Homework #1

RELEASE DATE: 12/01/2016

DUE DATE: 12/15/2016, 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. Which of the following problems are best suited for machine learning?

(i) Classifying numbers into primes and non-primes

(ii) Detecting potential fraud in credit card charges

(iii) Determining the time it would take a falling object to hit the ground

(iv) Determining the optimal cycle for traffic lights in a busy intersection

(v) Determining the age at which a particular medical test is recommended

Please provide explanation of your choices.

For Problems 2-5, identify the best type of learning that can be used to solve each task below. Suggested choices include supervised learning, unsupervised learning, active learning, and reinforcement learning. But you can put in other choices as long as your explanations are reasonable.

2. Play chess better by practicing different strategies and receive outcome as feedback. Please provide explanation of your choice.

3. Categorize books into groups without given topics. Please provide explanation of your choice.

4. Recognize whether there is a face in the picture by a thousand face pictures and ten thousand non-face pictures. Please provide explanation of your choice.

5. Selectively schedule experiments on mice to quickly evaluate the potential of cancer medicines. Please provide explanation of your choice.

Problem 6-8 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

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

6. Consider $f(\mathbf{x}) = +1$ for all $\mathbf{x}$ and $g(\mathbf{x}) = \begin{cases} +1, & \text{for } \mathbf{x} = \mathbf{x}_k \text{ and } k \text{ is odd} \\ -1, & \text{otherwise} \end{cases}$ and $1 \leq k \leq N + L$. $E_{OTS}(g, f) =$? Please provide proof of your answer.

7. We say that a target function f can "generate" $\mathcal{D}$ in a noiseless setting if $f(\mathbf{x}_n) = y_n$ for all $(\mathbf{x}_n, y_n) \in \mathcal{D}$. For all possible $f\colon \mathcal{X} \to \mathcal{Y}$, how many of them can generate $\mathcal{D}$ in a noiseless setting? Note that we call two functions f_1 and f_2 the same if $f_1(\mathbf{x}) = f_2(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$. Please provide proof of your answer.

8. 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 Problems 9-12, 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.

9. If $\mu = 0.5$, what is the probability of $\nu = \mu$? Please provide calculating steps of your answer.

10. If $\mu = 0.8$, what is the probability of $\nu = \mu$? Please provide calculating steps of your answer.

11. If $\mu = 0.8$, what is the probability of $\nu \leq 0.1$? Please provide calculating steps of your answer.

12. If $\mu = 0.8$, what is the bound given by Hoeffding's Inequality for the probability of $\nu \leq 0.1$? Please provide calculating steps of your answer, and use the two-sided inequality provided in class.

Problems 13-14 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

13. 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.

14. 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.

For Problems 15-20, you will play with PLA and pocket algorithm.

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

`http://www.csie.ntu.edu.tw/~htlin/course/mlfound16fall/hw1/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 $\mathbf{w} = 0$ and take sign(0) as -1. As a friendly reminder, remember to add $x_0 = 1$ as always!

15. (*) Implement a version of PLA by visiting examples in the naïve cycle using the order of examples in the data set. Run the algorithm on the data set. What is the number of updates before the algorithm halts? What is the index of the example that results in the most number of updates?

16. (*) 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 2000 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 frequency.

17. (*) Implement a version of PLA by visiting examples in fixed, pre-determined random cycles throughout the algorithm, while changing the update rule to be

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

with $\eta = 0.25$. Note that your PLA in the previous problem corresponds to $\eta = 1$. Please repeat your experiment for 2000 times, each with a different random seed. What is the average number of updates before the algorithm halts? Plot a histogram to show the number of updates versus frequency. Compare your result to the previous problem and briefly discuss your findings.

Next, we play with the pocket algorithm. Modify your PLA in Problem 16 to visit examples purely randomly, and then add the 'pocket' steps to the algorithm. We will use

`http://www.csie.ntu.edu.tw/~htlin/course/mlfound16fall/hw1/hw1_18_train.dat`

as the training data set $\mathcal{D}$, and

`http://www.csie.ntu.edu.tw/~htlin/course/mlfound16fall/hw1/hw1_18_test.dat`

as the test set for "verifying" the g returned by your algorithm (see lecture 4 about verifying). The sets are of the same format as the previous one.

18. (*) Run the pocket algorithm with a total of 50 updates on $\mathcal{D}$, and verify the performance of $\mathbf{w}_{\text{POCKET}}$ using the test set. Please repeat your experiment for 2000 times, each with a different random seed. What is the average error rate on the test set? Plot a histogram to show error rate versus frequency.

19. (*) Modify your algorithm in Problem 18 to run for 100 updates instead of 50, and verify the performance of $\mathbf{w}_{\text{POCKET}}$ using the test set. Please repeat your experiment for 2000 times, each with a different random seed. What is the average error rate on the test set? Plot a histogram to show error rate versus frequency. Compare your result to Problem 18 and briefly discuss your findings.

20. (*) Modify your algorithm in Problem 19 to return $\mathbf{w}_{100}$ (the PLA vector after 50 updates) instead of $\hat{\mathbf{w}}$ (the pocket vector) after 100 updates. Run the modified algorithm on $\mathcal{D}$, and verify the performance using the test set. Please repeat your experiment for 2000 times, each with a different random seed. What is the average error rate on the test set? Plot a histogram to show error rate versus frequency. Compare your result to Problem 19 and briefly discuss your findings.

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

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

21. (10 points) Prove that with the new update rule, $y_{n(t)}\mathbf{w}_{t+1}^T \mathbf{x}_{n(t)} > 0$. That is, $\mathbf{w}_{t+1}$ always classifies $(\mathbf{x}_{n(t)}, y_{n(t)})$ correctly.

22. (10 points) When the data set is linear separable, does this new update rule still ensure halting with a "perfect line"? Why or why not?