Q1. Planning ahead with HMMs

Pacman is tired of using HMMs to estimate the location of ghosts. He wants to use HMMs to plan what actions to take in order to maximize his utility. Pacman uses the HMM (drawn to the right) of length $T$ to model the planning problem. In the HMM, $X_{1:T}$ is the sequence of hidden states of Pacman’s world, $A_{1:T}$ are actions Pacman can take, and $U_t$ is the utility Pacman receives at the particular hidden state $X_t$. Notice that there are no evidence variables, and utilities are not discounted.

(a) The belief at time $t$ is defined as $B_t(X_t) = p(X_t | a_{1:t})$. The forward algorithm update has the following form:

$$B_t(X_t) = \boxed{\text{(i)}} \hspace{1cm} \boxed{\text{(ii)}} \hspace{1cm} B_{t-1}(x_{t-1})$$

Complete the expression by choosing the option that fills in each blank.

(i) $\max_{x_{t-1}} \hspace{1cm} \sum_{x_{t-1}} \hspace{1cm} \max_{x_t} \hspace{1cm} \sum_{x_t} \hspace{1cm} 1$

(ii) $p(X_t | x_{t-1}) \hspace{1cm} p(X_t | x_{t-1}) p(X_t | a_t) \hspace{1cm} p(X_t) \hspace{1cm} p(X_t | x_{t-1}, a_t) \hspace{1cm} 1$

None of the above combinations is correct

(b) Pacman would like to take actions $A_{1:T}$ that maximizes the expected sum of utilities, which has the following form:

$$\text{MEU}_{1:T} = \boxed{\text{(i)}} \hspace{1cm} \boxed{\text{(ii)}} \hspace{1cm} \boxed{\text{(iii)}} \hspace{1cm} \boxed{\text{(iv)}} \hspace{1cm} \boxed{\text{(v)}}$$

Complete the expression by choosing the option that fills in each blank.

(i) $\max_{a_{1:T}} \hspace{1cm} \max_{a_T} \hspace{1cm} \sum_{a_{1:T}} \hspace{1cm} \sum_{a_T} \hspace{1cm} 1$

(ii) $\max_{a_t} \hspace{1cm} \prod_{t=1}^T \hspace{1cm} \sum_{t=1}^T \hspace{1cm} \min_t \hspace{1cm} 1$

(iii) $\sum_{x_t, a_t} \hspace{1cm} \sum_{x_t} \hspace{1cm} \sum_{a_t} \hspace{1cm} \sum_{x_T} \hspace{1cm} 1$

(iv) $p(x_t | x_{t-1}, a_t) \hspace{1cm} p(x_t) \hspace{1cm} B_t(x_t) \hspace{1cm} B_T(x_T) \hspace{1cm} 1$

(v) $U_T \hspace{1cm} \frac{1}{U_t} \hspace{1cm} \frac{1}{U_T} \hspace{1cm} U_t \hspace{1cm} 1$

None of the above combinations is correct

(c) A greedy ghost now offers to tell Pacman the values of some of the hidden states. Pacman needs your help to figure out if the ghost’s information is useful. Assume that the transition function $p(x_t | x_{t-1}, a_t)$ is not deterministic. With respect to the utility $U_t$, mark all that can be True:

- $\checkmark$ VPI($X_{t-1}$ | $X_{t-2}$) > 0
- $\checkmark$ VPI($X_{t-2}$ | $X_{t-1}$) > 0
- VPI($X_{t-1}$ | $X_{t-2}$) = 0
- VPI($X_{t-2}$ | $X_{t-1}$) = 0

None of the above

(d) Pacman notices that calculating the beliefs under this model is very slow using exact inference. He therefore decides to try out various particle filter methods to speed up inference. Order the following methods by how accurate their estimate of $B_T(X_T)$ is? If different methods give an equivalently accurate estimate, mark them as the same number.
<table>
<thead>
<tr>
<th>Method</th>
<th>Most accurate</th>
<th>Least accurate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact inference</td>
<td>○ 1</td>
<td>○ 4</td>
</tr>
<tr>
<td>Particle filtering with no resampling</td>
<td>○ 1</td>
<td>○ 4</td>
</tr>
<tr>
<td>Particle filtering with resampling before every time elapse</td>
<td>○ 1</td>
<td>○ 4</td>
</tr>
<tr>
<td>Particle filtering with resampling before every other time elapse</td>
<td>○ 1</td>
<td>○ 4</td>
</tr>
</tbody>
</table>
Q2. Particle Filtering Apprenticeship

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 \( \Pi \) 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 \( \Pi \). 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 \( P(\Pi | a_1:t) \).

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

```
          \( \Pi \)
         /   \
\( \ldots \)  \( s_{t-1} \)  \( s_t \)  \( s_{t+1} \)  \( \ldots \)
         \( \ldots \)
```

We will compute our estimate for \( P(\Pi | a_1:t) \) by coming up with a recursive algorithm for computing \( P(\Pi, S_t | a_1:t) \). (We can then sum out \( S_t \) to get the desired distribution; in this problem we ignore that step.)

(i) Write a recursive expression for \( P(\Pi, S_t | a_1:t) \) in terms of the CPTs in the Bayes net above.

\[ P(\Pi, S_t | a_1:t) \propto \]

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

(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 \( P(\Pi, S_t | a_1:t) \).

1. Elapse time: for each particle \((s_t, \pi_i)\), sample a successor \( s_{t+1} \) from \( \ldots \).
   The policy \( \pi' \) in the new particle is \( \ldots \).
2. Incorporate evidence: To each new particle \((s_{t+1}, \pi')\), assign weight \( \ldots \).
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

```
          \( \Pi \)
         /   \
\( \ldots \)  \( s_{t-1} \)  \( s_t \)  \( s_{t+1} \)  \( \ldots \)
         \( \ldots \)
```

(i) Write a recursive expression for \( P(\Pi, S_t | a_1:t, r_1:t) \) in terms of the CPTs in the Bayes net above.

\[ P(\Pi, S_t | a_1:t, r_1:t) \propto \]

3
We now observe only the sequence of rewards and no longer observe the sequence of actions. The new Bayes net is:

![Bayes Net Diagram]

(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.

$$P(\Pi, S_t, A_t \mid r_{1:t}) \propto$$

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 $\pi_i$.

(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 $(s_t, a_t, \pi_i)$, sample a successor state $s_{t+1}$ from ________________.
   Then, sample a successor action $a_{t+1}$ from ________________.
   The policy $\pi'$ in the new particle is ________________.

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

3. Resample particles from the weighted particle distribution.