

# CS 156 Problem Set 2

Christopher Zhen

Oct 10, 2016

## Problem 1

(B) - The value of  $\nu_{min}$  was around 0.038 in my code. This makes sense because the minimum number of heads is usually 0.

## Problem 2

(D) - Only these two have  $\nu$  values fairly close to  $\mu$ .  $\nu_{rand}$  was 0.501 and  $\nu_1$  was 0.500 which are very close to  $\mu = 0.5$ . This makes sense because  $c_{rand}$  and  $c_1$  are pretty accurate samples of the population, but  $c_{min}$  is a very skewed sample.

## Problem 3

(E) - There are two possible ways to have an error: when  $y = f(x)$  and  $h$  gives an error or when  $y \neq f(x)$  and  $h = f$ . The first probability is  $\lambda * \mu$  and the second probability is  $(1 - \lambda) * (1 - \mu)$ .

## Problem 4

(B) - If  $\lambda$  is 0.5 then the expression simplifies to:

$$(1 - 0.5) * (1 - \mu) + 0.5 * \mu \tag{1}$$

$$0.5 * (1 - \mu + \mu) \tag{2}$$

Which simplifies to a probability of 0.5 and is independent of  $\mu$ .

## Problem 5

(C) - The value for  $E_{in}$  obtained in my code was 0.0368 which is closest to 0.01. This means linear regression is fairly accurate for classification, but definitely not as good as PLA which makes sense.

## Problem 6

(C) - This time,  $E_{out}$  was calculated to be 0.0419 which is closest to 0.01. This is slightly higher than  $E_{in}$  since this is not the data set we learned from, but it's also pretty low since the algorithm should do an okay job of learning  $f$ .

## Problem 7

(A) - When using the  $w$  determined from linear regression, PLA typically takes 4.811 iterations to converge. This makes sense because without using linear regression, PLA takes around 10-15 iterations to converge with 10 data points, so it should be considerably less if the initial guess is almost right.

## Problem 8

(D) - My code returned a value of  $E_{in}$  of 0.5128 which is closest to 0.5. This makes sense because we're trying to use a linear model to classify non-linear data so it'll probably result in a really bad fit.

## Problem 9

(A) - The value of  $\tilde{w}$  from the code was: [-1.01, 0.004, -0.003, -0.004, 1.59, 1.58]. This result was obtained by averaging 1000 iterations of linear regression and is most similar to answer choice A. This choice also makes sense because it is fairly close to the target function (which would be [-0.6, 0, 0, 0, 1, 1]).

## Problem 10

(B) - My code calculated  $E_{out}$  to be 0.0753 which is pretty close to 0.1.  $E_{in}$  was found to be around 0.0566. This is pretty impressive because it represents a significant increase over the original linear fitting algorithm.