IUI Assignment: Implement Hidden Markov Model (Due: May 7, 2020 – 23:59)

Submit your homework via Google Forms: https://forms.gle/dMf68JktRYJtgnWV6

Be sure your zip file is named as yourname_yoursurname.zip. The implementation can be done using Python, R or MATLAB. (*or JavaScript ~~ Please see the "Extra" note at the end of the document*)

The aim of the homework is to implement an HMM. In order to have a working HMM, you need to deal with 3 basic problems for HMM:

1. Given the observation sequence O = O1 O2. . .OT and a model λ = (A, B, π), how do we efficiently compute P(O|λ), the probability of the observation sequence, given the model? 2. Given the observation sequence O = O1 O2. . .OT and a model λ, how do we choose a corresponding state sequence Q= q1 q2 ….qT which is optimal in some meaningful sense (i.e. best "explains" the observations)? 3. How do we adjust the model parameters λ = (A, B, π) to maximize P (O|λ)?

Part 1:

Executable provided for part1 should take "obsv_prob" (as command), test sequence (testdata.txt) and model file (model.txt) as input in order to accomplish this part. Here, you need to return the probability of a given test sequence (provided in testdata.txt) which is P (O|λ). You may return P (O|λ) computed by Forward procedure. Print results properly.

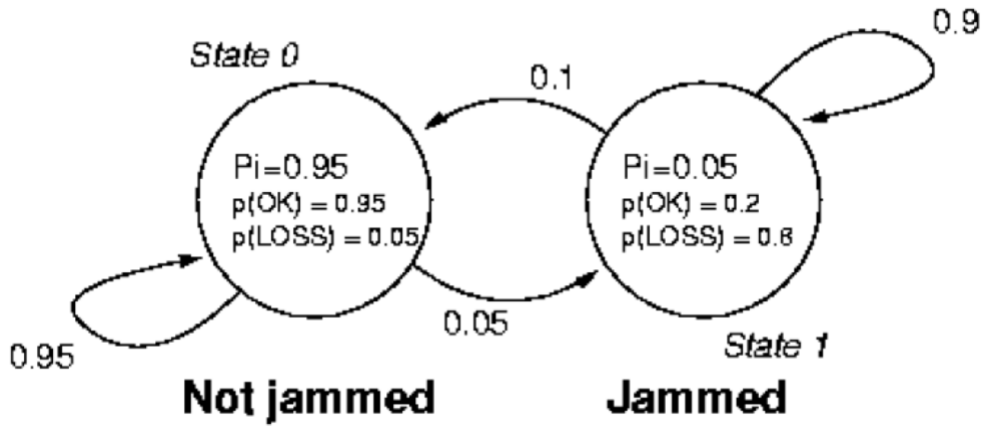
Part 2:

Executable provided for part2 should take "viterbi" (as command), test sequence (testdata.txt) and model file (model.txt) as input in order to accomplish this part. You need to find the best state sequence for a given test sequence (provided in testdata.txt) and return the best state sequence as well as P (O|λ) computed over this sequence. Also print the number of necessary state transitions. Print results properly.

Part 3:

Provide an executable file which takes "learn" (as command), number of states and observation sequence file (data.txt) as input. This executable should train HMM by using the given observation sequences and write description of the model to a file (HMMdescription.txt). Description should include at least number of states, state transition matrix (A), symbol probability distributions (B) and initial state distributions (π). You need to implement BaumWelch method for learning.

Description of the sample data file: Sample observation sequences (data.txt) are created by using the state diagram below. This diagram is for packet loss scenario in jammed or not jammed states.



You may set your models according to the given state diagram above and compute P(O|λ) and Viterbi sequence. Specify and set your model as described in the model file (model.txt). Sample data.txt has 0s and 1s, where 0 means packet is sent correctly, 1 means packet is dropped in the context of provided state diagram. Each line in data.txt is an observation. It is important to adjust your program for variable length of observation. You must be careful about underflow/overflow since for a long sequence of observation you may get underflow/overflow easily while computing probabilities. Also you should return log likelihoods for any probability answer that you are asked in homework.

Notes: Your implementation will be tested with a different state diagram than the one provided above. As a guide you may use Rabiner's article on HMM (provided in the homework folder). Also deliver a report explaining overall structure of your implementation and progress of your implementation. Proper commenting in your code is required.

Grading: Each part will count as a single assignment. Therefore, this assignment will count as 3 assignments as a whole.

Extra: **Explorable Explanations** is an umbrella name to designing a new reading and learning experience on the web. You can complete the assignment with creating a new Explorable for HMMs. If you are bored at home in these days and want to create a new learning experience for the others, please contact with the TA and plan your explorable. You will complete the above steps, but you will provide the content in an instructive manner on the web. For more information about how to create an explorable: https://blog.ncase.me/explorable-explanations/ [Please note that we do not provide bonus credit for choosing this path]