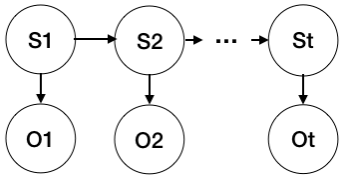
Fall 2019
CSC 591/791: HW 2
10/14/2019

This assignment contains 3 questions. Please read and follow the following instructions.

- **DUE DATE: Oct 30th, 11:45 PM**
- **TOTAL NUMBER OF POINTS: 100**
- Clearly list your team ID, each team member's **names and Unity IDs** at the top of your submission.
- Submit only a **single pdf file** of your answers.

---

1. (30 points) [Farzaneh Khoshnevisan] [**HMM**]
   Infection is a common condition among patients in ICU settings and can have various roots, which makes it challenging to be determined. Assume that we want to model infection using an HMM, while *infection* is the hidden state and the only available observation is *blood pressure* (0 for normal and 1 for abnormal). When patients entering ICU, the probability of being infected is 0.75. At any given time, infected patients have 40% chance of improving to be uninfected and uninfected patients have 20% chance of becoming infected. There is 80% chance of observing an abnormal blood pressure for infected patients while only 10% chance of observing abnormal observation for uninfected patients.



   (a) (5 points) How many parameters are required to fully define this HMM. Justify your answer.
   Solution: There are a total of three probability distributions that define an HMM, the initial probability distribution, the transition probability distribution, and the emission probability distribution. In general, when we have k hidden states, k parameters are required to define the initial probability distribution (we'll ignore all of the -1s for this problem to make things cleaner). For the transition distribution, transitions are from any one of k states to any of the k states (including staying in the same state), so k^2 parameters are required. If we have m different observations, we need a total of km parameters for the emission probability distribution, since each of the k states can emit each of the m observations. Thus, the total number of parameters required are $k + k^2 + km$. Note that the number of parameters doe snot depend on the length of the HMMs. In this specific question, $k = 2$ and $m = 2$, therefore there are 10 parameters required to define this HMM.

(b) (3 points) Create initial, transition, and emission probability tables based on the problem statement given above.

Solution:

S_i	S_{i+1}	$P(S_{i+1}\|S_i)$
T	F	0.4
T	T	0.6
F	T	0.2
F	F	0.8

S_i	O_i	$P(O_i\|S_i)$
T	0	0.2
T	1	0.8
F	0	0.9
F	1	0.1

S_1	$P(S_1)$
T	0.75
F	0.25

(c) (6 points) Using the described HMM and the generated probability tables, apply the forward algorithm to compute the probability that we observe the sequence $\{0, 1, 1\}$ blood pressure. Show your work (i.e., show each of your αs).

Solution: The values of different alphas and the probability of the sequences are as follows:

$$\alpha_1^T = 0.75 \times 0.2 = 0.15$$
$$\alpha_1^F = 0.25 \times 0.9 = 0.225$$
$$\alpha_2^T = 0.8 \times (0.15 \times 0.6 + 0.225 \times 0.2) = 0.108$$
$$\alpha_2^F = 0.1 \times (0.15 \times 0.4 + 0.225 \times 0.8) = 0.024$$
$$\alpha_3^T = 0.8 \times (0.108 \times 0.6 + 0.024 \times 0.2) = 0.05568$$
$$\alpha_3^F = 0.1 \times (0.108 \times 0.4 + 0.024 \times 0.8) = 0.00624$$
$$P(\{O_t\}_{t=1}^T) = 0.06192$$

(d) (6 points) Using the backward algorithm, compute the probability that we observe the aforementioned sequence ($\{0, 1, 1\}$). Again, show your work (i.e., show each of your βs).

