ASEN 5519-003 Decision Making under Uncertainty
Homework 2: Markov Decision Processes

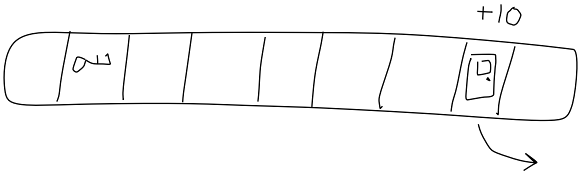
January 21, 2021

1 Conceptual Questions

Question 1. Give an example of an MDP with a unique nonzero optimal value function, but multiple optimal policies.[1]

Question 2. Consider a stationary infinite-horizon MDP with $\mathcal{S} = \{1, 2\}$, $\mathcal{R}(s, a) = s^2$, and $\gamma = 0.9$ ($\mathcal{A}$ and $\mathcal{T}$ are unknown). Suppose that policy π achieves a value at state 1 of $V^\pi(1) = 37$. What is $V^*(2)$, the optimal value at state 2? Justify your answer.

Question 3. Consider a 1 by 9 grid world with a key in cell 2 and a locked door in cell 8. An agent automatically receives a reward of 10 upon returning to cell 8 after having collected the key, and the problem terminates immediately. At each time step, the agent can take one of two actions, `left`, or `right` which always succeeds in moving the desired direction. If an end wall is hit, the agent bounces back to the previous cell. The discount factor is $\gamma = 0.95$.



a Formulate this problem as an MDP, and write down the state space $\mathcal{S}$, reward function $\mathcal{R}$, and transition distributions $\mathcal{T}$.

b What is the value of being in cell 9 without the key?

2 Exercise

Question 4. (Value iteration for Grid World)

Solve the MDP `HW2.grid_world` with your own implementation of value iteration with a discount of $\gamma = 0.95$ and plot the resulting value function. All of the necessary information to solve this problem can be extracted with the `HW2.transition_matrices` and `HW2.reward_vectors` functions, and plotting can be accomplished with `HW2.render(HW2.grid_world, v)` where `v` is the value function. See the starter code and function docstrings for more information.

[1] Hint: you can do this with $|\mathcal{S}| = 1$.

3 Challenge Problem

Question 5. (Value iteration for ACAS)

Your task is to find the optimal value function for an Aircraft Collision Avoidance System (ACAS). The encounter model will be specified as a Markov decision process, and your task will be to compute the value function for discount $\gamma = 0.99$ using value iteration or another suitable algorithm that you implement. The continuous physical state space will be discretized at various levels of granularity and the goal is to find the value function for the finest discretization possible.

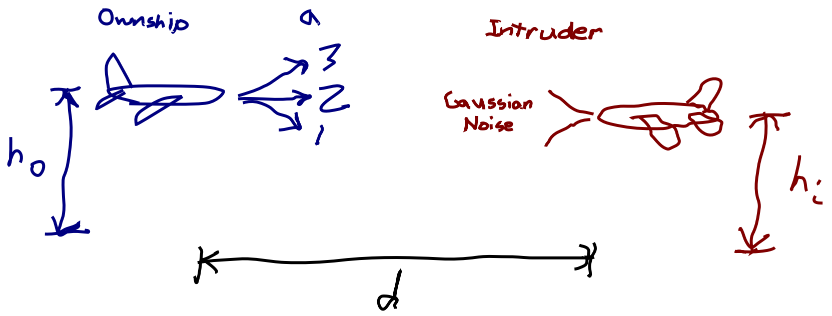
A model with discretization level `n` can be constructed with

```
m = HW2.UnresponsiveACASMDP(n)
```

The higher `n` is, the finer the discretization and the larger the state space. Again, all of the information needed to solve this problem can be extracted with the `HW2.transition_matrices` and `HW2.reward_vectors` functions, so you can start with your code from Question 4.

The score received for solving the problem is `n`. You must submit your code for this problem along with the `results.json` file from executing `HW2.evaluate(v, "email@colorado.edu")` where `v` is the value function vector. A score of `n` = 7 or higher will receive full credit[2].

The information above this line is sufficient to complete this homework. However, the description of the model below may help to get the highest score on the leaderboard. The `UnresponsiveACASMDP` model implements the POMDPs.jl explicit MDP interface[3], and students are welcome to explore the problem further using that interface as well as ask on Piazza and look at the source code for details. The underlying continuous model is defined as follows:



- The state space is 4-dimensional $\mathcal{S} = \mathbb{R}^4$, with each state consisting of $s = (h_o, \dot{h}_o, h_i, d)$ where h_o is the ownship altitude in feet, $\dot{h}_o$ is the rate of climb in ft/min, h_i is the intruder altitude in feet, and d is the distance between the aircraft in feet.
- The action space is $\mathcal{A} = \{-1500, 0, 1500\}$ and represents the change in rate of climb. The possible rates of climb are $\dot{h}_o \in \{-3000, -1500, 0, 1500, 3000\}$
- A reward of -100 is received for a near-mid-air collision, defined as the aircraft passing within 500 vertical feet and 100 horizontal feet of each other. Any change in rate of climb yields a reward of -1.

[2] By taking advantage of the structure of the problem, it is possible to attain a score of `n` = 20 with less than 10 minutes of computation time on a single core of a i7 laptop processor.

[3] `https://juliapomdp.github.io/POMDPs.jl/stable/dynamics/#Separate-explicit-or-generative-definition`

- The rate of climb, $\dot{h}_o$ changes instantly when an action is applied. Then the following dynamics are used: $d' = d - 2\,v\,\Delta t$ where v is the fixed horizontal velocity, $h_o' = h_o + \dot{h}_o\,\Delta t$, and $h_i' = h_i + W_{\Delta t \sigma^2}$ where W is the Wiener process[4]. Δt changes based on the discretization `n`.

[4] `https://en.wikipedia.org/wiki/Wiener_process`