CSCI 5512: Artificial Intelligence II (Spring 2022)

Homework 2

(Due Thu, Feb 24, 11:59 pm central)

1. **(50 points)** [Programming Assignment] Consider the Hidden Markov Model in Figure 1.

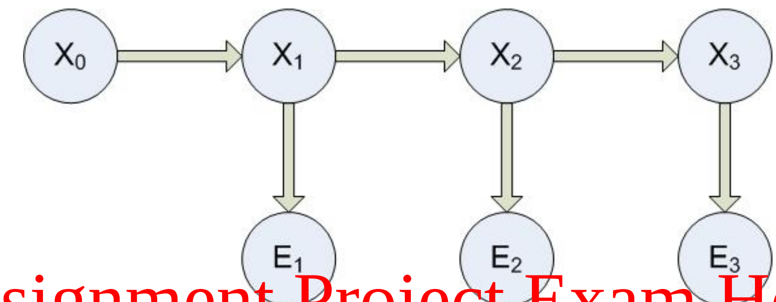

Figure 1: Hidden Markov Model

Assume each of the hidden variables $X_i, i = 0, 1, 2, 3, \ldots$ and the evidence variables $E_i, i = 1, 2, 3, \ldots$ to be boolean, and can take two values T and F. Let $P(X_0 = T) = P(X_0 = F) = 0.5$. Let the transition matrix $P(X_{t+1}|X_t)$ and sensor matrix $P(E_t|X_t)$ be given by

$$T = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} \qquad E = \begin{bmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{bmatrix} ,$$

where, in the T matrix,

$$T_{11} = P(X_{t+1} = T|X_t = T) \ , \qquad T_{12} = P(X_{t+1} = F|X_t = T) \ ,$$
$$T_{21} = P(X_{t+1} = T|X_t = F) \ , \qquad T_{22} = P(X_{t+1} = F|X_t = F) \ ,$$

and in the E matrix,

$$E_{11} = P(E_t = T|X_t = T) \ , \qquad E_{12} = P(E_t = F|X_t = T) \ ,$$
$$E_{21} = P(E_t = T|X_t = F) \ , \qquad E_{22} = P(E_t = F|X_t = F) \ .$$

Consider two sequences of evidence $\mathbf{e}_{1:10}$ over 10 time steps:

Evidence sequence 1: $\mathbf{e}_{1:10} = \langle F, F, F, T, T, T, T, F, F, F \rangle$

Evidence sequence 2: $\mathbf{e}_{1:10} = \langle F, T, F, T, F, T, F, T, F, T \rangle$

For each of the above two sequences of evidence:

(a) (25 points) Write a program to compute the smoothed estimates of $X_t, t = 1, \ldots, 10$ given evidence $\mathbf{e}_{1:10}$.

(b) (25 points) Write a program to find the most likely sequence of states $X_{1:10}$ given evidence $\mathbf{e}_{1:10}$.

In addition to the smoothed estimates and sequence of states, you have to submit code for SmoothHMM implementing computation of smoothed estimate and MaxSeq implementing computation of the most likely sequence. The code for both algorithms should take two input arguments:

(i) n, the length of the evidence ($n = 10$ for the two examples above)

(ii) The evidence sequence containing a '1' for T and '0' for F. Thus, for the first example $\mathbf{e}_{1:10} = \langle F, F, F, T, T, T, T, F, F, F \rangle$, implying the input should be 0 0 0 1 1 1 1 0 0 0.

The output for SmoothHMM should be an list of length n with smoothed estimates of $P(X_t = T), t = 1, \ldots, n$. The output should be clearly displayed on screen after running the program.

The output for MaxSeq should be a binary list of length n containing the most likely sequence of states with 1 corresponding to T and 0 corresponding to F. The output should be clearly displayed on screen after running the program.

Sample input for Python 3.6 for (a) and (b) when $n = 10$ and $\mathbf{e}_{1:10} = \langle F, F, F, T, T, T, T, F, F, F \rangle$

```
$python SmoothHMM.py 10 0 0 0 1 1 1 1 0 0 0
$python MaxSeq.py 10 0 0 0 1 1 1 1 0 0 0
```


2. **(50 points)** [Programming Assignment] Consider the umbrella network shown in Figure 2. Let $U_1, U_2, \ldots$ denote the sequence of evidence variables (umbrella), where $\forall i$, $U_i = T$ (true) or F (false). Let R_i be the random variable corresponding to the hidden state (rain) at step i. Assume the prior probability of rain $P(R_0 = T) = P(R_0 = F) = \frac{1}{2}$. Consider the following three evidence sequences:

