Math 40a: Introduction to Applied Mathematics
Homework 2

Upload your write-up to **Gradescope** to submit your assignment. You don't need to include python files, but please include a screenshot of your code for questions involving coding. **The graders should be able to evaluate your homework based only on the write-up without having to look at your separate code files.**

Problem 1 *(Modeling Exercise)*

Please turn in your results for the second modeling exercises that we started in class (see Latte).

- Part 2 of `information_theory_exercise1.pdf`.
- Part 2 of `information_theory_exercise2.pdf`.

Problem 2 *(3 Digit Information)*

Consider a channel where informaiton is sent as three-digit decimal numbers 000, 001, ..., 148, 149. Assume each of the 150 numbers is equally probable. The entropy of a number is therefore $H_N = -150\left(\frac{1}{150}\log_2\frac{1}{150}\right) = \log_2 150$.

1. Calculate the Shannon entropy of the first digit, second digit, and third digit.
2. Calculate the sum of the entropies of the three digits. Why do you think the sum is not equal to H_N?

Problem 3 *(English Language Word Pairs)*

In his paper, Shannon examines the frequency distributions of letter and word combinations. He provides a recipe for generating random English sentences as a tool to estimate the uncertainty (entropy) in the English language. We'll try to do this in a more modern way based on word-pair frequency data from the Corpus of Contemporary American English (`https://corpus.byu.edu/coca/`) for $> 1,000,000$ word pairs found in the English language.

You are given four .txt files: words.txt which contains the words, frequencies.txt which contains the sorted frequencies of word pairs, and the corresponding word indices `word1_index.txt` and `word2_index.txt`. These files may be read into Python using the code

```
words = open("words.txt",'r').read().split()
frequencies = open("frequencies.txt",'r').read().split()
index1 = open("word1_index.txt",'r').read().split()
index2 = open("word2_index.txt",'r').read().split().
```

The most frequently-occuring word pair is when `words[int(index2[0])]` follows `words[int(index1[0])]`, and it occurs `frequencies[0]` times. The second-most frequently-occuring word pair is `words[int(index2[1])]` following `words[int(index1[1])]`, and it occurs `frequencies[1]` times, and so on.

1. Inspect the 100 most common word pairs in English. Do any stand out?
2. Write down the 20 most common word pairs in English. Draw a diagram containing each word in a circle, and draw arrows from each first to second word in a word pair. This is a directed graph. What are the longest word sequences you can generate?
3. Can you travel from every word to every other word by following the arrows?

Problem 4 *(Conditional entropy)*

Derive $H(X,Y) = H(X|Y) + H(Y)$ where $H(Y)$ denotes the entropy of Y, $H(X,Y)$ denotes the entropy of the joint occurrence of X and Y, and $H(X|Y)$ denotes the entropy of X given Y.

Problem 5 *(Transmission rate in a noisy channel)*

Suppose there are two possible symbols 0 and 1, which are transmitted with probabilities $p_0 = p_1 = \frac{1}{2}$. If 0 is equally likely to be received as a 0 or 1, and similarly for 1, what is the rate of transmission? Please show all your work to earn full credit.