CS7641 Assignment 4
Markov Decision Processes

1 Assignment Weight

The assignment is worth 20% of the total points.

Read everything below carefully as this assignment has changed term-over-term.

2 Objective

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 and libraries. You may program in any language that you wish insofar as you feel the need to program. As always, *it is your responsibility* to make sure that we can actually recreate your narrative, if necessary.

Please note, this class implements changes to the assignments term-over-term as we are calibrating the course incrementally. Please read through everything, even if you are submitting work from a previous semester as the requirements will likely have changed.

3 Procedure

3.1 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?

Analysis writeup is limited to 8 pages. This is different from the first set of assignment. The page limit does include your citations. Anything past 8 pages will not be read. Good rule of thumb, double check your final submission after submitting to Canvas.

Please keep your analysis as short as possible while still covering the requirements of the assignment.

3.2 Acceptable Libraries

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

- bettermdptools (python) `https://github.com/jlm429/bettermdptools`

- BURLAP (java) `http://burlap.cs.brown.edu/`

4 Submission Details

You must submit:

- A file named README.txt containing instructions for running your code (see note below)

- A file named yourgtaccount-analysis.pdf containing your writeup (GT account is what you log in with, not your all-digits ID)

The file yourgtaccount-analysis.pdf should contain:

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

- A discussion of your experiments.

Note: we need to be able to get to your code and your data. Providing entire libraries isn't necessary when a URL would suffice; however, you should at least provide any files you found necessary to change and enough support and explanation so we can reproduce your results on a standard linux machine.

Discretionary Points: In an effort to encourage exploration and utilize theory and practice from the course material, we will award an additional 5 points of extra credit to this assignment to those who go above and beyond the requirements. This does not mean provide every single plot generated, but rather a different piece of analysis or utilization of a novel method from a recent paper to compliment what is asked in any particular section of the report. The discretionary points are optional as you can do extremely well on the assignment without it.

5 Rescoring Criteria

When your assignment is scored, you will receive feedback explaining your errors and successes in some level of detail. This feedback is for your benefit, both on this assignment and for future assignments. It is considered a part of your learning goal to internalize this feedback.

If you are convinced that your score is in error in light of the feedback, you may request a rescore within a week of the score and feedback being returned to you. A rescore request is only valid if it includes an explanation of where the grader made an error.

It is important to note that because we consider your ability to internalize feedback a learning goal, we also assess it. This ability is considered 10 percent of each assignment. We default to assigning you full credit. If you request a rescore and do not receive at least 5 points as a result of the request, you will lose those 10 points.

6 Plagiarism and Proper Citation

The easiest way to fail this class is to plagiarize. **Using the analysis, code or graphs of others in this class is considered plagiarism**. The assignments are designed to force you to immerse yourself in the empirical and engineering side of ML that one must master to be a viable practitioner and researcher. It is important that you understand why your algorithms work and how they are affected by your choices in data and hyperparameters. The phrase "as long as you participate in this journey of exploring, tuning, and analyzing" is key. We take this very seriously and you should too.

What is plagiarism?

If you copy any amount of text from other students, websites, or any other source without proper attribution, that is plagiarism. The most common form of plagiarism is copying definitions or explanations from wikipedia or similar websites. We use an anti-cheat tool to find out which parts of the assignments are your own and there is a near 100 percent chance we will find out if you copy or paraphrase text or plots from online articles, assignments of other students (even across sections and previous courses), or website repositories.

What does it mean to be original?

In this course, we care very much about your analysis. It must be original. Original here means two things: 1) the text of the written report must be your own and 2) the exploration that leads to your analysis must be your own. Plagiarism typically refers to the former explicitly, but in this case it also refers to the latter explicitly.

It is well known that for this course we do not care about code. We are not interested in your working out the edge cases in k-nn, or proving your skills with python. While there is some value in implementing algorithms yourselves in general, here we are interested in your grokking the practice of ML itself. That practice is about the interaction of algorithms with data. As such, the vast majority of what you're going to learn in order to master the empirical practice of ML flows from doing your own analysis of the data, hyper parameters, and so on; hence, you are allowed to steal ML code from libraries but are not allowed to steal code written explicitly for this course, particularly those parts of code that automate exploration. You will be tempted to just run said code that has already been overfit to the specific datasets used by that code and will therefore learn very little.

How to cite:

If you are referring to information you got from a third-party source or paraphrasing another author, you need to cite them right where you do so and provide a reference at the end of the document [Col]. Furthermore, "if you use an author's specific word or words, you must place those words within quotation marks and you must credit the source." [Wis]. It is good style to use quotations sparingly. Obviously, you cannot quote other people's assignment and assume that is acceptable. Speaking of acceptable, citing is not a get-out-of-jail-free card. You cannot copy text willy nilly, but cite it all and then claim it's not plagiarism just because you cited it. Too many quotes of more than, say, two sentences will be considered plagiarism and a terminal lack of academic originality.

Your README file will include pointers to any code and libraries you used.

If we catch you. . .

We report all suspected cases of plagiarism to the Office of Student Integrity. Students who are under investigation are not allowed to drop from the course in question, and the consequences can be severe, ranging from a lowered grade to expulsion from the program.

7   Version Control

- 01/22/2024 - TJL copied assignment to JM's LaTeX template.

- 03/01/2024 - TJL updated class percent total, phrasing, discretionary points.

References

[Col]   Williams College. *Citing Your Sources: Citing Basics*. URL: https://libguides.williams.edu/citing.

[Wis]   University of Wisconsin - Madison. *Quoting and Paraphrasing*. URL: https://writing.wisc.edu/handbook/assignments/quotingsources.

Original assignment by Charles Isbell. Updated for Spring 2024 by John Mansfield and Theodore LaGrow. Modified for LaTeX by John Mansfield.