Learning From Data
Yaser Abu-Mostafa, *Caltech*
http://work.caltech.edu/telecourse
Self-paced version

Homework # 2

*All questions have multiple-choice answers (***[a]***,* **[b]***,* **[c]***,* **...***). You can collaborate with others, but do not discuss the selected or excluded choices in the answers. You can consult books and notes, but not other people's solutions. Your solutions should be based on your own work. Definitions and notation follow the lectures.*

Note about the homework

- The goal of the homework is to facilitate a deeper understanding of the course material. The questions are not designed to be puzzles with catchy answers. They are meant to make you roll up your sleeves, face uncertainties, and approach the problem from different angles.

- 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 also encouraged to take part in the forum

  http://book.caltech.edu/bookforum

  where there are many threads about each homework set. We hope that you will contribute to the discussion as well. Please follow the forum guidelines for posting answers (see the "BEFORE posting answers" announcement at the top there).

- **Hoeffding Inequality**

Run a computer simulation for flipping 1,000 virtual fair coins. Flip each coin independently 10 times. Focus on 3 coins as follows: c_1 is the first coin flipped, c_{rand} is a coin chosen randomly from the 1,000, and $c_{\min}$ is the coin which had the minimum frequency of heads (pick the earlier one in case of a tie). Let ν_1, ν_{rand}, and $\nu_{\min}$ be the *fraction* of heads obtained for the 3 respective coins out of the 10 tosses.

Run the experiment 100,000 times in order to get a full distribution of ν_1, ν_{rand}, and $\nu_{\min}$ (note that c_{rand} and $c_{\min}$ will change from run to run).

1. The average value of $\nu_{\min}$ is closest to:

[**a**] 0

[**b**] 0.01

[**c**] 0.1

[**d**] 0.5

[**e**] 0.67

2. Which coin(s) has a distribution of ν that satisfies the (single-bin) Hoeffding Inequality?

[**a**] c_1 only

[**b**] c_{rand} only

[**c**] $c_{\min}$ only

[**d**] c_1 and c_{rand}

[**e**] $c_{\min}$ and c_{rand}

- **Error and Noise**

Consider the bin model for a hypothesis h that makes an error with probability μ in approximating a deterministic target function f (both h and f are binary functions). If we use the same h to approximate a noisy version of f given by:

$$P(y \mid \mathbf{x}) = \begin{cases} \lambda & y = f(x) \\ 1 - \lambda & y \neq f(x) \end{cases}$$

3. What is the probability of error that h makes in approximating y? *Hint: Two wrongs can make a right!*

[a] μ

[b] λ

[c] 1-μ

[d] $(1-\lambda) * \mu + \lambda * (1-\mu)$

[e] $(1-\lambda) * (1-\mu) + \lambda * \mu$

4. At what value of λ will the performance of h be independent of μ?

[a] 0

[b] 0.5

[c] $1/\sqrt{2}$

[d] 1

[e] No values of λ

- **Linear Regression**

In these problems, we will explore how Linear Regression for classification works. As with the Perceptron Learning Algorithm in Homework # 1, you will create your own target function f and data set $\mathcal{D}$. 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.

5. Take $N = 100$. Use Linear Regression to find g and evaluate E_{in}, the fraction of in-sample points which got classified incorrectly. Repeat the experiment 1000 times and take the average (keep the g's as they will be used again in Problem 6). Which of the following values is closest to the average E_{in}? (*Closest* is the option that makes the expression |your answer − given option| closest to 0. Use this definition of *closest* here and throughout.)

[a] 0

[b] 0.001

[c] 0.01

[d] 0.1

[e] 0.5

6. Now, generate 1000 fresh points and use them to estimate the out-of-sample error E_{out} of g that you got in Problem 5 (number of misclassified out-of-sample points / total number of out-of-sample points). Again, run the experiment 1000 times and take the average. Which value is closest to the average E_{out}?

[**a**] 0

[**b**] 0.001

[**c**] 0.01

[**d**] 0.1

[**e**] 0.5

7. Now, take $N = 10$. After finding the weights using Linear Regression, use them as a vector of initial weights for the Perceptron Learning Algorithm. Run PLA until it converges to a final vector of weights that completely separates all the in-sample points. Among the choices below, what is the closest value to the average number of iterations (over 1000 runs) that PLA takes to converge? (When implementing PLA, have the algorithm choose a point randomly from the set of misclassified points at each iteration)

[**a**] 1

[**b**] 15

[**c**] 300

[**d**] 5000

[**e**] 10000

- **Nonlinear Transformation**

In these problems, we again apply Linear Regression for classification. Consider the target function:

$$f(x_1, x_2) = \text{sign}(x_1^2 + x_2^2 - 0.6)$$

Generate a training set of $N = 1000$ points on $\mathcal{X} = [-1, 1] \times [-1, 1]$ with a uniform probability of picking each $\mathbf{x} \in \mathcal{X}$. Generate simulated noise by flipping the sign of the output in a randomly selected 10% subset of the generated training set.

8. Carry out Linear Regression without transformation, i.e., with feature vector:

$$(1, x_1, x_2),$$

to find the weight $\mathbf{w}$. What is the closest value to the classification in-sample error E_{in}? (Run the experiment 1000 times and take the average E_{in} to reduce variation in your results.)

[**a**] 0

[**b**] 0.1

[**c**] 0.3

[**d**] 0.5

[**e**] 0.8

9. Now, transform the $N = 1000$ training data into the following nonlinear feature vector:

$$(1, x_1, x_2, x_1x_2, x_1^2, x_2^2)$$

Find the vector $\tilde{\mathbf{w}}$ that corresponds to the solution of Linear Regression. Which of the following hypotheses is closest to the one you find? Closest here means agrees the most with your hypothesis (has the highest probability of agreeing on a randomly selected point). Average over a few runs to make sure your answer is stable.

[**a**] $g(x_1, x_2) = \text{sign}(-1 - 0.05x_1 + 0.08x_2 + 0.13x_1x_2 + 1.5x_1^2 + 1.5x_2^2)$

[**b**] $g(x_1, x_2) = \text{sign}(-1 - 0.05x_1 + 0.08x_2 + 0.13x_1x_2 + 1.5x_1^2 + 15x_2^2)$

[**c**] $g(x_1, x_2) = \text{sign}(-1 - 0.05x_1 + 0.08x_2 + 0.13x_1x_2 + 15x_1^2 + 1.5x_2^2)$

[**d**] $g(x_1, x_2) = \text{sign}(-1 - 1.5x_1 + 0.08x_2 + 0.13x_1x_2 + 0.05x_1^2 + 0.05x_2^2)$

[**e**] $g(x_1, x_2) = \text{sign}(-1 - 0.05x_1 + 0.08x_2 + 1.5x_1x_2 + 0.15x_1^2 + 0.15x_2^2)$

10. What is the closest value to the classification out-of-sample error E_{out} of your hypothesis from Problem 9? (Estimate it by generating a new set of 1000 points and adding noise, as before. Average over 1000 runs to reduce the variation in your results.)

[**a**] 0

[**b**] 0.1

[**c**] 0.3

[**d**] 0.5

[**e**] 0.8