CS 188
Spring 2023

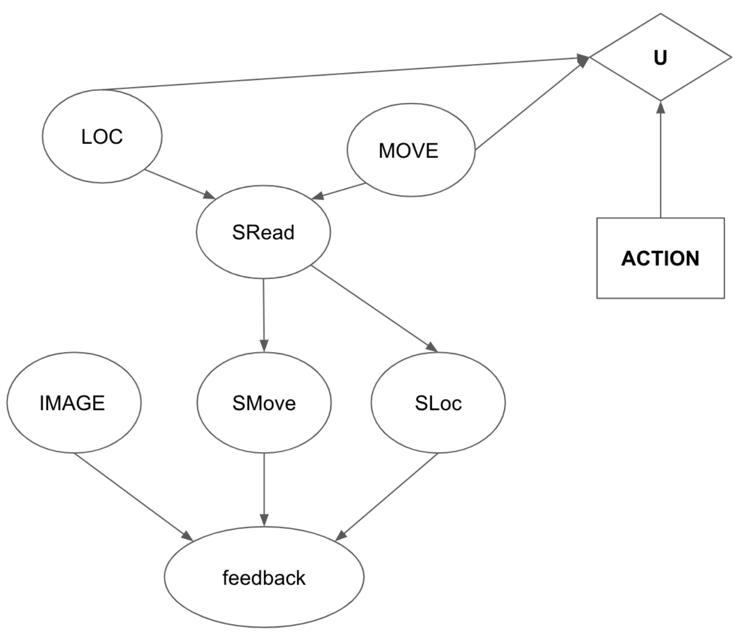
Final Review: HMMs & VPI Solutions

Q1. Vehicle Perception Indication

A vehicle is trying to identify the situation of the world around it using a set of sensors located around the vehicle.

Each sensor reading (SRead) is based off of an object's location (LOC) and an object's movement (MOVE). The sensor reading will then produce various values for its predicted location (SLoc) and predicted movement (SMove). The user will receive these readings, as well as the the image (IMAGE) as feedback.

(a) The vehicle takes an action, and we assign some utility to the action based on the object's location and movement. Possible actions are MOVE TOWARDS, MOVE AWAY, and STOP. Suppose the decision network faced by the vehicle is the following.



(i) Based on the diagram above, which of the following **could possibly be** true?

- ■ VPI (Image) = 0
- ☐ VPI (SRead) < 0
- ☐ VPI (SMove, SRead) > VPI (SRead)
- ■ VPI (Feedback) = 0
- ○ None of the above

VPI(Image) = 0 because there is not active path connecting Image and U

VPI cannot be negative, so option 2 is not selected.

VPI(SMove, SRead) = VPI(SMove | SRead) + VPI(SRead), therefore we can cancel VPI(SRead) from both side, and it becomes asking if VPI(SMove | SRead) > 0. And we can see that cannot be true, because shading in SRead, there is no active path connecting SMove and U.

There is an active path connecting Feedback and U, therefore VPI(Feedback) ≥ 0. It could still be 0 because active path only gives the possibility of > 0.

(ii) Based on the diagram above, which of the following **must necessarily be** true?

- ■ VPI (Image) $= 0$
- ☐ VPI (SRead) $= 0$
- ■ VPI (SMove, SRead) $=$ VPI (SRead)
- ☐ VPI (Feedback) $= 0$
- ○ None of the above

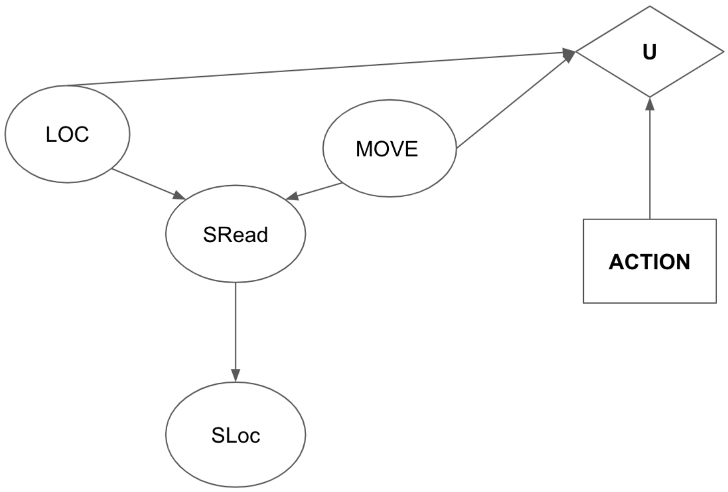
VPI(Image) = 0 because there is not active path connecting Image and U

