CS 188
Fall 2019

Exam Prep 9 Solutions

Q1. HMMs

Consider a process where there are transitions among a finite set of states $s_1, \cdots, s_k$ over time steps $i = 1, \cdots, N$. Let the random variables $X_1, \cdots, X_N$ represent the state of the system at each time step and be generated as follows:

- Sample the initial state s from an initial distribution $P_1(X_1)$, and set $i = 1$
- Repeat the following:
  1. Sample a duration d from a duration distribution P_D over the integers $\{1, \cdots, M\}$, where M is the maximum duration.
  2. Remain in the current state s for the next d time steps, i.e., set
  $$x_i = x_{i+1} = \cdots = x_{i+d-1} = s \tag{1}$$
  3. Sample a successor state s' from a transition distribution $P_T(X_t|X_{t-1} = s)$ over the other states $s' \neq s$ (so there are no self transitions)
  4. Assign $i = i + d$ and $s = s'$.

This process continues indefinitely, but we only observe the first N time steps.

(a) Assuming that all three states s_1, s_2, s_3 are different, what is the probability of the sample sequence $s_1, s_1, s_2, s_2, s_2, s_3, s_3$? Write an algebraic expression. Assume $M \geq 3$.

$$p_1(s_1)p_D(2)p_T(s_2|s_1)p_D(3)p(s_3|s_2)(1 - p_D(1)) \tag{2}$$

At each time step i we observe a noisy version of the state X_i that we denote Y_i and is produced via a conditional distribution $P_E(Y_i|X_i)$.

(b) Only in this subquestion assume that $N > M$. Let $X_1, \cdots, X_N$ and $Y_1, \cdots, Y_N$ random variables defined as above. What is the maximum index $i \leq N - 1$ so that $X_1 \perp\!\!\!\perp X_N | X_i, X_{i+1}, \cdots, X_{N-1}$ is guaranteed?

$i = N - M$

(c) Only in this subquestion, assume the max duration $M = 2$, and P_D uniform over $\{1, 2\}$ and each x_i is in an alphabet $\{a, b\}$. For $(X_1, X_2, X_3, X_4, X_5, Y_1, Y_2, Y_3, Y_4, Y_5)$ draw a Bayes Net over these 10 random variables with the property that removing any of the edges would yield a Bayes net inconsistent with the given distribution.

(X1) at (0,0) X1; (X2) at (2,-2) X2; (X3) at (4,0) X3; (X4) at (6,-2) X4; (X5) at (8,0) X5; (Y1) at (0,-4)Y1; (Y2) at (2,-4)Y2; (Y3) at (4,-4)Y3; (Y4) at (6,-4)Y4; (Y5) at (8,-4)Y5; (X1) – (X2);(X2) – (X3);(X3) – (X4);(X4) – (X5);(X1) – (Y1);(X2) – (Y2);(X3) – (Y3);(X4) – (Y4);(X5) – (Y5);(X1) – (X3);(X2) – (X4);(X3) – (X5);

(d) In this part we will explore how to write the described process as an HMM with an extended state space. Write the states $z = (s, t)$ where s is a state of the original system and t represents the time elapsed in that state. For example, the state sequence $s_1, s_1, s_1, s_2, s_3, s_3$ would be represented as $(s_1, 1), (s_1, 2), (s_1, 3), (s_2, 1), (s_3, 1), (s_3, 2)$.

Answer all of the following in terms of the parameters $P_1(X_1), P_D(d), P_T(X_{j+1}|X_j), P_E(Y_i|X_i), k$ (total number of possible states), N and M (max duration).

(i) What is $P(Z_1)$?

$$P(x_1, t) = \begin{cases} P_1(x_1) & \text{if } t = 1 \\ 0 & \text{o.w.} \end{cases} \tag{3}$$

(ii) What is $P(Z_{i+1}|Z_i)$? Hint: You will need to break this into cases where the transition function will behave differently.

$$P(X_{i+1}, t_{i+1}|X_i, t_i) = \begin{cases} P_D(d \geq t_i + 1 | d \geq t_i) & \text{when } X_{i+1} = X_i \text{ and } t_{i+1} = t_i + 1 \text{ and } t_{i+1} \leq M \\ P_T(X_{i+1}|X_i) P_D(d = t_i | d \geq t_i) & \text{when } X_{i+1} \neq X_i \text{ and } t_{i+1} = 1 \\ 0 & \text{o.w.} \end{cases}$$