Solution: The values of different alphas and the probability of the sequences are as follows:

$$\beta_3^T = 1$$
$$\beta_3^F = 1$$
$$\beta_2^T = 0.6 \times 0.8 \times 1 + 0.4 \times 0.1 \times 1 = 0.52$$
$$\beta_2^F = 0.2 \times 0.8 \times 1 + 0.8 \times 0.1 \times 1 = 0.24$$
$$\beta_1^T = 0.6 \times 0.8 \times 0.52 + 0.4 \times 0.1 \times 0.24 = 0.2592$$
$$\beta_1^F = 0.2 \times 0.8 \times 0.52 + 0.8 \times 0.1 \times 0.24 = 0.1024$$
$$P(\{O_t\}_{t=1}^T) = 0.15 \times 0.2592 + 0.225 \times 0.1024 = 0.06192$$

(e) (4 points) Using the forward-backward algorithm, compute the most likely setting for each state. Show your work.

Solution: We already have the alphas and betas from the previous two computations. Note that the most likely state at time t is state T if $\alpha_t^T \beta_1^T > \alpha_1^F \beta_1^F$ and state F if the inequality is reversed. We predict that $\{T, T, T\}$ is the most likely setting of states. The relevant values for the computation are:

$$\alpha_1^T \beta_1^T = 0.03888$$
$$\alpha_1^F \beta_1^F = 0.02304$$
$$\alpha_2^T \beta_2^T = 0.05616$$
$$\alpha_2^F \beta_2^F = 0.00576$$
$$\alpha_3^T \beta_3^T = 0.055$$
$$\alpha_3^F \beta_3^F = 0.00624$$

(f) (6 points) Use the Viterbi algorithm to compute the most likely sequence of states. Show your work.
Solution: The Viterbi algorithm predicts that the most likely sequence of states is $\{T, T, T\}$, based on the following computations:

$$V_1^T = 0.75 \times 0.2 = 0.15$$
$$V_1^F = 0.25 \times 0.9 = 0.225$$
$$V_2^T = 0.8 \times \max\{0.225 \times 0.2, 0.15 \times 0.6\} = 0.072$$
$$V_2^F = 0.1 \times \max\{0.225 \times 0.8, 0.15 \times 0.4\} = 0.018$$
$$V_3^T = 0.8 \times \max\{0.072 \times 0.6, 0.018 \times 0.2\} = 0.03456$$
$$V_3^F = 0.1 \times \max\{0.072 \times 0.4, 0.018 \times 0.8\} = 0.00288$$

parts

(g) (20 points) [Farzaneh Khoshnevisan] [**HMM**]
We design a two-state HMM for a dice toss game between two players where the output of their toss is our only observation. S_1 and S_2 indicate the hidden state where the dice is tossed by player 1 and player 2, respectively. The transition probabilities between these two states are given in diagram below. Both players are equally likely to start the game and they each play with a biased dice. The output distribution corresponding to each player is defined over $\{1, 2, 3, 4, 5, 6\}$ and is given in the table below the diagram.

	S_1	S_2
P(x=1)	0.01	0.32
P(x=2)	0.21	0.14
P(x=3)	0.3	0.14
P(x=4)	0	0.08
P(x=5)	0.18	0
P(x=6)	0.3	0.32

(a) (5 points) Give an example of one output sequence of length 2, which cannot be generated by the given HMM. Justify your answer.
Solution: {5,4}. This output sequence cannot be generated by the HMM model. Because observation 5 only shows up in S1, observation 4 only shows up in S2. However, the transition probability from S1 to S2 is zero. That is to say, if the first observation is 5, the corresponding hidden state must be S1. And we will stay at S1 forever, the observation 1 will never happen. So there is no way this HMM model could generate this sequence {5,4}.

(b) (5 points) We generated a sequence of $20,601^{2020}$ observations from this HMM, and found that the last observation in the sequence was 1. What is the most likely hidden state corresponding to the last observation? Justify your answer.
Solution: S1 is the most likely hidden state. Even though the transition probability from S2 to S1 is 0.01, which is relatively small, the length of observations sequence $20,601^{2020}$ is extremely large. This statistically makes the probability where S1 happens to be 1 in the almost sure sense. And once it is in S1, the future state will be S1 forever. So the most likely hidden state corresponding to the last observation is S1.

