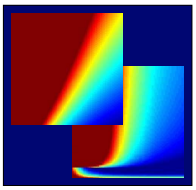
CS/CNS/EE 156a Learning Systems
Caltech - Fall 2019
http://cs156.caltech.edu
(Learning From Data campus version)



Homework # 1

Due Monday, October 7, 2019, at 2:00 PM PDT

*Definitions and notation follow the lectures. All questions have multiple-choice answers (***[a], [b], [c], ...***). Collaboration is allowed but without discussing selected or excluded choices. Your solutions must be based on your own work. See the initial "***Course Description and Policies***" handout for important details about collaboration and "open book" policies.*

Note about the homework

- Answer each question by deriving the answer (carries 6 points) then selecting from the multiple-choice answers (carries 4 points). You can select 1 or 2 of the multiple-choice answers for each question, but you will get 4 or 2 points, respectively, for a correct answer. See the initial "**Course Description and Policies**" handout for important details.
- The problems range from easy to difficult, and from practical to theoretical. Some problems require running a full experiment to arrive at the answer.
- The answer may not be obvious or numerically close to one of the choices, but one (and only one) choice will be correct if you follow the instructions precisely in each problem. You are encouraged to explore the problem further by experimenting with variations on these instructions, for the learning benefit.
- You are encouraged to take part in the discussion forum. Please make sure you don't discuss specific answers, or specific excluded answers, before the homework is due.

- **The Learning Problem**

1. What types of Machine Learning, if any, best describe the following three scenarios:

(i) A coin classification system is created for a vending machine. The developers obtain exact coin specifications from the U.S. Mint and derive a statistical model of the size, weight, and denomination, which the vending machine then uses to classify coins.

(ii) Instead of calling the U.S. Mint to obtain coin information, an algorithm is presented with a large set of labeled coins. The algorithm uses this data to infer decision boundaries which the vending machine then uses to classify its coins.

(iii) A computer develops a strategy for playing Tic-Tac-Toe by playing repeatedly and adjusting its strategy by penalizing moves that eventually lead to losing.

[a] (i) Supervised Learning, (ii) Unsupervised Learning, (iii) Reinforcement Learning

[b] (i) Supervised Learning, (ii) Not learning, (iii) Unsupervised Learning

[c] (i) Not learning, (ii) Reinforcement Learning, (iii) Supervised Learning

[d] (i) Not learning, (ii) Supervised Learning, (iii) Reinforcement Learning

[e] (i) Supervised Learning, (ii) Reinforcement Learning, (iii) Unsupervised Learning

2. 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.

[a] (ii) and (iv)

[b] (i) and (ii)

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

[d] (iii)

[e] (i) and (iii)

- **Bins and Marbles**

3. We have 2 opaque bags, each containing 2 balls. One bag has 2 black balls and the other has a black ball and a white ball. You pick a bag at random and then pick one of the balls in that bag at random. When you look at the ball, it is black. You now pick the second ball from that same bag. What is the probability that this ball is also black?

[a] 1/4

[b] 1/3

[c] 1/2

[d] 2/3

[e] 3/4

Consider a sample of 10 marbles drawn from a bin containing red and green marbles. The probability that any marble we draw is red is $\mu = 0.55$ (independently, with replacement). We address the probability of getting no red marbles ($\nu = 0$) in the following cases:

4. We draw only one such sample. Compute the probability that $\nu = 0$. The closest answer is ('closest answer' means: $|\text{your answer} - \text{given option}|$ is closest to 0):

[a] 7.331×10^{-6}

[b] 3.405×10^{-4}

[c] 0.289

[d] 0.450

[e] 0.550

5. We draw 1,000 independent samples. Compute the probability that (at least) one of the samples has $\nu = 0$. The closest answer is:

[a] 7.331×10^{-6}

[b] 3.405×10^{-4}

[c] 0.289

[d] 0.450

[e] 0.550

• **Feasibility of Learning**

Consider a Boolean target function over a 3-dimensional input space $\mathcal{X} = \{0, 1\}^3$ (instead of our ± 1 binary convention, we use 0,1 here since it is standard for Boolean functions). We are given a data set $\mathcal{D}$ of five examples represented in the table below, where $y_n = f(\mathbf{x}_n)$ for $n = 1, 2, 3, 4, 5$.