Where $P_D(d \geq t_i + 1 | d \geq t_i) = P_D(d \geq t_i + 1)/P_D(d \geq t_i)$.
Being in X_i, t_i, we know that d was drawn $d \geq t_i$. Conditioning on this fact, we have two choices, if $d > t_i$ then the next state is $X_{i+1} = X_i$, and if $d = t_i$ then $X_{i+1} \neq X_i$ drawn from the transition distribution and $t_{i+1} = 1$. (4)

(iii) What is $P(Y_i|Z_i)$?
$p(Y_i|X_i, t_i) = P_E(Y_i|X_i)$

(e) In this question we explore how to write an algorithm to compute $P(X_N|y_1, \cdots, y_N)$ using the particular structure of this process.

Write $P(X_t|y_1, \cdots, y_{t-1})$ in terms of other factors. Construct an answer by checking the correct boxes below:

$$P(X_t|y_1, \cdots, y_{t-1}) = \underline{\quad \textbf{(i)} \quad} \; \underline{\quad \textbf{(ii)} \quad} \; \underline{\quad \textbf{(iii)} \quad}$$

(i)
- ● $\sum_{i=1}^{k} \sum_{d=1}^{M} \sum_{d'=1}^{M}$
- ○ $\sum_{i=1}^{k} \sum_{d=1}^{M}$
- ○ $\sum_{i=1}^{k}$
- ○ $\sum_{d=1}^{M}$

(ii)
- ○ $P(Z_t = (X_t, d)|Z_{t-1} = (s_i, d))$
- ○ $P(X_t|X_{t-1} = s_i)$
- ○ $P(X_t|X_{t-1} = s_d)$
- ● $P(Z_t = (X_t, d')|Z_{t-1} = (s_i, d))$

(iii)
- ○ $P(Z_{t-1} = (s_d, i)|y_1, \cdots, y_{t-1})$
- ○ $P(X_{t-1} = s_d|y_1, \cdots, y_{t-1})$
- ● $P(Z_{t-1} = (s_i, d)|y_1, \cdots, y_{t-1})$
- ○ $P(X_{t-1} = s_i|y_1, \cdots, y_{t-1})$

(iv) Now we would like to include the evidence y_t in the picture. What would be the running time of each update of the **whole table** $P(X_t|y_1, \cdots, y_t)$?. Assume tables corresponding to any factors used in **(i)**, **(ii)**, **(iii)** have already been computed.

- ○ $O(k^2)$
- ○ $O(k^2M)$
- ● $O(k^2M^2)$
- ○ $O(kM)$

Note: Computing $P(X_N|y_1, \cdots, y_N)$ will take time $N\times$ your answer in **(iv)**.
Just the running time for filtering when the state space is the space of pairs (x_i, t_i),
Given $B_{t-1}(z)$, the step $p(z_t|y_1, \cdots, y_{t-1})$ can be done in time kM. (size of the statespace for z).
The computation to include the y_t evidence can be done in $O(1)$ per z_t.
Therefore each update to the table per evidence point will take $(Mk)^2$. So it is $O((Mk)^2)$.
Using N steps, the whole algorithm will take $O(Nk^2M^2)$ to compute $P(X_N|Y_1, \cdots, Y_N)$.

(v) Describe an update rule to compute $P(X_t|y_1, \cdots, y_{t-1})$ that is faster than the one you discovered in parts **(i)**, **(ii)**, **(iii)**. **Specify its running time**. Hint: Use the structure of the transitions $Z_{t-1} \to Z_t$.

Answer is $O(k^2M + kM)$.
The answer from the previous section is:

$$P(X_t|y_1, \cdots, y_{t-1}) = \sum_{i=1}^{k} \sum_{d=1}^{M} \sum_{d'=1}^{M} P(Z_t = (X_t, d')|Z_{t-1} = (s_i, d))P(Z_{t-1} = (s_i, d)|y_1, \cdots, y_{t-1}) \quad (5)$$

