# Assignment 5: Partially Observable Markov Decision Processes

## 1 Introduction

Recall that an MDP is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$ , where $\mathcal{S}$ is the state set, $\mathcal{A}$ is the action set, $\mathcal{T}$ is the transition, a mapping from $\mathcal{S} \times \mathcal{A}$ to a probability distribution over the $\mathcal{S}$ with non-zero probabilities assigned to possible next states and $\mathcal{R}$ is a reward function mapping $\mathcal{S} \times \mathcal{A}$ to the real numbers corresponding to the instantaneous rewards received for that action in that state.

When the state is not completely observable, we instead model observations. Here, because there is some uncertainty in which state we are in, we use the observations gathered from taking an action in whatever state we might be in to augment the information that we currently have about our state position. Now we have a finite set of possible observations $\Omega$, mapping $\mathcal{A} \times \mathcal{S}$ into discrete probability distributions over $\Omega$. We also have $\mathcal{O}$, a function of observation probabilities $P(o|s, a)$, for the probability of making observation o from state s after having taken action a. Thus our POMDP is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O}, \gamma \rangle$

Instead of solving this, we introduce a probabilistic representation of the agent's internal belief of what state it is in. Each belief state is a distribution over the entire state set where each probability corresponds to the strength of the agent's belief it is in a particular state. To solve the POMDP, we can model it as an MDP over the belief states. Here, the set of all possible belief states is the new state set of the MDP, the actions remain the same, and we need only create a transition over beliefs and a reward funciton over beliefs. However, even if the original state space of the POMDP was discrete, the new belief state space consists of all possible mixtures of beliefs about which states the agent is most likely to be in, making it continuous. Continuous MDPs can no longer be solved by simple value iteration, and require approximation methods.

To avoid this issue in the homework, we constrain the belief space to have a small, finite number of possibilities, making it a special case, still solvable by our value iteration algorithm.

Additionally, the new reward function can be derived as:

$$\rho(b, a) = \sum_{s \in \mathcal{S}} R(s, a) b(s) \tag{1}$$

Essentially, the summed reward of being in each possible state and taking the action, weighted by how likely being in that particular state is.

The new transition is now a belief transition:

$$\tau(b, a, b') = \sum_{o \in \Omega} Pr(b'|b, a, o) Pr(o|a, b) \tag{2}$$

where $Pr(b'|b, a, o)$ is an indicator function of when $SE(b, a, o) = b'$ and:

$$SE_{s'}(b, a, o) = \frac{Pr(o|s, a) \sum_{s \in \mathcal{S}} T(s, a, s') b(s)}{Pr(o|a, b)} \tag{3}$$

See lecture slides, Wikipedia's POMDP page, or Cassandra, Kaelbling, and Littman 1994 for derivation and details.

## 2 Setup

Consider the environment in Figure 1. There are two possible goal states, labeled 1 and 2 in green. One will give a reward and the other a large cost, but the agent does not know which is which to begin with. Each action also incurs a small cost. In the environment, there is also a sign, visible to the agent from all of the orange states below the barrier. Once the agent sees the sign, it receives the observation of the correct goal state with probability 1. We are interested in creating a policy mapping belief states to actions for each state in this environment.

