CSE 150A/250A. Assignment 6 *(42 pts total)*

Out: *Mon May 19*
Due: *Mon May 26* (by 11:59 PM, Pacific Time, via Gradescope)
Grace period: 24 hours

6.1 Viterbi algorithm *(14 pts)*

In this problem, you will decode an English sentence from a long sequence of non-text observations. To do so, you will implement the same basic algorithm used in most engines for automatic speech recognition. In a speech recognizer, these observations would be derived from real-valued measurements of acoustic waveforms. Here, for simplicity, the observations only take on binary values, but the high-level concepts are the same.

Consider a discrete HMM with $n = 27$ hidden states $S_t \in \{1, 2, \ldots, 27\}$ and binary observations $O_t \in \{0, 1\}$. Download the ASCII data files from Canvas for this assignment. These files contain parameter values for the initial state distribution $\pi_i = P(S_1 = i)$, the transition matrix $a_{ij} = P(S_{t+1} = j \mid S_t = i)$, and the emission matrix $b_{ik} = P(O_t = k \mid S_t = i)$, as well as a long bit sequence of $T = 430000$ observations.

Use the Viterbi algorithm to compute the most probable sequence of hidden states conditioned on this particular sequence of observations. As always, you may program in the language of your choice. Turn in the following:

(a) a print-out of your source code *(10 pts)*

(b) a plot of the most likely sequence of hidden states versus time. *(4 pts)*

To check your answer: suppose that the hidden states $\{1, 2, \ldots, 26\}$ represent the letters $\{\texttt{a}, \texttt{b}, \ldots, \texttt{z}\}$ of the English alphabet, and suppose that hidden state 27 encodes a space between words. If you have implemented the Viterbi algorithm correctly, the most probable sequence of hidden states *(ignoring repeated elements)* will reveal a famous quotation.

6.2 Inference in HMMs *(13 pts)*

Consider a discrete HMM with hidden states S_t, observations O_t, transition matrix $a_{ij} = P(S_{t+1}=j \mid S_t=i)$ and emission matrix $b_{ik} = P(O_t=k \mid S_t=i)$. In class, we defined the forward-backward probabilities:

$$\alpha_{it} = P(o_1, o_2, \ldots, o_t, S_t=i)$$
$$\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T \mid S_t=i)$$

for a particular observation sequence $\{o_1, o_2, \ldots, o_T\}$ of length T. In terms of these probabilities, which you may assume to be given, as well as the transition and emission matrices of the HMM, show how to (efficiently) compute the following quantities:

(a) $P(S_{t+1}=j \mid S_t=i, o_1, o_2, \ldots, o_T)$ *(3 pts)*

(b) $P(S_t=i \mid S_{t+1}=j, o_1, o_2, \ldots, o_T)$ *(3 pts)*

(c) $P(S_{t-1}=i, S_t=j, S_{t+1}=k \mid o_1, o_2, \ldots, o_T)$ *(4 pts)*

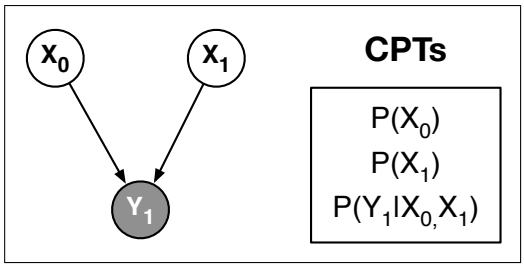
(d) $\hat{s}_t = \operatorname{argmax}_i P(S_t=i \mid o_1, o_2, \ldots, o_T)$ *(3 pts)*

You may assume that $1 < t < T$; in particular, you are *not* asked to consider the boundary cases.