VPI(SRead) could be > 0 because SRead-MOVE-U is an active path between SRead and U

VPI(SMove, SRead) = VPI(SMove | SRead) + VPI(SRead), therefore we can cancel VPI(SRead) from both side, and it becomes asking if VPI(SMove | SRead) == 0. And we can see that must true, because shading in SRead, there is no active path connecting SMove and U.

VPI(Feedback) could be > 0 because feedback-SLoc-SRead-MOVE-U is an active path

Let's assume that your startup has less money, so we use a simpler sensor network. One possible sensor network can be represented as follows.



You have distributions of $P(\text{LOC})$, $P(\text{MOVE})$, $P(SRead|\text{LOC}, \text{MOVE})$, $P(SLoc|SRead)$ and utility values $U(a, l, m)$.

(b) Complete the equation for determining the expected utility for some ACTION a.

$EU(a) = \left(\underline{\quad\textbf{(i)}\quad} \ \underline{\quad\textbf{(ii)}\quad} \ \underline{\quad\textbf{(iii)}\quad} \right) \ U(a, l, m)$

(i) ● $\sum_l P(l)$ ○ $\sum_{sloc} P(sloc|l)$ ○ $\sum_l \sum_{sloc} P(sloc|l)$ ○ 1

(ii) ● $\sum_m P(m)$ ○ $\sum_m P(sloc|m)$ ○ $\sum_l \sum_m \sum_{sloc} P(sloc|l)P(sloc|m)$ ○ 1

(iii) ○ $*\sum_l \sum_m \sum_{sloc} P(sloc|l)P(sloc|m)$ ○ $+\sum_l \sum_m \sum_{sloc} P(sloc|l)P(sloc|m)$
○ $+\sum_l \sum_m \sum_{sloc} P(sloc|l, m)P(l)P(m)$ ● $*1$

$EU(a) = \sum_l P(l) \sum_m P(m)U(a, l, m)$

We can eliminate SRead and SLoc via marginalization, so they don't need to be included the expression

(c) Your colleague Bob invented a new sensor to observe values of $SLoc$.

(i) Suppose that your company had no sensors till this point. Which of the following expression is equivalent to VPI($SLoc$)?

■ $\text{VPI}(SLoc) = (\sum_{sloc} P(sloc)\ \text{MEU}(SLoc = sloc)) - \max_a \text{EU}(a)$

■ $\text{VPI}(SLoc) = \text{MEU}(SLoc) - \text{MEU}(\emptyset)$

□ $\text{VPI}(SLoc) = \max_{sloc} \text{MEU}(SLoc = sloc) - \text{MEU}(\emptyset)$

○ None of the above

Option 2 is correct by definition, and option 1 is the expanded version of option 2

(ii) Gaagle, an established company, wants to sell your startup a device that gives you $SRead$. Given that you already have Bob's device (that gives you $SLoc$), what is the maximum amount of money you should pay for Gaagle's device? Suppose you value \$1 at 1 utility.

□ $\text{VPI}(SRead)$

■ $\text{VPI}(SRead) - \text{VPI}(SLoc)$

□ $\text{VPI}(SRead, SLoc)$

■ $\text{VPI}(SRead, SLoc) - \text{VPI}(SLoc)$

○ None of the above

Choice 4 is correct by definition

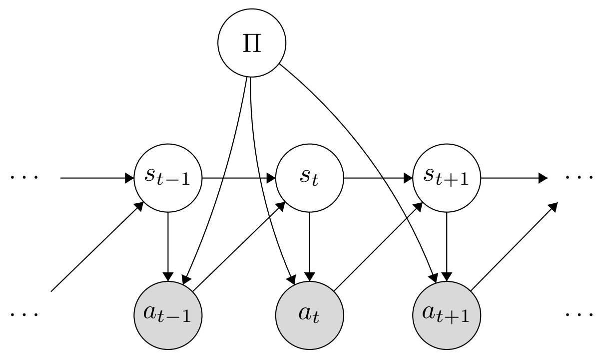
Choice 2 is true because VPI(SLoc | SRead) = 0, and thus
VPI(SRead) = VPI(SRead) + 0 = VPI(SRead) + VPI(SLoc | SRead) = VPI(SRead, SLoc), which makes choice 2 the same as choice 4