(c) (5 points) Consider an output sequence $\{1, 1\}$. What is the most likely sequence of hidden states corresponding to this output observation sequence? Show your work.
Solution: {S2,S2}. This conclusion is obtained based on Viterbi Algorithm as below. For the 1st observation, we have

$$\delta_1(1) = 0.01 \times 0.5 = 0.005$$
$$\delta_1(2) = 0.32 \times 0.5 = 0.16$$

For the 2nd observation, we have

$$\delta_2(1) = 0.01 \times \max\{\delta_1(1) \times 1, \delta_1(2) \times 0\} = 0.01 \times \delta_1(1) \times 1 = 0.00005$$
$$\delta_2(2) = 0.32 \times \max\{\delta_1(1) \times 0, \delta_1(2) \times 0.99\} = 0.32 \times \delta_1(2) \times 0.99 = 0.050688$$

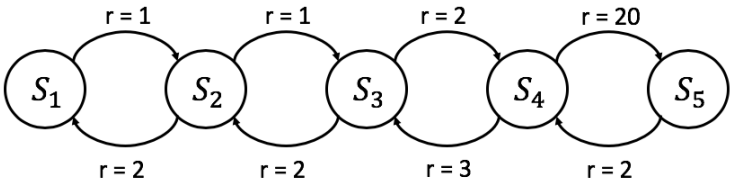
Thus, S2 is the most likely hidden state for second observation 1, then the path can be retrieved by back-tracking as S2,S2.

(d) (5 points) Now, consider an output sequence $\{1, 1, 3\}$. What are the first two states of the most likely hidden state sequence? Show your work.
Solution: {S2,S2}. We continue to calculate δ_3 based on problem (c).
For the 3rd observation, we have

$$\delta_3(1) = 0.3 \times \max\{\delta_2(1) \times 1, \delta_2(2) \times 0\} = 0.3 \times \delta_2(2) \times 1 = 0.0152$$
$$\delta_3(2) = 0.14 \times \max\{\delta_2(1) \times 0, \delta_2(2) \times 0.99\} = 0.14 \times \delta_2(2) \times 0.99 = 0.00702$$

Thus, S2 is the most likely hidden state for last observation 3, then the path can be retrieved by back-tracking as S2,S2,S2.

(e) (50 points) [Song Ju] [**Reinforcement Learning**]
Consider the following Markov Decision Process:



Our state space S: $\{S_1, S_2, S_3, S_4, S_5\}$ and our action space A: $\{"Left", "Right"\}$. For all parts of this problem, assume that $\gamma = 0.8$.

For subquestions (a)-(c) below, we assume that all actions are deterministic:

(a) (5 points) What is the optimal policy for this MDP?
Solution: The optimal policy is:

$$\pi = \left| \begin{array}{ll} S_1: & right \\ S_2: & right \\ S_3: & right \\ S_4: & right \\ S_5: & left \end{array} \right|$$

(b) (15 points) Calculate $V^*_{S_5}$. Show your work. (The acceptable answer should be a numeric number and you should show all the key steps).
Solution: The update value functions for $S4$ and $S5$ are listed as follows:

$$V_4^{t+1} = 20 + \gamma V_5^t$$
$$V_5^{t+1} = 2 + \gamma V_4^t$$

So we will get:

$$V_5^t = 2 + \gamma(20 + \gamma V_5^{t-2})$$

When we solve this series as time t goes to infinite, we will get the $V^*(S5)$ as follows:

$$V^*(S5) = \frac{2 + 20\gamma}{1 - \gamma^2} = 50$$

(c) (10 points) Consider executing Q-learning on this MDP. Assume that 1) all of the initial Q values are 0, 2) $\alpha = 0.5$, and 3) it uses a greedy exploration policy by always choosing the action with maximum Q value at any given state. The algorithm breaks ties by choosing "Left". What are the first 10 (state, action) pairs if our robot learns using Q-learning and starts in state S_3? (A candidate answer can be expressed in the form of: (S_3, Left),(S_2, Right),(S_3, Right), . . .)