6.3 Belief updating *(9 pts)*

Consider the simple belief network on the right with nodes X_0, X_1, and Y_1. To compute the posterior probability $P(X_1 \mid Y_1)$, we can use Bayes rule:

$$P(X_1 \mid Y_1) = \frac{P(Y_1 \mid X_1)\, P(X_1)}{P(Y_1)}$$



(a) Show how to compute the term $P(Y_1 \mid X_1)$ in the numerator of Bayes rule. *(2 pts)*

(b) Show how to compute the term $P(Y_1)$ in the denominator of Bayes rule. *(2 pts)*

Now consider the belief network shown at the bottom of the page. It does not have the same structure as an HMM, but using similar ideas we can derive efficient algorithms for inference. In particular, consider how to compute the posterior probability $P(X_t \mid Y_1, Y_2, \ldots, Y_t)$ that accounts for evidence up to and including time t. We can derive an efficient recursion from Bayes rule:

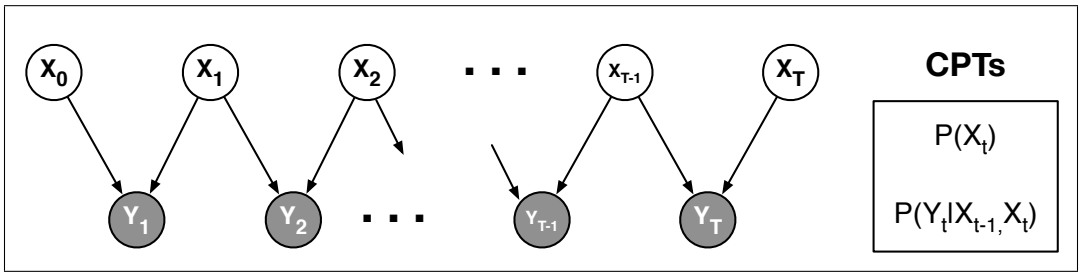
$$P(X_t \mid Y_1, Y_2, \ldots, Y_t) = \frac{P(Y_t \mid X_t, Y_1, Y_2, \ldots, Y_{t-1})\, P(X_t \mid Y_1, Y_2, \ldots, Y_{t-1})}{P(Y_t \mid Y_1, \ldots, Y_{t-1})},$$

where the nodes $Y_1, Y_2, \ldots, Y_{t-1}$ are treated as background evidence. In parts (c–e) of this problem you will compute the individual terms that appear in this version of Bayes rule. You should express your answers in terms of the CPTs of the belief network and the probabilities $P(X_{t-1}{=}x \mid Y_1, Y_2, \ldots, Y_{t-1})$, *which you may assume have been computed in a previous step of the recursion*. Your answers to parts (a) and (b) may be instructive for parts (d) and (e).

(c) Show how to simplify the term $P(X_t \mid Y_1, Y_2, \ldots, Y_{t-1})$ in the numerator of Bayes rule. *(1 pt)*

(d) Show how to compute the term $P(Y_t \mid X_t, Y_1, Y_2, \ldots, Y_{t-1})$ in the numerator of Bayes rule. *(2 pts)*

(e) Show how to compute the term $P(Y_t \mid Y_1, Y_2, \ldots, Y_{t-1})$ in the denominator of Bayes rule. *(2 pts)*



6.4 Most likely hidden states *(6 pts)*

The Viterbi algorithm in HMMs computes the most likely *sequence* of hidden states for a particular sequence of observations:

$$\{s_1^*, s_2^*, \ldots, s_T^*\} = \underset{s_1, s_2, \ldots, s_T}{\operatorname{argmax}} P(s_1, s_2, \ldots, s_T \mid o_1, o_2, \ldots, o_T).$$

Compare these *collectively* optimal hidden states s_t^* to the *individually* optimal hidden states $\hat{s}_t$:

$$\hat{s}_t = \underset{s_t}{\operatorname{argmax}} P(S_t = s_t \mid o_1, o_2, \ldots, o_T) \qquad \text{for } t \in \{1, 2, \ldots, T\}.$$

Answer the following True/False questions and explain your reasoning:

(a) It is *possible* that $P(\hat{s}_1, \ldots, \hat{s}_T \mid o_1, \ldots, o_T) > P(s_1^*, \ldots, s_T^* \mid o_1, \ldots, o_T)$. *(2 pts)*

(b) It is *possible* that $\hat{s}_t = s_t^*$ for all t. *(2 pts)*

(c) It is *always* true that $\hat{s}_t = s_t^*$ for all t. *(2 pts)*