Online Homework # 1

DUE TUESDAY, April 10, 2012, AT 10:00 AM PDT (5:00 PM GMT)

Collaboration in the sense of discussions is allowed, but you should NOT discuss your selected answers with anyone. Books and notes can be consulted. All questions will have multiple choice answers, the choices being labeled **[a]**, **[b]**, **[c]**, **[d]**, **[e]**. *Most questions will have 5 possible choices, although some may have less. You should enter your solutions online by logging into your account at the course web site.*

The Learning Problem

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

(i) A coin classification system must be created for a vending machine. In order to do this, 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 its 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 the learning approach?

(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 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 that has 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:

- **[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 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 with just 2 of the points, with just 1, and how many do not agree on any points. The final score for each hypothesis is computed as follows:

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

6. Which hypothesis g agrees the most the 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 1's 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 number of 1's is odd, it returns 0, otherwise returns 1.
- [**e**] They are all equivalent.

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 choose a random line in the plane as your target function f (do this by taking two random uniformly distributed points on $[-1, 1] \times [-1, 1]$ 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 in the plane, and evaluate the target function on each $\mathbf{x}_n$ to get the corresponding output y_n.

7. Take $\mathcal{X} = [-1, 1] \times [-1, 1]$ with uniform probability of picking each $\mathbf{x} \in \mathcal{X}$. Take $N = 10$. Run the Perceptron Learning Algorithm to find g and measure the difference between f and g as $\Pr(f(\mathbf{x}) \neq g(\mathbf{x}))$ (you can either calculate this exactly, or approximate it by generating a sufficiently large separate set of points to evaluate it). Repeat the experiment 1000 times and take the average. Start the PLA with the weight vector $\mathbf{w}$ being all zeros.

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.

[a] 1

[b] 15

[c] 300

[d] 5000

[e] 10000

8. Which of the following is closest to $\Pr(f(\mathbf{x}) \neq g(\mathbf{x}))$ 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] 1

[b] 15

[c] 300

[d] 5000

[e] 10000

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

[a] 0.001

[b] 0.01

[c] 0.1

[d] 0.5

[e] 0.8