ASEN 6519-007 Decision Making under Uncertainty
Homework 2: Markov Decision Processes

April 1, 2020

1 Conceptual Questions

Question 1. (Particle Filtering, 25 pts) Consider a 3-door version of the tiger POMDP with

$$S = \{1,2,3\}, \quad A = \{1,2,3,4\}, \quad O = \{1,2,3\}$$

where the state is the door that the tiger is behind, actions 1, 2, and 3 correspond to opening one of the doors, action 4 corresponds to a listen action, and each observation corresponds to a noisy indication that the tiger is behind the corresponding door. Let the reward function be

$$R(s,a) = \begin{cases} 0 & \text{if } a = 4 \quad \text{(listening)} \\ -100 & \text{if } s = a \quad \text{(opening tiger door)} \\ 10 & \text{otherwise} \quad \text{(opening non-tiger door)} \end{cases}$$

At each step, the tiger has a 5% chance of moving one door up (wrapping from 3 back around to 1), so, if v is a uniformly-distributed random variable between 0 and 1, a state generative model is defined by G below. If the listen action is taken, the observation is accurate 80% of the time, so the observation model *for* $a = 4$ is given by $\mathcal{Z}$ below.

$$s' = G(s,a,v) = \begin{cases} s+1 \pmod 3 & \text{if } v < 0.05 \\ s & \text{otherwise} \end{cases} \qquad \mathcal{Z}(o \mid a=4, s') = \begin{cases} 0.8 & \text{if } o = s' \\ 0.1 & \text{otherwise} \end{cases} \tag{1}$$

Suppose that you decide to use a particle filter with 3 particles ($|\hat{b}| = 3$) for approximate belief updates. The initial particle-based belief approximation is $\hat{b}_0 = \{1,2,3\}$. Perform a single belief update (following Algorithm 6.3) with $a = 4$, $o = 1$, using G from (1) in Alg 6.3, line 5. Use the following outcomes for the random parts of the algorithm:

- In line 4: $s_1 = 1$, $s_2 = 2$, $s_3 = 3$
- In line 5, the values of v input to G are: $v_1 = 0.01$, $v_2 = 0.32$, $v_3 = 0.74$.
- In line 8, select the new states however you would like.

(a) List the state particles in the updated belief approximation, $\hat{b}_1$.[1]

(b) If the true belief were $\hat{b}_1$, what would the optimal action be? Given that $\hat{b}_1$ is only an approximation of the true belief, why is this action actually a poor choice? How would you change the belief update approach to avoid this pitfall?

[1] there are multiple correct answers because of the randomness in line 8

2 Exercises

Question 2. (SARSOP vs QMDP on cancer and tiger POMDPs, 25 pts) In this problem, you will compare optimal solutions calculated with SARSOP to suboptimal QMDP solutions on the cancer POMDP from Homework 5 and the `TigerPOMDP` from `POMDPModels.jl`. Use the solvers from `SARSOP.jl` and `QMDP.jl` and Monte Carlo policy evaluation to fill in the following table with the value $V^{\pi}(b_0)$:

	Cancer	Tiger
SARSOP		
QMDP		

Both problems have information gathering actions, however in one problem QMDP is nearly optimal and in the other it is significantly suboptimal. Explain why this is the case.

3 Challenge Problem

Question 3. (Lasertag POMDP, 50 pts) In this problem, you will find a policy and belief updater for the laser tag POMDP model `DMUStudent.HW6.lasertag`.

In this POMDP a robot seeks to tag a moving target in a grid world with obstacles. The state space consists of all positions the robot and the target can take, and the problem ends with a reward of 100 as soon as they occupy the same cell. There are five actions, `:up`, `:down`, `:left`, `:right`, and `:measure`. The `:measure` action has a cost of -2 and gives the robot returns from lasers pointed in the four directions indicating the *exact* distance to the first wall, obstacle, or target that the laser encounters. When any other action is taken, the reward is -1 and the laser return for each direction is noisy; it is uniformly distributed between 0 and the distance to the nearest object. The `POMDPs.jl` interface including `POMDPModelTools.render` can be used to further explore the problem.

Using the `DMUStudent.submit` function, you must submit either a `POMDPs.Policy` or a tuple containing a `POMDPs.Policy` and a `POMDPs.Updater`. A score of 45 or greater will receive full credit (a score above 40 also good and will receive 90% credit). You are encouraged to use any tools available to you - any POMDPs.jl solvers, deep reinforcement learning, a particle filter from ParticleFilters.jl, or heuristic policies are all acceptable.