

## CARNEGIE MELLON UNIVERSITY 10-607

## HOMEWORK 2

DUE: Monday, Nov. 22, 2021

<https://www.cs.cmu.edu/~10607>

## INSTRUCTIONS

- **Format:** Use the provided LaTeX template to write your answers in the appropriate locations within the \*.tex files and then compile a pdf for submission. We try to mark these areas with STUDENT SOLUTION HERE comments. Make sure that you don't change the size or location of any of the answer boxes and that your answers are within the dedicated regions for each question/part. If you do not follow this format, we may deduct points.

You may also type you answer or write by hand on the digital or printed pdf. Illegible handwriting will lead to lost points. However, we suggest that try to do at least some of your work directly in LaTeX.

Programming components

- **How to submit written component:** Submit to Gradescope a pdf with your answers. Again, make sure your answer boxes are aligned with the original pdf template.
- **How to submit programming component:** See section Programming Submission for details on how to submit to the Gradescope autograder.
- **Policy:** See the course website for homework policies, including late policy, and academic integrity policies.

Name	
Andrew ID	
Hours to complete all components (nearest hour)	

## 1 Computational Complexity [12 pts]

1. [4 pts] Prove or disprove that  $n^4$  is in  $O(n^5)$ .

Each statement in your proof should have a justification, however, you don't need to explicitly write this justification. Make sure to avoid steps in your proof that are too large to be justified.

Proof

2. [4 pts] Prove or disprove that  $4n^4 + 3n^3 + 2n^2 + n \log n$  is in  $O(n^4)$ ?

Each statement in your proof should have a justification, however, you don't need to explicitly write this justification. Make sure to avoid steps in your proof that are too large to be justified.

Proof

3. [4 pts] Prove or disprove that  $n^2 \log n$  is in  $O(n^2)$ .

Each statement in your proof should have a justification, however, you don't need to explicitly write this justification. Make sure to avoid steps in your proof that are too large to be justified.

Proof

## 2 Perceptron (Programming) [10 pts]

In the accompanying file `perceptron.py`, fill out code to complete the provided functions for perceptron prediction and training updates. Don't include any plotting code in your `perceptron.py` submission.

1. [1 pts] Notice that we ran a fixed number of epochs (loops through the data) in `perceptron.py`. If we were checking for convergence after each epoch to see if the perceptron algorithm should stop, how many epochs would have run when the linearly separable training data i.e. `train_separable`?

2. [1 pts] After training on the linearly separable dataset, how many errors does the perceptron make on the test data `test`?

3. [2 pts] Plot the two-dimensional data with the training datapoints `train_separable` test datapoints `test`. Be sure to label your plot so it is clear which points are train vs test. Also include in the plot a line indicating the separating decision boundary learned by the perceptron training process.

