Assignment 3

In this assignment, you will use the Posterior Decoding and Viterbi algorithms on a Hidden Markov Model (HMM) to detect C/G rich regions in a DNA sequence.

DNA sequences are made up of the four letters A, C, T, and G (each letter corresponding to a particular nucleic acid). It turns out that, in the genomes of many organisms (including humans), the DNA letters are not distributed randomly, but rather have some regions with more A's and T's and some regions with more C's and G's. It turns out that C/G rich regions tend to have more genes, whereas A/T rich regions tend to have more non-functional DNA. In this assignment, you have to analyze 19 DNA sequences extracted from human chromosomes.

We will use an HMM to detect C/G rich regions in a DNA sequence. Our HMM has two states which correspond to low- and high-C/G regions respectively. The prior, transition and emission probabilities are given in the template file.

You need to implement two functions. HMM.viterbi() uses the Viterbi algorithm to calculate the most likely sequence of states given a sequence of DNA characters. That is, HMM.viterbi() computes argmax_states P(sequence, states). HMM.posterior() uses the smoothing algorithm to compute the posterior distribution of the state at a given position: P(state_i | sequence). HMM.posterior_decode(), which is provided in the starter code will receive the output of HMM.posterior() and generate sequence of states for you. Although both Viterbi and Posterior Decoding algorithms are trying to decode your HMM, the generated sequences of states might be different on various inputs.

In order to avoid underflow problems, you should represent all probability values in log form. To add probabilities, use the provided log_sum() function, which uses the log-sum-exp method to add log probabilities in a numerically-stable way.

Files

You can download all the files required for this assignment in here [Updated 15 November]. You have to write all your code in a3_template.py between the comments marked "Start/End your code". You can run code by running:

```
python a3_template.py "small.out"
```

The input file is a set of DNA sequences. A tiny dataset of 19 sequences is provided for you to test your implementations.

Your program should output two files file in the following format:

The first file is the output of the Viterbi algorithm. For each sequence of observations in the dataset, you have to write three lines in the output file. The first and second lines are the number of inferred 0 and 1 states respectively in your inferred sequence of states. The last line is a sequence of states (0 or 1).

The second file is the same as the first one but is generated using the Posterior Decoding algorithm.

Use the write_output() function to produce both output files. You can see the correct output files for the small dataset in the "reference_posterior.txt" and "reference_viterbi.txt" provided in the assignment folder.

Guidelines

Use Python 3 for this assignment. You may add extra functions as you like. Please do not change the name of classes, functions, or variable names.

100 points total, worth 12% of the course grade. Turn in on CourSys. Submit a compressed directory (.zip or .tar.gz) with your code in one file name "a3.py".

For this assignment, you have to run your a3.py implementation first to create output files, and then use the autograder to grade the output. You should be able to test your code running the following commands:

```
python autograder.py viterbi "small_viterbi_output.txt"
python autograder.py posterior "small_posteri_output.txt"
```

Grading: Your solution will be evaluated on accuracy and clarity. Please use a logical structure, use informative variable names, and document your code thoroughly so that your solution is understandable.

You are encouraged to work in a group. Feel free to discuss solution strategies and check each other’s work. However, you must write all the text and code you submit on your own. Joint submissions are not allowed, nor is copying someone else's text or code. Plagiarism is not okay and will be taken very seriously. If you’re not sure whether something is okay, please ask.

If you are having trouble, please take advantage of office hours and the discussion forum. If you see a question from another student on the discussion forum that you think you know the answer to, please respond. (Do your best to lead your fellow student to a solution, rather than simply giving the solution directly.)