Homework 0

Due: September 8, 2025 at 11:59pm

Instructions

- See https://rpmml.github.io/homework_instructions/ for instructions on accessing the accompanying code and submitting homework via GradeScope.
- See https://rpmml.github.io/policies for additional policies regarding collaboration, LLMs, and late submissions.
- Use Piazza or attend office hours to ask questions about the homework.

Part 1: Written Exercises

For the first two questions in this homework, consider the following MDP:

- The state space is $\mathcal{S} = \{\texttt{hot}, \texttt{neutral}, \texttt{cold}\}$.
- The action space is $\mathcal{A} = \{\texttt{heat}, \texttt{cool}\}$.
- The reward function is

$$R(s, a, s') = \begin{cases} -1 & s = \texttt{hot} \\ 0 & s = \texttt{neutral} \\ -2 & s = \texttt{cold} \end{cases}$$

- The transition distribution is

s	a	$P(\texttt{hot})$	$P(\texttt{neutral})$	$P(\texttt{cold})$
hot	cool	0.2	0.7	0.1
hot	heat	0.9	0.1	0.0
neutral	cool	0.1	0.3	0.6
neutral	heat	0.6	0.3	0.1
cold	cool	0.0	0.2	0.8
cold	heat	0.2	0.6	0.2

- The horizon is infinite.
- The temporal discount factor is $\gamma = 0.9$.

Practice with Policy Evaluation (15 points)

Consider the policy

$$\pi(s) = \begin{cases} \texttt{cool} & s = \texttt{hot} \\ \texttt{heat} & s = \texttt{neutral} \\ \texttt{heat} & s = \texttt{cold} \end{cases}$$

We wish to compute the value function for this policy $V^\pi : \mathcal{S} \to \mathbb{R}$. Write a system of linear equations for V^π. You should simplify the expressions, but you do not need to solve the system.

Your equations should only contain the three variables $V^\pi(\texttt{hot})$, $V^\pi(\texttt{neutral})$, and $V^\pi(\texttt{cold})$, which you should abbreviate h, n, and c respectively.

Pause to Ponder (No Points & No Submission Required)

Is it possible to construct an MDP and a policy that leads to a non-invertible linear system?

Practice with Value Iteration (15 points)

Show the first two iterations of value iteration by filling in the table below. You can check your work with code, but you should do the computation by hand first to internalize the algorithm. Show your work to receive partial credit.

iteration	$V^*(\texttt{hot})$	$V^*(\texttt{neutral})$	$V^*(\texttt{cold})$
0	0.0	0.0	0.0
1			
2			

Pause to Ponder (No Points & No Submission Required)

How does the initialization (all zeros in the example above) impact value iteration?

Stretching the Definitions (10 points)

Consider the following dialogue.

Alex: *"I read in a book that an MDP has a reward function $R(s)$. But I have invented a new model called AlexMDP, which is the same as an MDP, except that its reward function $R(s, a, s')$ is a function of the current state, action, and next state. I will develop new algorithms for AlexMDP that will make me rich and famous."*

Blake: *"I hate to burst your bubble, but AlexMDPs are basically the same as MDPs. In fact, I*

