

Homework #1

RELEASE DATE: 10/01/2014

DUE DATE: 10/15/2014, BEFORE NOON

QUESTIONS ABOUT HOMEWORK MATERIALS ARE WELCOMED ON THE COURSERA 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.

There are two kinds of regular problems.

- *multiple-choice question (MCQ): There are several choices and **only one of them is correct**. You should choose one and only one.*
- *multiple-response question (MRQ): There are several choices and **none, some, or all of them are correct**. You should write down every choice that you think to be correct.*

Some problems also come with (+ ...) that contains additional todo items.

If there are big bonus questions (BBQ), please simply follow the problem guideline to write down your solution, if you choose to tackle them.

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. (MCQ) 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

[a] (ii), (iv), and (v)

[b] (i) and (ii)

[c] (i), (ii), (iii), and (iv)

[d] (iii) and (iv)

[e] none of the other choices

(+ explanation of your choice)

For Problems 2-5, identify the best type of learning that can be used to solve each task below.

2. (MCQ) Improve the recommendation of books to each user gradually from daily sales data
- [a] supervised learning
 - [b] reinforcement learning
 - [c] active learning
 - [d] unsupervised learning
 - [e] none of the other choices
- (+ explanation of your choice)
3. (MCQ) Schedule experiments on mice to model the effectiveness of potential cancer medicines
- [a] supervised learning
 - [b] reinforcement learning
 - [c] active learning
 - [d] unsupervised learning
 - [e] none of the other choices
- (+ explanation of your choice)
4. (MCQ) Categorize books into groups without given topics
- [a] supervised learning
 - [b] reinforcement learning
 - [c] active learning
 - [d] unsupervised learning
 - [e] none of the other choices
- (+ explanation of your choice)
5. (MCQ) Decide the maximum allowed debt for each bank customer
- [a] supervised learning
 - [b] reinforcement learning
 - [c] active learning
 - [d] unsupervised learning
 - [e] none of the other choices
- (+ explanation of your choice)

Problem 6-8 are about *Off-Training-Set error*.

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

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

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

- [a] $\frac{1}{L} \times (\lceil \frac{N+L}{2} \rceil - \lfloor \frac{N}{2} \rfloor)$
- [b] $\frac{1}{L} \times (\lceil \frac{N+L}{2} \rceil - \lceil \frac{N}{2} \rceil)$
- [c] $\frac{1}{L} \times (\lfloor \frac{N+L}{2} \rfloor - \lceil \frac{N}{2} \rceil)$
- [d] $\frac{1}{L} \times (\lfloor \frac{N+L}{2} \rfloor - \lfloor \frac{N}{2} \rfloor)$
- [e] none of the other choices

(+ proof of your choice)

7. (MCQ) 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 $f: \mathcal{X} \rightarrow \mathcal{Y}$, 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 $\mathbf{x} \in \mathcal{X}$.

- [a] 1
- [b] 2^N
- [c] 2^L
- [d] 2^{N+L}
- [e] none of the other choices

(+ proof of your choice)

8. (MRQ) Following the previous question, select all the correct statements below.

- [a] For a fixed g and a given integer k between 0 and L , the number of f that generates \mathcal{D} and satisfies $E_{OTS}(g, f) = \frac{k}{L}$ is $\binom{L}{k}$.
- [b] For a fixed g and a given integer k between 0 and L , the number of f that generates \mathcal{D} and satisfies $E_{OTS}(g, f) = \frac{k}{L}$ is $\binom{L+N}{k}$.
- [c] For a fixed g , if all those f that can “generate” \mathcal{D} in a noiseless setting are equally likely in probability, the expected OTS $\mathbb{E}_f\{E_{OTS}(g, f)\}$ is $\frac{1}{2}$.
- [d] For a fixed g , if all those f that can “generate” \mathcal{D} in a noiseless setting are equally likely in probability, the expected OTS $\mathbb{E}_f\{E_{OTS}(g, f)\}$ cannot be computed.
- [e] A deterministic 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,

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

(+ proof of your choices)

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. (MCQ) If $\mu = 0.5$, what is the probability of $\nu = \mu$?

- [a] 0.24
- [b] 0.39
- [c] 0.56
- [d] 0.90
- [e] none of the other choices

(+ calculating step of your choice)

10. (MCQ) If $\mu = 0.9$, what is the probability of $\nu = \mu$?

