Problem Set 1

Due Sunday, January 30th, by Midnight
Human-Robot Interaction
Spring 2022

Write legibly and do not attempt to fit your work into the smallest space possible. It is important to show all work, but basic arithmetic can be omitted. You are encouraged to use Matlab, Python, etc., but you must submit a pdf of your commented code or use the live scripts feature in Matlab. Do not submit .m or .py files.

1 MDPs

In this homework you will work with the environment shown in Figure 17.1 of Artificial Intelligence: A Modern Approach (see Figure 1). This should also be familiar from lecture!

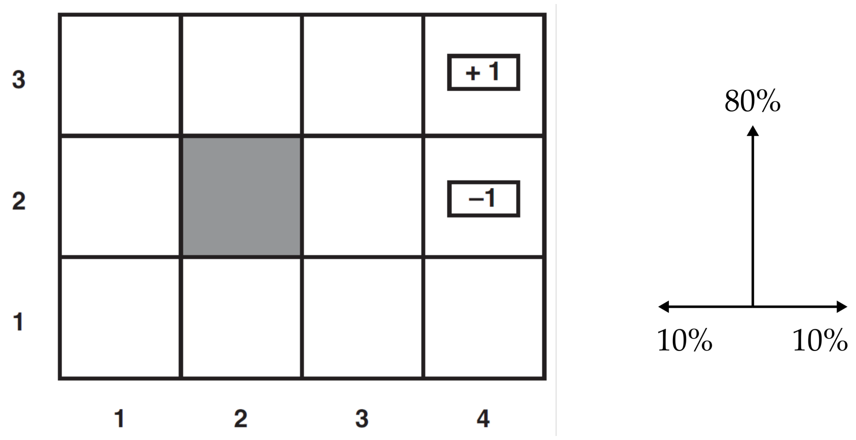

Figure 1: We have a quadrotor navigating through this grid-world environment. We cannot go into the shaded region, and once we reach either the +1 or −1 cell the interaction is over. Because of the wind our quadrotor moves stochastically: 80% of the time it does what we intend, but 20% of the time it accidentally veers to the right or left.

Your goal in this problem is to help the quadrotor make optimal decisions when navigating through the grid-world.

1.1 Value Iteration

To get started, download the `mdp.m` and `transition.m` functions from Canvas → Files → Code. On line 23 of `mdp.m` you will notice that the value iteration algorithm is currently just a `for loop`, and we have arbitrarily picked a big number (hoping that value iteration converges in that many iterations). **Write** a **termination condition** that tells us when we should stop the value iteration process. **Update** the code so that we have a `while loop` that repeats until this termination condition is satisfied.

Aside. If you do not want to use Matlab, I encourage you to use this repo:
`https://github.com/aimacode`

1.2 Optimal Policy

So far our code numerically solves for the value function $V(s)$. But ultimately we want to find the optimal policy $\pi(s)$ for our quadrotor. Given the value function, how do we recover $\pi(s)$? **Write** additional code that outputs the optimal action for the quadrotor in each state of the environment. Remember that states $[4,2]$ and $[4,3]$ are terminal states, and we cannot take any actions in those positions.

You can verify your code using Figure 17.2 from Artificial Intelligence: A Modern Approach. Set $\gamma = 1$ to reproduce these results.

1.3 Why Are These Decisions Optimal?

Set $\gamma = 0.7$ on line 13 of `mdp.m`. Run your final code for the following conditions:

- `r_empty = -0.9`
- `r_empty = +0.1`
- `r_empty = +2.0`

Draw the environment with arrows to show the quadrotor's optimal actions in each of these cases (i.e., you should have three separate drawings). **For each case, write a few sentences** to explain **why** the quadrotor is making these decisions.

2 QMDPs

Only assigned to students in ME5824 or CS5844.

Your quadrotor is uncertain about the reward assigned to the empty cells. In practice, the quadrotor might be trying to land, and is unsure whether or not the empty cells are safe.

The quadrotor has a belief over three possible rewards:

$$b(\texttt{r\_empty = -2}) = 0.1 \quad b(\texttt{r\_empty = -0.5}) = 0.7 \quad b(\texttt{r\_empty = +0.5}) = 0.2$$

Assume that — no matter what the quadrotor does — it will gain no additional information about the rewards (i.e., b is constant). What is the quadrotor's optimal policy?

Set $\gamma = 1$ when solving this problem. **Hint.** Start by finding $V(s)$ for each possible reward. Then see if you can leverage the following equation where R is short for reward:

$$\pi(s) = \arg\max_{a \in \mathbb{A}} \sum_{R} b(R) \cdot Q(s, a, R)$$

Submit your code and **draw** the robot's QMDP policy.