(i) $\mathbf{u}_{1:10} = (T, T, T, T, T, F, F, F, F, F)$

(ii) $\mathbf{u}_{1:10} = (F, F, F, F, F, F, F, T, T, T)$

(iii) $\mathbf{u}_{1:10} = (F, T, F, T, F, T, F, T, F, T)$

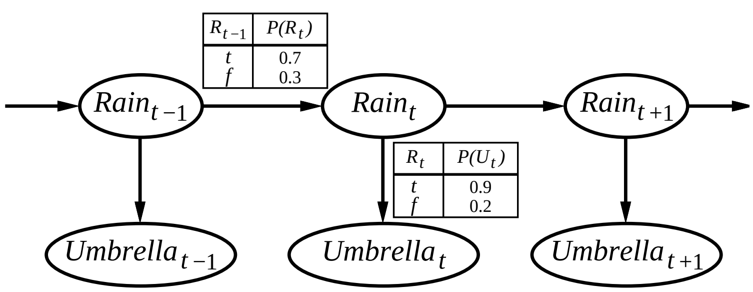

Figure 2: The Umbrella Network

For each of the three choices of $\mathbf{u}_{1:10}$, your code should output the filtering probability $P(R_{10}|\mathbf{u}_{1:10})$. We want to approximate this probability using two separate sample methods as follows:

(a) (20 points) Estimate $P(R_{10}|\mathbf{u}_{1:10})$ using likelihood weighting with 100 and 1000 samples. You have to submit code for lwUmbrella which takes 3 arguments: an integer numSamples, denoting the number of samples (set to 100 and 1000), an integer numSteps, denoting the number of steps (set to 10), and evidence, denoting the evidence $\mathbf{u}_{1:numSteps}$ $(T = 1, F = 0)$ of length numSteps. Run the likelihood weighting 10 times, and find the average of the estimate $P(R_{numSteps}|\mathbf{u}_{1:numSteps})$ and the variance of the estimate.

Sample input for Python 3.6 for when $numSamples = 100$, $numSteps = 10$, and $\mathbf{u}_{1:10} = (T, T, T, T, T, F, F, F, F, F)$

```
$python lwUmbrella.py 100 10 1 1 1 1 1 0 0 0 0 0
```

(b) (25 points) Estimate $P(R_{10}|\mathbf{u}_{1:10})$ using particle filtering with 100 and 1000 particles. You have to submit code for pfUmbrella, which takes 3 arguments: an integer numSamples, denoting the number of particles (set to 100 and 1000), an integer numSteps, denoting the number of steps (set to 10), and evidence, denoting the evidence $\mathbf{u}_{1:numSteps}$ $(T = 1, F = 0)$ of length numSteps. The output should be the estimate $P(R_{numSteps}|\mathbf{u}_{1:numSteps})$ and the variance of the estimate.

Sample input for Python 3.6 for when $numSamples = 100$, $numSteps = 10$, and $\mathbf{u}_{1:10} = (T, T, T, T, T, F, F, F, F, F)$

```
$python pfUmbrella.py 100 10 1 1 1 1 1 0 0 0 0 0
```


(c) (5 points) Comment on the relative performance of the two methods on the three evidence sequences and two sample sizes. In particular, how close are the estimates to the true probabilities and what are the variance of the estimates.

Instructions

Code can only be written in Python 3.6+; no other programming languages will be accepted. One should be able to execute all programs from the Python command prompt or terminal. Each program must take the inputs in the order specified in the problem and display the textual output via the terminal and plots/figures should be included in the report. Please specify instructions on how to run your program in the README file.

For each part, you can submit additional files/functions as needed. You may use libraries for basic matrix computations and plotting such as numpy, pandas, and matplotlib. Put comments in your code so that one can follow the key parts and steps in your code.

Your code must be runnable on a CSE lab machine (e.g., csel-kh1260-01.cselabs.umn.edu). One option is to SSH into a machine. Learn about SSH, and other options, at these links: `https://cse.umn.edu/cseit/self-help-guides/secure-shell-ssh` and `https://cse.umn.edu/cseit/self-help-guides`.

Instructions

Follow the rules strictly. If we cannot run your code, you will not get any credit.

- **Things to submit**

  1. hw2.pdf: A document which contains solutions to all problems. This document must be in PDF format (doc, jpg, etc. formats are not accepted). If you submit a scanned copy of a hand-written document, make sure the copy is clearly readable, otherwise no credit may be given.

2. Python code for Problem 1 (a) and (b) (must include the required `SmoothHMM.py` and `MaxSeq.py` files).
3. Python code for Problem 2 (a) and (b) (must include the required `lwUmbrella.py` and `pfUmbrella.py` files).
4. README.txt: README file that contains your name, student ID, email, assumptions you are making, other students you discussed the homework with, and other necessary details.
5. Any other files which are necessary for your code (such as package dependencies like a requirements.txt or yml file).

Homework Policy. (1) You are encouraged to collaborate with your classmates on homework problems, but each person must write up the final solutions individually. You need to list in the README.txt which problems were a collaborative effort and with whom. (2) Regarding online resources, you should **not**:

- Google around for solutions to homework problems,
- Ask for help on online,
- Look up things/post on sites like Quora, StackExchange, etc.