- [a] 0.24
- [b] 0.39
- [c] 0.56
- [d] 0.90
- [e] none of the other choices

(+ calculating step of your choice)

11. (MCQ) If $\mu = 0.9$, what is the probability of $\nu \leq 0.1$?

- [a] 0.1×10^{-9}
- [b] 1.0×10^{-9}
- [c] 8.5×10^{-9}
- [d] 9.1×10^{-9}
- [e] none of the other choices

(+ calculating step of your choice)

12. (MCQ) If $\mu = 0.9$, what is the bound given by Hoeffding's Inequality for the probability of $\nu \leq 0.1$?

- [a] 5.52×10^{-4}
- [b] 5.52×10^{-6}
- [c] 5.52×10^{-8}
- [d] 5.52×10^{-10}
- [e] none of the other choices

(+ calculating step of your choice)

Problems 13-14 illustrate what happens with multiple bins.

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. (MCQ) If we pick 5 dice from the bag, what is the probability that we get five orange 1's?

- [a] $\frac{1}{256}$
- [b] $\frac{8}{256}$
- [c] $\frac{31}{256}$
- [d] $\frac{46}{256}$
- [e] none of the other choices

(+ calculating step of your choice)

14. (MCQ) If we pick 5 dice from the bag, what is the probability that we get "some number" that is purely orange?

- [a] $\frac{1}{256}$
- [b] $\frac{8}{256}$
- [c] $\frac{31}{256}$

[d] $\frac{46}{256}$

[e] none of the other choices

(+ calculating step of your choice)

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/ml13fall/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 $\text{sign}(0)$ as -1

15. (MCQ, *) 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?

[a] < 10 updates

[b] 11 - 30 updates

[c] 31 - 50 updates

[d] 51 - 200 updates

[e] ≥ 201 updates

16. (MCQ, *) 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?

[a] < 10 updates

[b] 11 - 30 updates

[c] 31 - 50 updates

[d] 51 - 200 updates

[e] ≥ 201 updates

17. (MCQ, *) 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 $\eta = 0.5$. 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?

[a] < 10 updates

[b] 11 - 30 updates

[c] 31 - 50 updates

[d] 51 - 200 updates

[e] ≥ 201 updates

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/ml13fall/hw1/hw1_18_train.dat

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

http://www.csie.ntu.edu.tw/~htlin/course/ml13fall/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.** (MCQ, *) 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?

- [a] < 0.2
- [b] $0.2 - 0.4$
- [c] $0.4 - 0.6$
- [d] $0.6 - 0.8$
- [e] ≥ 0.8

- 19.** (MCQ, *) Modify your algorithm in Problem 18 to return \mathbf{w}_{50} (the PLA vector after 50 updates) instead of $\hat{\mathbf{w}}$ (the pocket vector) after 50 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?

- [a] < 0.2
- [b] $0.2 - 0.4$
- [c] $0.4 - 0.6$
- [d] $0.6 - 0.8$
- [e] ≥ 0.8

- 20.** (MCQ, *) 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?

- [a] < 0.2
- [b] $0.2 - 0.4$
- [c] $0.4 - 0.6$
- [d] $0.6 - 0.8$
- [e] ≥ 0.8

Bonus: More about Proof of Perceptron Learning Algorithm

- 21.** (BBQ, 20 points) Page 16 in lecture 2 suggests that the radius R affects the convergence of PLA. So Dr. Learn plans to conduct the following procedure: scale down all \mathbf{x}_n linearly by a factor of 10, with the hope that the PLA algorithm would run 10 times faster. Will his plan work? Why or why not?

Answer guidelines. First, please write down your name and school ID number.

Name:	School ID:
-------	------------

Then, fill in your answers for MCQ and MRQ in the table below.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20

Lastly, please write down your solution to those (+ ...) parts and bonus problems, using as many additional pages as you want.

Each problem is of 10 points.

- For Problem with (+ ...), the answer in the table is of 3 score points, and the (+ ...) part is of 7 score points. If your solution to the (+ ...) part is clearly different from your answer in the table, it is regarded as a suspicious violation of the class policy (plagiarism) and the TAs can deduct some more points based on the violation.
- For Problem without (+ ...), the problem is of 10 points by itself and the TAs can decide to give you partial credit or not as long as it is fair to the whole class.