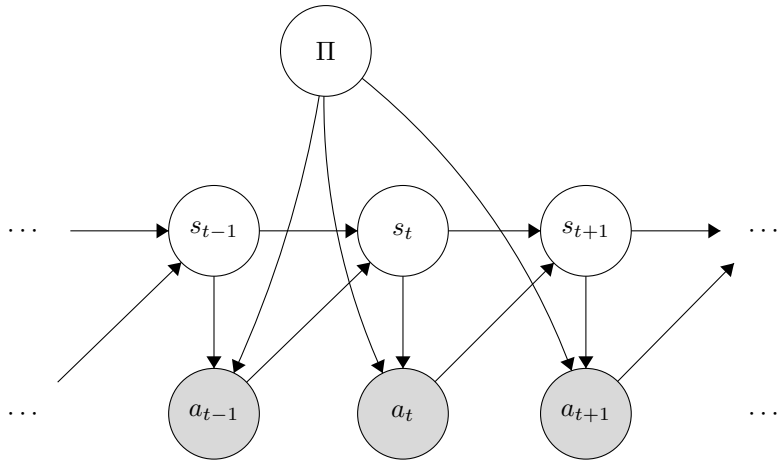
To compute this value we only really need to loop through those transitions $P(Z_t = (X_t, d')|Z_{t-1} = (s_i, d))$ that can happen with nonzero probability.
For all $X_t = s$ we need to sum over all factors of the form $P(Z_t = (s, d')|Z_{t-1} = (s_i, d))P(X_{t-1} = s_i|y_i, \cdots, y_{t-1})$. For a fixed s the factor $P(Z_t = (X_t, d')|Z_{t-1} = (s_i, d))$ can be nonzero only when $s_i = s$ and $d' = d + 1$ (M tuples). And when $s_i \neq s$ and $d' = 1$ and $d = 1, \cdots, M$ (kM tuples).
Since this needs to be performed for all k possible values of s, the answer to update the whole table is $O(k^2M + kM)$.

Q2. Particle Filtering Apprenticeship

Consider a modified version of the apprenticeship problem. We are observing an agent's actions in an MDP and are trying to determine which out of a set $\{\pi_1, \ldots, \pi_n\}$ the agent is following. Let the random variable Π take values in that set and represent the policy that the agent is acting under. We consider only *stochastic* policies, so that A_t is a random variable with a distribution conditioned on S_t and Π. As in a typical MDP, S_t is a random variable with a distribution conditioned on S_{t-1} and A_{t-1}. The full Bayes net is shown below.

The agent acting in the environment knows what state it is currently in (as is typical in the MDP setting). Unfortunately, however, we, the observer, cannot see the states S_t. Thus we are forced to use an adapted particle filtering algorithm to solve this problem. Concretely, we will develop an efficient algorithm to estimate $P(\Pi \mid a_{1:t})$.

(a) The Bayes net for part (a) is



(i) Select all of the following that are guaranteed to be true in this model for $t > 10$:

- ☐ $S_t \perp\!\!\!\perp S_{t-2} \mid S_{t-1}$
- ■ $S_t \perp\!\!\!\perp S_{t-2} \mid S_{t-1}, A_{1:t-1}$
- ☐ $S_t \perp\!\!\!\perp S_{t-2} \mid \Pi$
- ☐ $S_t \perp\!\!\!\perp S_{t-2} \mid \Pi, A_{1:t-1}$
- ■ $S_t \perp\!\!\!\perp S_{t-2} \mid \Pi, S_{t-1}$
- ■ $S_t \perp\!\!\!\perp S_{t-2} \mid \Pi, S_{t-1}, A_{1:t-1}$
- ☐ None of the above

We will compute our estimate for $P(\Pi \mid a_{1:t})$ by coming up with a recursive algorithm for computing $P(\Pi, S_t \mid a_{1:t})$. (We can then sum out S_t to get the desired distribution; in this problem we ignore that step.)

(ii) Write a recursive expression for $P(\Pi, S_t \mid a_{1:t})$ in terms of the CPTs in the Bayes net above. Hint: Think of the forward algorithm.

$$P(\Pi, S_t \mid a_{1:t}) \propto \sum_{s_{t-1}} P(\Pi, s_{t-1} \mid a_{1:t-1}) P(a_t \mid S_t, \Pi) P(S_t \mid s_{t-1}, a_{t-1})$$

We now try to adapt particle filtering to approximate this value. Each particle will contain a single state s_t and a potential policy π_i.

