picking an example among the incorrect ones. But the description above is usually much easier to implement. (2) The stopping criterion above is a randomized, more efficient implementation of checking whether $\mathbf{w}_t$ makes no mistakes on the data set.

Repeat your experiment for 1000 times, each with a different random seed. What is the median number of updates before the algorithm returns $\mathbf{w}_{\text{PLA}}$? Choose the closest value.

[a] 8
[b] 11
[c] 14
[d] 17
[e] 20

17. (*) Among all the w_0 (the zero-th component of $\mathbf{w}_{\text{PLA}}$) obtained from the 1000 experiments above, what is the median? Choose the closest value.

[a] -10
[b] -5
[c] 0
[d] 5
[e] 10

18. (*) Set $x_0 = 10$ to every $\mathbf{x}_n$ instead of $x_0 = 1$, and repeat the 1000 experiments above. What is the median number of updates before the algorithm returns $\mathbf{w}_{\text{PLA}}$? Choose the closest value.

[a] 8
[b] 11
[c] 14
[d] 17
[e] 20

19. (*) Set $x_0 = 0$ to every $\mathbf{x}_n$ instead of $x_0 = 1$. This equivalently means not adding any x_0, and you will get a separating hyperplane that passes the origin. Repeat the 1000 experiments above. What is the median number of updates before the algorithm returns $\mathbf{w}_{\text{PLA}}$?

[a] 8
[b] 11
[c] 14
[d] 17
[e] 20

20. (*) Now, in addition to setting $x_0 = 0$ to every $\mathbf{x}_n$, scale down each $\mathbf{x}_n$ by 4. Repeat the 1000 experiments above. What is the median number of updates before the algorithm returns $\mathbf{w}_{\text{PLA}}$? Choose the closest value.

[a] 8
[b] 11
[c] 14
[d] 17
[e] 20