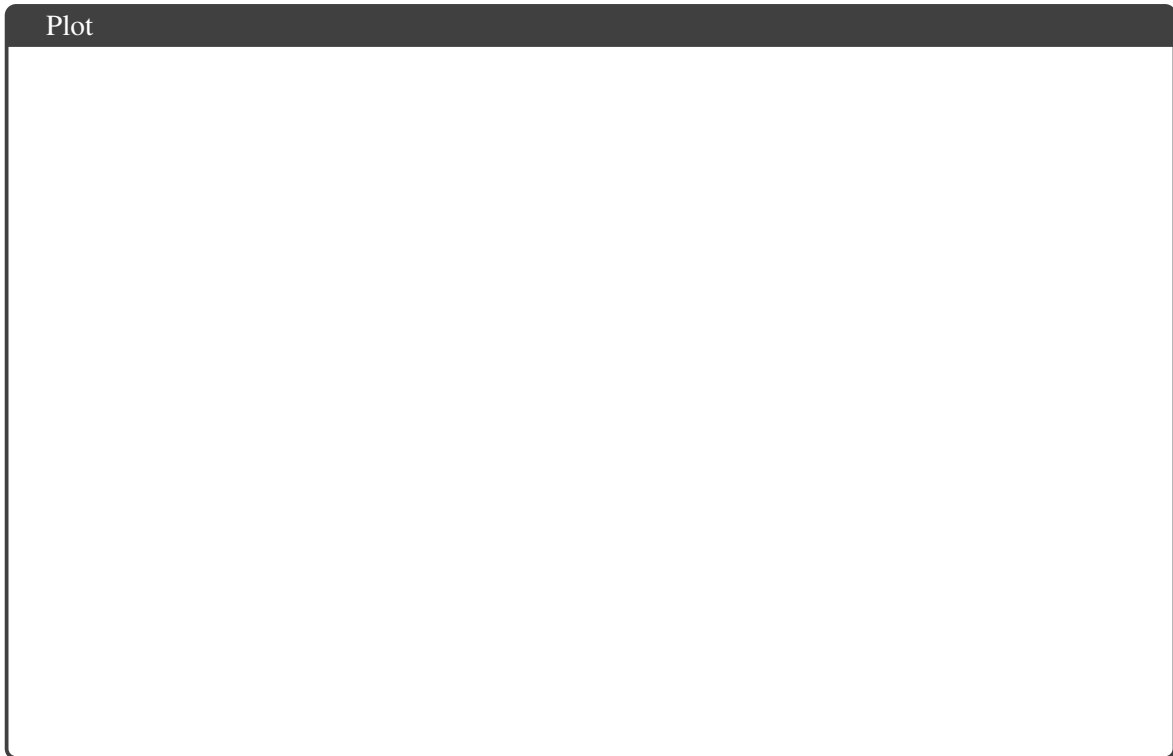


4. **[2 pts]** Plot the two-dimensional data with the training datapoints `train_nonseparable` and test datapoints `test`. Be sure to label your plot so it is clear which points are train vs test. Also include in the plot a line indicating the separating decision boundary learned by the perceptron training process for the non-separable training data.



5. **[1 pts]** In our implementation, you will notice that we have allowed the algorithm to run for a specific maximum number of epochs. In theory i.e. without any implementation-specific bound on the maximum number of epochs, will the perceptron algorithm converge to a training error rate of 0 for the training data `train_nonseparable` that is not linearly separable? If yes, how many epochs will it take for the algorithm to converge?

6. **[3 pts]** Plot the number of training errors versus training epoch for the two datasets `train_separable` and `train_nonseparable` in a single plot. You can get these quantities from the console output when you run the `perceptron.py` file after coding the perceptron predict and update rules. Be sure to label your plot so it is clear which dataset is which.



### 3 Dynamic Programming with Markov Chains (Programming) [3 pts]

In the accompanying file `markov_chain.py`, fill out code to complete the provided functions `calc_p_xt_from_p_xt_minus1` and `calc_p_xt`. See the comments in the code for instructions for the implementation of these methods.

Make sure to do the online questions about Markov chains first to understand the theory behind the programming.

1. [1 pts] Do the outputs from your implementation match your manually calculated answers to the questions in the online component?

Select one:

Yes     No

2. [2 pts] After running `python3 markov_chain.py`, include a legible screenshot of your console output here.

Output Screenshot





## 4 Programming Submission [1 pts]

1. [1 pts] In addition to the PDF submission with your written answers and plots, please submit the files `perceptron.py` and `markov_chain.py` directly to Gradescope under the HW2 (programming) assignment.

Your code file submissions should follow the provided code template files. In particular, they should NOT include any additional Python imports such as `matplotlib` which might make it fail on the autograder.

Please maintain any code for creating your plots in separate files. You do not have to submit your plotting code on Gradescope.

You may submit your code as many times as you like before the deadline.

Have you made the code submission on Gradescope?

**Select one:**

Yes     No

## 5 Collaboration Policy

After you have completed all other components of this assignment, report your answers to the following collaboration questions.

1. Did you receive any help whatsoever from anyone in solving this assignment? If so, include full details including names of people who helped you and the exact nature of help you received.

2. Did you give any help whatsoever to anyone in solving this assignment? If so, include full details including names of people you helped and the exact nature of help you offered.

3. Did you find or come across code that implements any part of this assignment? If so, include full details including the source of the code and how you used it in the assignment.