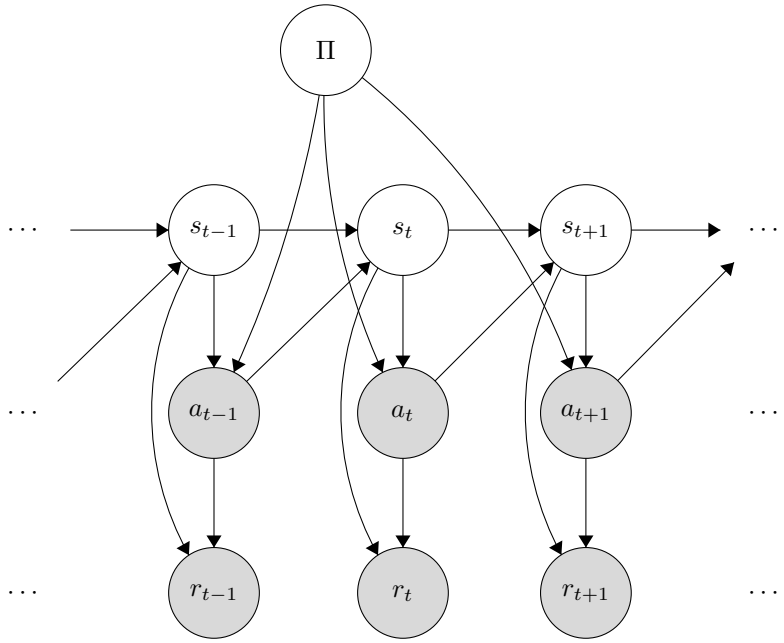
(iii) The following is pseudocode for the body of the loop in our adapted particle filtering algorithm. Fill in the boxes with the correct values so that the algorithm will approximate $P(\Pi, S_t \mid a_{1:t})$.

1. Elapse time: for each particle (s_t, π_i), sample a successor s_{t+1} from $\boxed{P(S_{t+1} \mid s_t, a_t)}$. The policy π' in the new particle is $\boxed{\pi_i}$.

2. Incorporate evidence: To each new particle (s_{t+1}, π'), assign weight $\boxed{P(a_{t+1} \mid s_{t+1}, \pi')}$.

3. Resample particles from the weighted particle distribution.

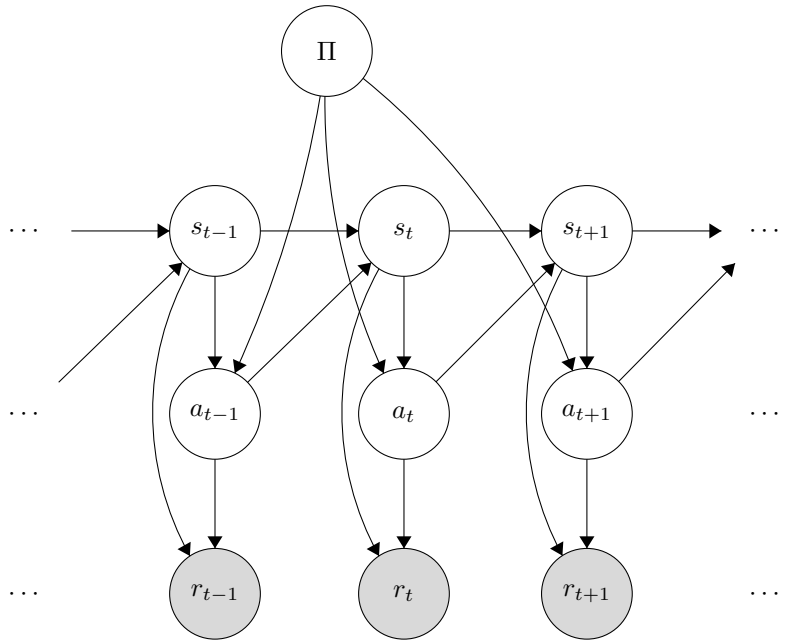
(b) We now observe the acting agent's actions *and* rewards at each time step (but we still don't know the states). Unlike the MDPs in lecture, here we use a stochastic reward function, so that R_t is a random variable with a distribution conditioned on S_t and A_t. The new Bayes net is given by



Notice that the observed rewards do in fact give useful information since d-separation does not give that $R_t \perp\!\!\!\perp \Pi \mid A_{1:t}$. Give an active path connecting R_t and Π when $A_{1:t}$ are observed. Your answer should be an ordered list of nodes in the graph, for example "$S_t, S_{t+1}, A_t, \Pi, A_{t-1}, R_{t-1}$".

R_t, S_t, A_t, Π. This list reversed is also correct, and many other similar (though more complicated) paths are also correct.