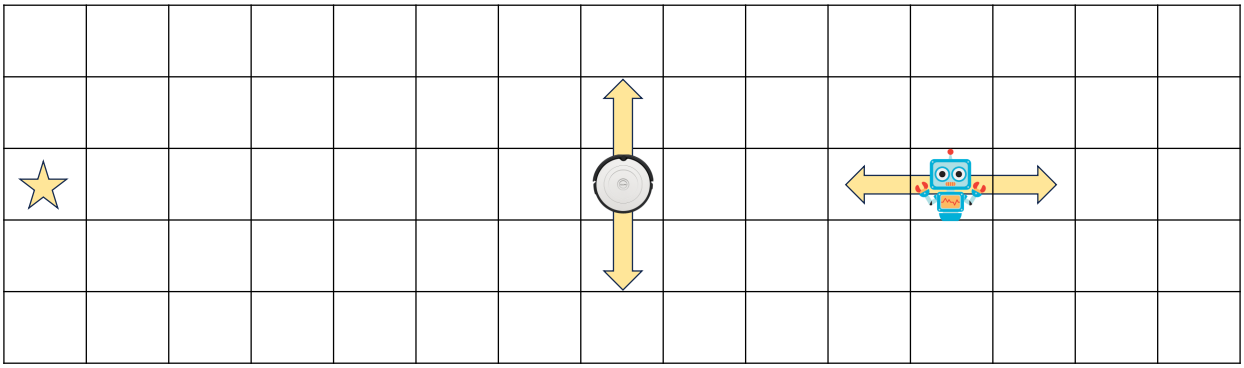

Figure 1: Illustration of the home robot design problem.

learned the $R(s, a, s')$ definition in my class."

Alex: *"No way! Prove they are the same."*

Blake: *"Okay. I will write a program that efficiently converts any AlexMDP into an MDP. Then, I will run value iteration on that MDP to get an optimal policy. Finally, I will write another program that efficiently converts the MDP optimal policy into an optimal policy for the original AlexMDP."*

Alex: *"I don't believe that's possible. Also, you can't go the other way."*

Who will eventually win this argument? Explain your answer in 3-5 sentences.

Part 2: Coding Exercises

Coding Warm-Up (5 points)

Complete the function `treats_are_sufficient` in `warmup.py`. Examine the unit tests in `test_warmup.py` to make sure you understand the question. No written answer required.

Is Uncertainty Necessary? (20 points)

In this exercise, you will explore the extent to which it is necessary to reason about uncertainty during MDP planning. Your job is to implement `CustomMDP` in `uncertainty.py` so that when the transition distribution is *determinized* (see `DeterminizedCustomMDP`), running value iteration leads to a grave mistake. See `test_determinized_worse_than_custom_mdp`. To keep things interesting, the rewards in your MDP must be bounded between -1 and 1. No written answer required.

Practice with Problem Formulation (30 points)

You have been hired as a consultant for a company that is developing a home robot. The company is considering a home that already has a robot vacuum that could get in the way of their new robot. This is a very difficult problem, but fortunately, the home and robot are rather simple (Figure 1):

1. The home is just a 2D grid with 5 rows and 15 columns.

2. The vacuum only moves up and down in the middle column of the grid. Specifically, every minute, it moves up one cell with probability 0.25, down with probability 0.25, and stays in place with probability 0.5, except at the edges, where it stays in place with probability 0.75.
3. The new home robot only moves left and right in the middle row of the grid. Its movement is deterministic—it can choose to move one cell left, or one right. It cannot stay still.
4. The robot's job is done when it reaches the leftmost column.
5. It costs \$0.1 to operate the robot for one minute. It costs \$1000 if the robot runs into the vacuum and breaks the robot permanently.
6. There is a 1% chance that the user will permanently turn off the robot after every minute. This incurs no cost.

Your job is to design an optimal policy for the robot. However, the robot has limited sensors. It cannot detect its exact position in the grid. Instead, it can only sense:

- The Manhattan distance between the home robot and the vacuum.
- Whether the home robot's current column is greater than, less than, or equal to the vacuum's (middle) column.

Design an optimal (in terms of \$ cost) policy that uses only these features, or show that this is impossible. You should make your case either way with code, which you should submit, along with a written explanation (3-5 sentences). You can use any code provided in the `rpmml` library or any of its dependencies (see `pyproject.toml`).

⚠ Important Note

Submit your answer to this question in the **Written** assignment on Gradescope. Provide a link to your code that can be accessed by course instructors (e.g, use Google Drive).

Part 3: Feedback

General Course Feedback (5 points)

How can we improve the course in future years? Any and all feedback is welcome. Please comment on this problem set, lectures, and anything else. Some feedback is required for full credit. If you prefer to additionally submit anonymous feedback, please do so through the course website.