$\mathbf{x}_n$			y_n
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1

Note that in this simple Boolean case, we can enumerate the entire input space (since there are only $2^3 = 8$ distinct input vectors), and we can enumerate the set of all possible target functions (there are only $2^{2^3} = 256$ distinct Boolean function on 3 Boolean inputs).
Let us look at the problem of learning f. Since f is unknown except inside $\mathcal{D}$, any function that agrees with $\mathcal{D}$ could conceivably be f. Since there are only 3 points in $\mathcal{X}$ outside $\mathcal{D}$, there are only $2^3 = 8$ such functions.
The remaining points in $\mathcal{X}$ which are not in $\mathcal{D}$ are: 101, 110, and 111. We want to determine the hypothesis that agrees the most, *on these 3 points*, with the possible target functions. In order to quantify this, count how many of the 8 possible target functions agree with each hypothesis on all 3 points, how many agree on just 2 of the points, on just 1 point, and how many do not agree on any points. How a hypothesis agrees with the target function in sample (on $\mathcal{D}$ itself) has no bearing on its score:

Score = *(# of target functions agreeing with hypothesis on all 3 points)*×3 + *(# of target functions agreeing with hypothesis on exactly 2 points)*×2 + *(# of target functions agreeing with hypothesis on exactly 1 point)*×1 + *(# of target functions agreeing with hypothesis on 0 points)*×0.

6. Which hypothesis g agrees the most with the possible target functions in terms of the above score?

[a] g returns 1 for all three points.

[b] g returns 0 for all three points.

[c] g is the XOR function applied to $\mathbf{x}$, i.e., if the number of 1s in $\mathbf{x}$ is odd, g returns 1; if it is even, g returns 0.

[d] g returns the opposite of the XOR function: if the number of 1s is odd, it returns 0, otherwise returns 1.

[e] They are all equivalent (equal scores for g in **[a]** through **[d]**).

• **The Perceptron Learning Algorithm**

In this problem, you will create your own target function f and data set $\mathcal{D}$ to see how the Perceptron Learning Algorithm works. Take $d = 2$ so you can visualize the problem, and assume $\mathcal{X} = [-1, 1] \times [-1, 1]$ with uniform probability of picking each $\mathbf{x} \in \mathcal{X}$.

In each run, choose a random line in the plane as your target function f (do this by taking two random, uniformly distributed points in $[-1, 1] \times [-1, 1]$ and taking the line passing through them), where one side of the line maps to $+1$ and the other maps to -1. Choose the inputs $\mathbf{x}_n$ of the data set as random points (uniformly in $\mathcal{X}$), and evaluate the target function on each $\mathbf{x}_n$ to get the corresponding output y_n.

Now, in each run, use the Perceptron Learning Algorithm to find g. Start the PLA with the weight vector $\mathbf{w}$ being all zeros (consider $\text{sign}(0) = 0$, so all points are initially misclassified), and at each iteration have the algorithm choose a point randomly from the set of misclassified points. We are interested in two quantities: the number of iterations that PLA takes to converge to g, and the disagreement between f and g which is $\mathbb{P}[f(\mathbf{x}) \neq g(\mathbf{x})]$ (the probability that f and g will disagree on their classification of a random point). You can either calculate this probability exactly, or approximate it by generating a sufficiently large, separate set of points to estimate it.

In order to get a reliable estimate for these two quantities, you should repeat the experiment for 1000 runs (each run as specified above) and take the average over these runs.

7. Take $N = 10$. How many iterations does it take on average for the PLA to converge for $N = 10$ training points? Pick the value closest to your results (again, 'closest' means: |your answer − given option| is closest to 0).

[a] 1

[b] 15

[c] 300

[d] 5000

[e] 10000

8. Which of the following is closest to $\mathbb{P}[f(\mathbf{x}) \neq g(\mathbf{x})]$ for $N = 10$?

[a] 0.001

[b] 0.01

[c] 0.1

[d] 0.5

[e] 0.8

9. Now, try $N = 100$. How many iterations does it take on average for the PLA to converge for $N = 100$ training points? Pick the value closest to your results.

[a] 50

[b] 100

[c] 500

[d] 1000

[e] 5000

10. Which of the following is closest to $\mathbb{P}[f(\mathbf{x}) \neq g(\mathbf{x})]$ for $N = 100$?

[a] 0.001

[b] 0.01

[c] 0.1

[d] 0.5

[e] 0.8