

Markov Decision Processes

Numbers

The assignment is worth 15% of your final grade.

Read everything below carefully, this assignment has changed from previously posted material.

Why?

In some sense, we have spent the semester thinking about machine learning techniques for various forms of function approximation. It's now time to think about using what we've learned in order to allow an agent of some kind to act in the world more directly. This assignment asks you to consider the application of some of the techniques we've learned from reinforcement learning to make decisions.

The same ground rules apply for programming languages as with the previous assignments.

The Problems Given to You

You are being asked to explore Markov Decision Processes (MDPs):

1. Come up with two interesting MDPs. Explain why they are interesting. They don't need to be overly complicated or directly grounded in a real situation, but it will be worthwhile if your MDPs are inspired by some process you are interested in or are familiar with. It's ok to keep it somewhat simple. For the purposes of this assignment, though, make sure one MDP has a "small" number of states, and the other MDP has a "large" number of states. The judgement and rationalization of what is "small" and "large" will be up to you. For initial intuition, 200 states is not considered "large". Additionally, no more than one of the MDPs you choose should be a grid world problem.
2. Solve each MDP using value iteration as well as policy iteration. How many iterations does it take to converge? Which one converges faster? Why? How did you choose to define convergence? Do they converge to the same answer? How did the number of states affect things, if at all?
3. Now pick your favorite reinforcement learning algorithm and use it to solve the two MDPs. How does it perform, especially in comparison to the cases above where you knew the model, rewards, and so on? What exploration strategies did you choose? Did some work better than others?

Coding Resources

The algorithms used in this assignment are relatively easy to implement. Existing implementations are easy to find too, below are java and python examples.

[Brown-UMBC Reinforcement Learning and Planning \(BURLAP\) java code library](#)[Links to an external site.](#) (java)

[bettermdptools](#)[Links to an external site.](#) (python)

What to Turn In

You must submit:

- a file named *README.txt* that contains instructions for running your code, including a link to your code.
- a file named *yourgtaccount-analysis.pdf* that contains your writeup.

The file *yourgtaccount-analysis.pdf* should contain:

- A description of your MDPs and why they are interesting.
- A discussion of your experiments.

Note: The report is limited to 8 pages total. We will not grade anything past page 8. This includes any title pages (I do not recommend anyone use a title page in this class), figures, and references. This is different from the first two assignments. This is standard IEEE conference length and good practice in conciseness and brevity.

Grading Criteria

You are being graded on your analysis more than anything else. I will refer you to the grading sections from Assignment #1 for a more detailed explanation. Additional discretionary points will be given to analysis and discussion that goes above and beyond the minimum requirements for analyzing your favorite reinforcement learning problem. This is your chance to show off what you have learned this term. The maximum score of this report is capped at 100%. As always, start now and have fun!