CS 571: Natural Language Processing **Spring 2023**

Homework 2

Lecturer: Dr. Fei Liu *Due: Monday, 2/20 11:59PM EST*

Note: *Homework modified from Eric Xing at Carnegie Mellon.*

2.1 HMM for DNA Sequence (7 points)

The goal of this assignment is for you to gain familiarity with **hidden Markov model (HMM)**. You will use HMM to decode a DNA sequence. It is well known that a DNA sequence is a series of components from A, C, G, T. Now let's assume there is one hidden variable S that controls the generation of DNA sequence. S takes 2 possible states S_1, S_2. Assume the following transition probabilities for HMM

$$P(S_1|S_1) = 0.8, P(S_2|S_1) = 0.2, P(S_1|S_2) = 0.3, P(S_2|S_2) = 0.7$$

emission probabilities as following

$$P(A|S_1) = 0.3, P(C|S_1) = 0.2, P(G|S_1) = 0.3, P(T|S_1) = 0.2$$
$$P(A|S_2) = 0.1, P(C|S_2) = 0.4, P(G|S_2) = 0.1, P(T|S_2) = 0.4$$

and initial probabilities as following

$$P(S_1) = 0.5, P(S_2) = 0.5$$

All transition, emission, initial probabilities are together referred to as θ. Assuming the observation sequence is $O = CGTCA$, in the first part of this assignment, you will manually compute the most likely hidden state sequence using the Viterbi algorithm.

Please submit: A report named `report_firstname_lastname.pdf`. Please report the decoded state sequence with intermediate calculations, i.e., V and backtracking matrices.

2.2 Text Classification (8 points)

For this question, you'll be using the 20 Newsgroups dataset for a binary text classification task. You can download the dataset from `http://qwone.com/~jason/20Newsgroups/` and choose the file named **"20news-bydate.tar.gz"**. Unpack it and look through the directories at some of the files. The documents are divided into a training set and a test set. Overall, there are 20 categories and roughly 19,000 documents. The label (category) of a document is its folder name. For this homework, you will only be using **two categories**: `rec.autos` and `comp.sys.mac.hardware` in the train/test folders.

Your task is to implement a naïve Bayes classifier for text classification. As a preprocessing step, you need to convert each text document into a sequence of tokens using spaCy (`https://spacy.io/usage/linguistic-features`). The preprocessing will include the following steps: sentence segmentation, word tokenization and lowercasing, and optionally, lemmaization and stopword removal (see `https://spacy.io/usage/spacy-101#annotations-pos-deps`).

To build the classifier, you can modify the provided starter code in HW1 or implement your own naïve Bayes classifier using an existing library/toolkit (see `https://scikit-learn.org/stable/modules/generated/`

`sklearn.naive_bayes.MultinomialNB.html`). If you choose to modify the starter code, be sure to modify the train/test splits so that the classifier will report results on the provided test set.

- **(2 points)** After preprocessing the documents in the training set (those in `rec.autos` and `comp.sys.mac.hardware` folders), how many documents are there in each category? What is the size of your vocabulary?
- **(6 points)** Evaluate performance of your model on the test set (documents in `rec.autos` and `comp.sys.mac.hardware` folders) by calculating the **F1-score**, assuming `rec.autos` is the positive category (see `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html`). What scores do you get?

Please submit: (1) A report named `report_firstname_lastname.pdf`. In the report, answer the above questions and describe your experimental setup, including but not limited to the programming language, preprocessing steps, running time, etc. (2) source code of your implementation in a zipped/tar file.