Learning From Data
Caltech
http://work.caltech.edu/telecourse.html
2013

Online Homework # 2

*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 (***[a], [b], [c], ...***). You should enter your solutions online by logging into your account at the course web site.*

Note about the homeworks

- The goal of the homeworks 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 your sleeves, face uncertainties, and approach the problem from different angles.
- The problems range from easy to hard, and from theoretical to practical. 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. The intent is to prompt discussion and exchange of ideas.
- Speaking of discussion, you are encouraged to take part in the forum

  http://book.caltech.edu/bookforum

  where there are many threads about each homework. 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 reduce variability in your results (note that c_{rand} and $c_{\min}$ will change from run to run) and compute average values for the ν's.

1. $\nu_{\min}$ is closest to:

[**a**] 0

[**b**] 0.01

[**c**] 0.1

[**d**] 0.5

[**e**] 0.67

2. Which coin or coins satisfy Hoeffding's 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?

[**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

[**d**] All values of λ

[**e**] No values of λ

Linear Regression

In this problem, 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. Which is *closest* to the average E_{in}? (*closest* is the answer that makes the expression |your answer − given option| closest to 0)

[**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} (number of out-of-sample points misclassified / 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}? (use the same definition of *closest* as in 5)

[**a**] 0

[**b**] 0.001

[**c**] 0.01

[**d**] 0.1

[**e**] 0.5

7. Now, after finding the weights by using Linear Regression, use them as a vector of initial weights for the Perceptron Learning Algorithm, and run the PLA to find a final vector of weights that completely separates all the in-sample points. For $N = 10$ (average 1000 runs again), what is the closest value to the number of iterations it takes? (When implementing the PLA, assume that at each iteration the algorithm chooses a point randomly from the set of misclassified points).

[**a**] 1

[**b**] 15

[**c**] 300

[**d**] 5000

[**e**] 10000

Nonlinear Transformation

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 uniform probability of picking each $\mathbf{x} \in \mathcal{X}$. Generate simulated noise by flipping the sign of a random 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} in order to reduce variation in your results.

[a] 0

[b] 0.1

[c] 0.3

[d] 0.5

[e] 0.8

Now, transform the 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.

9. Which of the following hypotheses is closest to the one you find using Linear Regression on the transformed input? Closest here means agrees the most with your hypothesis (has the most probability of agreeing on a randomly selected point).

[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? (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