(c) We now observe *only* the sequence of rewards and no longer observe the sequence of actions. The new Bayes net is:



We will compute our estimate for $P(\Pi \mid r_{1:t})$ by coming up with a recursive algorithm for computing $P(\Pi, S_t, A_t \mid r_{1:t})$. (We can then sum out S_t and A_t to get the desired distribution; in this problem we ignore that step.)

(i) Write a recursive expression for $P(\Pi, S_t, A_t \mid r_{1:t})$ in terms of the CPTs in the Bayes net above.

$$P(\Pi, S_t, A_t \mid r_{1:t}) \propto \sum_{s_{t-1}} \sum_{a_{t-1}} P(\Pi, s_{t-1}, a_{t-1} \mid r_{1:t-1}) P(A_t \mid S_t, \Pi) P(S_t \mid s_{t-1}, a_{t-1}) P(r_t \mid S_t, A_t)$$

We now try to adapt particle filtering to approximate this value. Each particle will contain a single state s_t, a single action a_t, and a potential policy π_i.

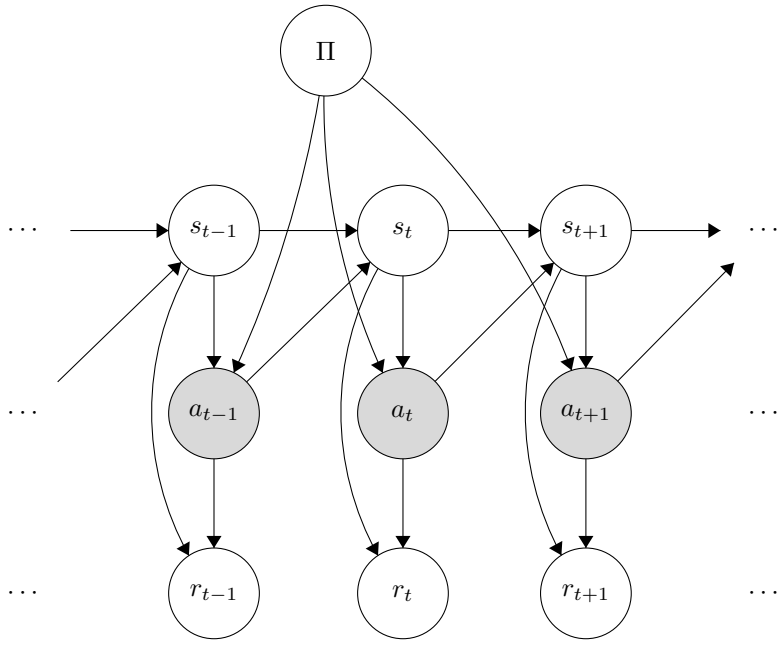
(ii) The following is pseudocode for the body of the loop in our adapted particle filtering algorithm. Fill in the boxes with the correct values so that the algorithm will approximate $P(\Pi, S_t, A_t \mid r_{1:t})$.

1. Elapse time: for each particle (s_t, a_t, π_i), sample a successor state s_{t+1} from $\boxed{P(S_{t+1} \mid s_t, a_t)}$. Then, sample a successor action a_{t+1} from $\boxed{P(A_{t+1} \mid s_{t+1}, \pi_i)}$. The policy π' in the new particle is $\boxed{\pi_i}$.

2. Incorporate evidence: To each new particle (s_{t+1}, a_{t+1}, π'), assign weight $\boxed{P(r_{t+1} \mid s_{t+1}, a_{t+1})}$.

3. Resample particles from the weighted particle distribution.

(d) Finally, consider the following Bayes net:



Here, the task is identical to that in part (a); we see only the actions and want to approximate $P(\Pi, \mid a_{1:t})$. However, now we are also accounting for the hidden reward variables.

(i) For a fixed state action pair (s_t, a_t), what is $\sum_{r_t} P(r_t \mid s_t, a_t)$?

1

Suppose for the following questions we adapt particle filtering to this model as in previous parts. In particular, in this algorithm, our particles will also track r_t values.

(ii) Comparing to the algorithm in (a), with the same number of particles, this algorithm will give an estimate of $P(\Pi \mid a_{1:t})$ that is

○ More accurate ● Equally accurate ○ Less accurate

(iii) Comparing to the algorithm in (a), with the same number of particles, to compute an estimate, this algorithm will take

● More time ○ The same amount of time ○ Less time