CSE 417T: Homework 1

Due: Feburary 5, 2019, by 11:59 PM

Notes:

- Please check the submission instructions for Gradescope provided on the course website. You must follow those instructions exactly.
- Please complete and submit the following two stub Matlab files for Problem 2:
  `https://www.cse.wustl.edu/~sanmay/teaching/cse417-spring19/perceptron_experiment.m`
  `https://www.cse.wustl.edu/~sanmay/teaching/cse417-spring19/perceptron_learn.m`
- Homework is due **by 11:59 PM on the due date.** Remember that you may not use more than 2 late days on any one homework, and you only have a budget of 5 in total.
- Please keep in mind the collaboration policy as specified in the course syllabus. If you discuss questions with others you **must** write their names on your submission, and if you use any outside resources you **must** reference them. **Do not look at each others' writeups, including code.**
- There are 7 problems on 2 pages in this homework.
- **Keep in mind that problems and exercises are distinct in LFD.**
- All graphs should have clearly labeled axes. The Matlab `hist` function should be useful.

Problems:

1. (25 points) LFD **Exercise** (not Problem) 1.10. Note that you should be able to run a single simulation (flipping 1000 fair coins independently 10 times each, and finding the frequency of heads for each of them) in *one line* of Matlab code. For part (a), in addition to answering the question, also write the single line of Matlab code you used for this. There is no need to separately submit any code for this problem.
2. (25 points) Consider the following experiment on perceptron learning for random training sets of dimension 10:
   - Generate an 11-dimensional weight vector $\mathbf{w}^*$, where the first dimension is 0 and the other 10 dimensions are sampled independently at random from the uniform $(0, 1)$ distribution (the first dimension will serve as the threshold, and we'll just set it to 0 for convenience).
   - Generate a random training set with 100 examples, where each dimension of each training example is sampled independently at random from the uniform $(-1, 1)$ distribution, and the examples are all classified in accordance with $\mathbf{w}^*$.

- Run the perceptron learning algorithm, starting with the zero weight vector, on the training set you just generated, and keep track of the number of iterations it takes to learn a hypothesis that correctly separates the training data.

Write code in Matlab to perform the above experiment and then repeat it 1000 times (note that you're generating a new $\mathbf{w}^*$ and a new training set each time). We have provided, in your SVN repository, two stub files that you should complete for this purpose. The files have comments that explain their inputs and outputs. You need to commit your final versions along with your homework writeup.

Once you have your code working, plot a histogram of the number of iterations the algorithm takes to learn a linear separator (you should submit this with your writeup). How does the number of iterations compare with the bound on the number of errors we derived in class? Note that this bound will be different for each instantiation of $\mathbf{w}^*$ and the training set, so in order to answer this question, you should analyze the distribution of differences between the bound and the number of iterations. Plot and submit a histogram of the **log** of this difference, and discuss your interpretation of these results.

For up to 10 points of extra credit, can you characterize the situations in which the algorithm takes more iterations to correctly learn a hypothesis that separates the training data? Back up your answer with evidence from your experiments. Hint: You may want to try and visualize a lower dimensional version, and/or hold $\mathbf{w}^*$ fixed and vary the training set as you try to figure this out.

3. (10 points) LFD Problem 1.7

4. (10 points) LFD Problem 1.8

5. (10 points) LFD Problem 1.12

6. (10 points) LFD Problem 2.3

7. (10 points) LFD Problem 2.8