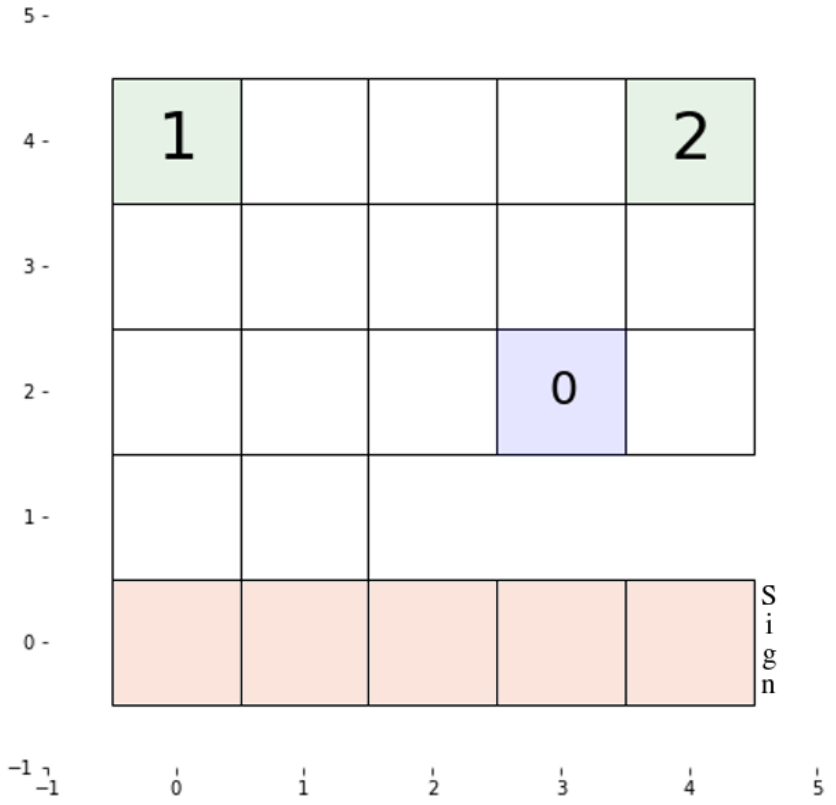


Figure 1: POMDP Environment

Based on this policy, we would expect that if the agent started in the initial blue state labeled 0 with no information about which goal was the rewarding one, it should take information seeking actions, as the cost of gathering that information is much lower than the risk of traveling to the wrong goal.

In this problem, we make some simplifying assumptions: first that the position transition function is deterministic and fully observable. That is if an agent takes action (-1, 0) in position (3,2), its next position is (2,2) with probability 1 and it knows this. There is no chance that it will move to any other position. Second, the agent's observation of the sign is perfect. The agent sees what is written on the sign and reads it correctly with probability 1 from any of the orange spaces. This allows us to hugely reduce the belief state space.

The the state can now be thought of as having two distinct pieces: a position (xy-coordinate) and a world (either the reward is in position 1 or the reward is in position 2). The resulting belief set of states is reduced to |positions| $\times$ |world beliefs|. Here on the 5 by 5 grid with a 3 block barrier, there are 22 positions and three world beliefs. Either you believe you are in the world with 1 as the goal (world 1), you believe you are in the world with 2 as the goal (world 2), or you are agnostic and have no information. For convenience, we can represent this with a single number.

We define the belief state to be 0 if $p$(being in world 1) = .5 (we do not know), 1 if $p$(being in world 1) = 1

(world 1), and -1 if $p$(being in world 1) = 0 (world 2). Thus, initially, an agent's belief will be 0 and only after having received an observation in a sign state can it transition to either 1 (you're in world 1) or -1 (you're in world 2).

# 3 Requirements

For this assignment, you are in charge of coming up with a belief transition function, belief reward function, and policy over beliefs given the two possible signs.

Fill in the code provided in the discretePOMDP file and upload the completed python file. The main function should return the two policies (one for each possible sign/world). You are given the position space, belief space, possible actions, positions where observations of the sign are, and the original reward function for each of the two worlds.

**Setup Belief Transition Class $\tau$**

Form: {(position, world belief): {action: {(next position, next world belief) : probability }}}

Position is a tuple of the xy-position in the world. Note that the belief transition is actually deterministic, so the probability will always be 1.

**Belief Reward Class $\rho$**

Form: {(position, world belief): {action: {(next position, next world belief) : scalar reward }}}.

You will be provided with the original reward functions R(s,a), one for each world. From this, you should construct the new belief reward function, rho. Note: R(s,a) is defined over s, a, but our value iteration function takes in a reward defined over s, a, next s. So when you create rho, the (next position, next world belief) tuple is any possible next belief state from (position, world belief), action in the transition dictionary.