Solution:

At S_3, $Q(S_3, left) = Q(S_3, right) = 0$, so we choose left and move to S_2, update $Q(S_3, left) = 0.5 * 0 + 0.5[2 + 0.8 * max(Q(S_2, left), Q(S_2, right))] = 0.5 * (2 + 0.8 * 0) = 1$.

At S_2, $Q(S_2, left) = Q(S_2, right) = 0$, so we choose left and move to S_1, update $Q(S_2, left) = 0.5 * 0 + 0.5[2 + 0.8 * max(Q(S_1, left), Q(S_1, right))] = 0.5 * (2 + 0.8 * 0) = 1$.

At S_1, we can only choose right, so we choose right and move to S_2, update $Q(S_1, right) = 0.5 * 0 + 0.5[1 + 0.8 * max(Q(S_2, left), Q(S_2, right))] = 0.5 * (1 + 0.8 * 1) = 0.9$.

At S_2, $Q(S_2, left) = 1$ and $Q(S_2, right) = 0$, so we choose left and move to S_1, update $Q(S_2, left) = 0.5 * 1 + 0.5[2 + 0.8 * max(Q(S_1, left), Q(S_1, right))] = 0.5 + 0.5 * (2 + 0.8 * 0.9) = 1.86$.

It happens the state oscillate between S_1 and S_2. Below are the 10 state action pairs. $Q(S_3, left)$, $Q(S_2, left)$, $Q(S_1, right)$, $Q(S_2, left)$, $Q(S_1, right)$, $Q(S_2, left)$, $Q(S_1, right)$, $Q(S_2, left)$, $Q(S_1, right)$, $Q(S_2, left)$.

For the subquestions [d]-[e], assume the actions are not deterministic in that after taking an action, it's possible to stay in the same state:

(d) (10 points) Consider executing Value Iteration on this MDP. The transition matrix shown below indicates the probability of transitioning from state s to state s' by taking action a. In the matrix, the first column is the start state and the first row is the ending state. For example, when take action $left$ in state S_2, the probability of transit to S_1 is 0.6 while the probability of stay in S_2 is 0.4. All the empty cell in the matrix means there's no transition.

Action: Left

S_t \ S_{t+1}	S_1	S_2	S_3	S_4	S_5
S_1					
S_2	0.6	0.4			
S_3		0.7	0.3		
S_4			0.6	0.4	
S_5				1	

Action: Right

S_t \ S_{t+1}	S_1	S_2	S_3	S_4	S_5
S_1		1			
S_2		0.3	0.7		
S_3			0.8	0.2	
S_4				0.5	0.5
S_5					

For a given iteration t, the value functions of each state are: $V^t_{S_1} = 20$, $V^t_{S_2} = 30$, $V^t_{S_3} = 20$, $V^t_{S_4} = 30$, $V^t_{S_5} = 10$, compute new value function of all states in the next iteration, $t + 1$, using Value Iteration.

Solution:

The Value Iteration update is below:

$V(S_{t+1}) = \max_a(R_a + \gamma * \sum_a P(S'|S_t, a) * V(S'_t))$

For $V(S_1)$:

$left = 0$

$right = (1 + 0.8 * P(S_2|S_1, right) * V(S_2)) = 1 + 0.8 * 1 * 30 = 25$

$V(S_1) = 25$

For $V(S_2)$:

$left = (2 + 0.8 * P(S_1|S_2, left) * V(S_1) + 0.8 * P(S_2|S_2, left) * V(S_2)) = 2 + 0.8 * 0.6 * 20 + 0.8 * 0.4 * 30 = 21.2$

$right = (1 + 0.8 * P(S_3|S_2, right) * V(S_3) + 0.8 * P(S_2|S_2, right) * V(S_2)) = 1 + 0.8 * 0.7 * 20 + 0.8 * 0.3 * 30 = 19.4$