2 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 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 $t > 3$:

- [ ] $S_t \perp\!\!\!\perp S_{t-2} \mid S_{t-1}$
- [x] $S_t \perp\!\!\!\perp S_{t-2} \mid S_{t-1}, A_{1:t-1}$
- [ ] $S_t \perp\!\!\!\perp S_{t-2} \mid \Pi$
- [ ] $S_t \perp\!\!\!\perp S_{t-2} \mid \Pi, A_{1:t-1}$
- [x] $S_t \perp\!\!\!\perp S_{t-2} \mid \Pi, S_{t-1}$
- [x] $S_t \perp\!\!\!\perp S_{t-2} \mid \Pi, S_{t-1}, A_{1:t-1}$
- [ ] 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.

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

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 $P(\Pi, S_t \mid a_{1:t})$.

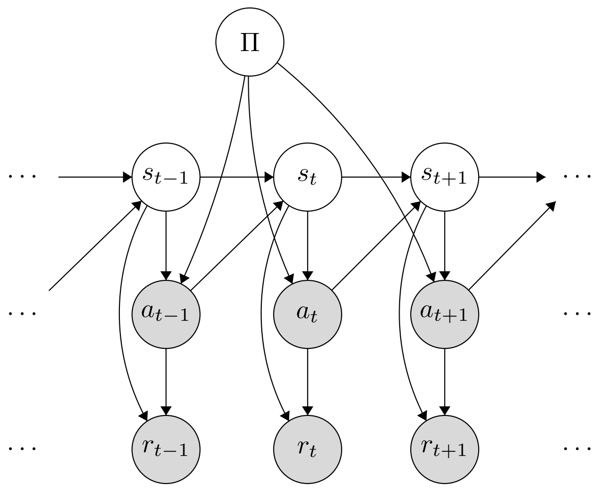
1. Elapse time: for each particle (s_t, π_i), sample a successor s_{t+1} from $P(S_{t+1} \mid s_t, a_t)$.

   The policy π' in the new particle is π_i .

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

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 $R_t \perp\!\!\!\perp \Pi \mid A_{1:t}$.

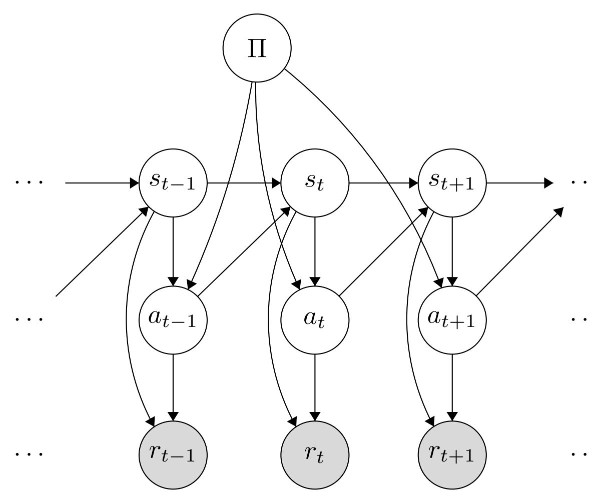
(i) Give an active path connecting R_t and Π when $A_{1:t}$ are observed. Your answer should be an ordered list of nodes in the graph, for example "$S_t, S_{t+1}, A_t, \Pi, A_{t-1}, R_{t-1}$".

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

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

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

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



(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 $P(S_{t+1} \mid s_t, a_t)$.

   Then, sample a successor action a_{t+1} from $P(A_{t+1} \mid s_{t+1}, \pi_i)$.

   The policy π' in the new particle is π_i.

2. Incorporate evidence: To each new particle (s_{t+1}, a_{t+1}, π'), assign weight $P(r_{t+1} \mid s_{t+1}, a_{t+1})$.
3. Resample particles from the weighted particle distribution.