CSE 417T: Homework 1

Due on September 22, 2023

Notes:

- Please submit your homework via Gradescope. Please check the submission instructions for Gradescope provided on the course website.
- Please complete and submit this Python file for Problem 2.
- There will be two submission links for homework 1: one for report and the other for code. **Your score will be based on the report.** The code you submit is only used for checking the correctness and for running plagiarism checkers. Note that even if your report is correct, if there are issues with your code, you might lose points.
  - Report: Make sure you include all explanations and results in the report. Any results/explanations written in the code will not be graded.
  - Code: You only need to submit your code for Problem 2 (fill in the function implementations in the file and submit `hw1.py`). Your code will only be used for checking correctness (especially when/if we think there are discrepancies) and for running plagiarism checkers.
  - We plan to run automatic plagiarism tests for the code submissions (including the code submitted in previous semesters). The tests are performed after some initial level of compilation, so changing variable names and changing code sequences will be detected. Submissions that are too similar to other sources will be considered as violations of academic integrity and will be reported to the university.
- Make sure you **specify the pages for each problem correctly**. You **will not get points** for problems that are not correctly connected to the corresponding pages.
- 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 in this homework.
- **Keep in mind that problems and exercises are distinct in LFD.**
- All graphs should have clearly labeled axes.

Problems:

1. (25 points) LFD Problem 1.3.

2. (25 points) Implement perceptron learning algorithm (PLA) and examine its performance. Please complete and submit this Python file for this problem.

   You need to submit both your code and the report of this problem. For the code submission, fill in the function implementations and submit `hw1.py`. You can write additional functions or additional code in the main function. You need to include the figures/answers **in the report**. Figures/answers in the code do not count.

   Description: Consider the following experiment on running PLA for random training sets of size 100 and dimension 10 (i.e., $N = 100$ and $d = 10$.)

   - Create a random optimal separator $\vec{w}^*$:
     Generate an 11-dimensional weight vector $\vec{w}^*$, where the first dimension (i.e., w_0^*) is 0 and the other 10 dimensions are sampled independently and uniformly at random between from $[0, 1]$ (we just set w_0^* to 0 for convenience).
   - Generate a random training set with 100 data points, i.e., $D = \{(\vec{x}_1, y_1), ...(\vec{x}_{100}, y_{100})\}$, that are separable by $\vec{w}^*$:
     For each training data point $\vec{x}$, sample each of the 10 dimensions independently and uniformly at random from $[-1, 1]$ (Note that you need to insert $x_0 = 1$ for each data point $\vec{x}$). Calculate the label y of each data point $\vec{x}$ using the separator $\vec{w}^*$ (we assume separable data, so each point needs to be correctly classified by $\vec{w}^*$).
   - Run the perceptron learning algorithm:
     Implement and run PLA on the training set you just generated, starting with the zero weight vector. Keep track of the number of iterations it takes to learn a hypothesis that correctly separates the training data.

   Write code in Python to perform the above experiment and then repeat it 1000 times (note that you're generating a new $\vec{w}^*$ and a new training set D each time). We have provided two function headers (`perceptron_experiment` and `perceptron_learn`) that you should complete for this purpose. The file has comments that explain their inputs and outputs.

   Summarize your results in the report. Note that only the content included in the report will be graded. In particular, include the following in your report:

   - Plot a histogram of the number of iterations PLA takes to learn a linear separator.
   - Compare the number of iterations with the theoretical bound derived in Problem 1. Note that the bound will be different for each instantiation of $\vec{w}^*$ and the training set D. In order to answer this question, you should analyze the distribution of differences between the bound and the number of iterations. Plot a histogram of the **log** of this difference.
   - Discuss your interpretation of these results.

3. (10 points) LFD **Exercise** 1.10 (a) to (d). You do not need to submit code for this problem. (Note that *exercise* and *problem* are different in LFD! Exercises are embedded in chapters, and problems are listed after the end of each chapter.) Note that all plots need to have clearly labeled axes.

4. (10 points) LFD Problem 1.8.

   - Hint for (a): Try to write down $E(t)$ using the law of total expectation with the two partitions on t: $t \geq \alpha$ and $t < \alpha$.

5. (10 points) LFD Problem 1.12.

6. (10 points) LFD Problem 2.3.

7. (10 points) LFD Problem 2.8. Please provide explanations.