$V(S_2) = 21.2$

For $V(S_3)$:

$left = (2 + 0.8 * P(S_2|S_3, left) * V(S_2) + 0.8 * P(S_3|S_3, left) * V(S_3)) = 2 + 0.8 * 0.7 * 30 + 0.8 * 0.3 * 20 = 23.6$

$right = (2 + 0.8 * P(S_4|S_3, right) * V(S_4) + 0.8 * P(S_3|S_3, right) * V(S_3)) = 2 + 0.8 * 0.2 * 30 + 0.8 * 0.8 * 20 = 19.6$

$V(S_3) = 23.6$

For $V(S_4)$:

$left = (3 + 0.8 * P(S_3|S_4, left) * V(S_3) + 0.8 * P(S_4|S_4, left) * V(S_4)) = 3 + 0.8 * 0.6 * 20 + 0.8 * 0.4 * 30 = 22.2$

$right = (20 + 0.8 * P(S_5|S_4, right) * V(S_5) + 0.8 * P(S_4|S_4, right) * V(S_4)) = 20 + 0.8 * 0.5 * 10 + 0.8 * 0.5 * 30 = 36$

$V(S_4) = 36$

For $V(S_5)$:

$left = (2 + 0.8 * P(S_4|S_5, left) * V(S_4)) = 2 + 0.8 * 1 * 30 = 26$

$right = 0$
$V(S_5) = 26$

(e) (10 points) Based on the Value function and transition matrix in previous subquestion (d), answer the following questions: 1) what's the optimal policy at time t? 2) what's the optimal policy at $t+1$, after running Value Iteration? 3) Are the two policies different?

Solution:

For $V(S_1)$:
$\pi(S_1) = right$

For $V(S_2)$:
$left = (2 + 0.8 * P(S_1|S_2, left) * V(S_1) + 0.8 * P(S_2|S_2, left) * V(S_2)) = 2 + 0.8 * 0.6 * 25 + 0.8 * 0.4 * 21.2 = 20.784$
$right = (1 + 0.8 * P(S_3|S_2, right) * V(S_3) + 0.8 * P(S_2|S_2, right) * V(S_2)) = 1 + 0.8 * 0.7 * 23.6 + 0.8 * 0.3 * 21.2 = 19.304$
$\pi(S_2) = left$

For $V(S_3)$:
$left = (2 + 0.8 * P(S_2|S_3, left) * V(S_2) + 0.8 * P(S_3|S_3, left) * V(S_3)) = 2 + 0.8 * 0.7 * 21.2 + 0.8 * 0.3 * 23.6 = 19.536$
$right = (2 + 0.8 * P(S_4|S_3, right) * V(S_4) + 0.8 * P(S_3|S_3, right) * V(S_3)) = 2 + 0.8 * 0.2 * 36 + 0.8 * 0.8 * 23.6 = 22.864$
$\pi(S_3) = right$

For $V(S_4)$:
$left = (3 + 0.8 * P(S_3|S_4, left) * V(S_3) + 0.8 * P(S_4|S_4, left) * V(S_4)) = 3 + 0.8 * 0.6 * 23.6 + 0.8 * 0.4 * 36 = 25.848$
$right = (20 + 0.8 * P(S_5|S_4, right) * V(S_5) + 0.8 * P(S_4|S_4, right) * V(S_4)) = 20 + 0.8 * 0.5 * 26 + 0.8 * 0.5 * 36 = 44.8$
$\pi(S_4) = right$

For $V(S_5)$:
$\pi(S_5) = right$

From the answer in (c), the old policy is: $\pi(S_1) = right$, $\pi(S_2) = left$, $\pi(S_3) = left$, $\pi(S_4) = right$, $\pi(S_5) = left$
After Value Iteration, the new policy is: $\pi(S_1) = right$, $\pi(S_2) = left$, $\pi(S_3) = right$, $\pi(S_4) = right$, $\pi(S_5) = left$
So the difference is that the optimal action for state S_3 is changed.