Machine Learning - CS 7641 Assignment 4

Anthony Menninger
Georgia Tech OMSCS Program
amenninger3
tmenninger@gatech.edu

Abstract

This paper explores Markov Decision Processes through the use of Value Iteration, Policy Iteration and Q Learning. It looks at a simple "Forest" MDP and a larger Frozen Lake "Grid World".

Problem Introduction

A key element in both Markov Decision Processes (MDPs) is the idea of a discount, which is often referred to as ***Gamma (γ)***. This discount factor, between 1 and 0, allows solutions for infinite MDPs by discounting future value by ***Gamma*** for each step into the future. In an infinite activity, this ends up valuing future value at $\frac{1}{1-\gamma}$. Gamma close to 0 creates a very close horizon where only immediate rewards are maximized, while Gamma close to 1 creates a long horizon, where maximizing long term reward is emphasized. As will be seen, MDPs are very sensitive to this and small changes can create very different solutions.

The key toolset I used was the hiive fork [3] of the MDP Toolbox for Python [2], which is derived from the MDP Toolbox [1]. I also used the OpenAI Gym Frozen Lake environment [4] for the larger Frozen Lake MDP.

Forest MDP

Forest MDP is a simple MDP from the hiive MDP Toolbox [2] that models the value of a forest with respect to two actions that can be performed each year: ***wait*** or ***cut***. There is a stochastic element of forest fires that occur with probability ***p***. Each year is a state, with a max of ***S*** years / states. Cutting deterministically transitions to the initial state, ***state=0*** and provides a ***cutting reward***. Waiting transitions to the next year, or if in the max state, remains there. Forest fires provide a stochastic element, with a fire transitioning back to the initial state. A ***waiting reward*** only occurs in the max state, ***state=S***. This is a continual MDP, with no terminal or absorbing states.

The base setup for ***Forest*** is seven years (***S=7***), a 10% chance of forest fire (***p=0.1***), a cutting reward of 2 (***cutting reward=2***) and a waiting reward of 4 (***waiting reward = 4***).

In the context of MDP's, each state has an action that maximizes expected value and the set of maximizing actions for all states is called a ***policy***. This policy can be examined to determine what rational actors are likely going to do.

There are several very interesting aspects to this problem. ***Forest MDP*** models a key choice being made today around ecology and the environment. What are the rewards needed to keep forests around? How do maximizing actions (***policy***) change as the chance of forest fires increase? How does the length of the considered horizon (***Gamma discount***) influence expected outcomes.

This also highlights a key challenge in MDP's and Reinforcement Learning, which is understanding rewards and setting them appropriately. ***Forest MDP*** allows modeling of different situations to inform the construction of public policy for governments, NGO's and corporations. The ***cutting reward*** might be clearly modeled using expected timber value, but other types of reward may want to be considered in the ***waiting reward***, such as carbon reduction and maintenance of ecological diversity. This MDP can be used to model economic rewards and penalties to compel desired outcomes, such as a carbon tax that would reduce the value of the ***cutting reward***.

As described, this MDP is more likely to be "solved" using Value Iteration (VI) or Policy Iteration (PI) in order to use it for comparing different tradeoffs and outcomes.

Lake MDP

Lake MDP is a larger MDP that uses the OpenAI gym Frozen Lake [4,5] to create its environment. It is a 2D grid with four actions to move ***up, down, left, right*** and can be of ***Size***, which is the number of grids on one size of the square, creating ***Size x Size*** states. The lake can be set to not slippery, where the action is deterministic, or slippery, where the action is stochastic. When slippery, the action has a 1/3 chance of moving in the desired direction, a 1/3 to the left of the desired direction and 1/3 to the right of the desired direction, which is fairly stochastic. This is an episodic MDP where the player starts in the first state - (0,0) upper left corner. The goal state (size, size) is in the bottom right corner and has a reward of 1. There are also random holes which are absorbing states with a reward of 0.

I created Small (***Size=4***), Medium (***Size=8***), and Large (***Size=16***) worlds to explore the impacts of different size worlds.

This MDP is interesting because it embodies a core Reinforcement Learning challenge of exploring an environment and learning what to do. It describes a common, everyday

task of exploring an environment and learning where the rewards and traps are. Robots are often in a situation where they need to move about and find out what they should be doing. In addition, it creates a much more interesting set of transitions between the different states, as compared with the ***Forest MDP***, where transitions are either next state or back to beginning. Also, ***Lake MDP*** is episodic vs continual for ***Forest MDP***.

As described, this MDP is more likely to be ”learned” in a setting using Q Learning to explore how agents work.

Forest and Lake MDPs Value Iteration and Policy Iteration

Lake MDP Value Iteration and Policy Iteration

Forest MDP Q Learning

Lake MDP Q Learning

Summary

References

1 Markov Decision Processes (MDP) Toolbox, https://miat.inrae.fr/MDPtoolbox/ Accessed: 11/1/2022.

2 Markov Decision Process (MDP) Toolbox for Python, https://github.com/sawcordwell/pymdptoolbox: Accessed: 11/1/2022.

3 hiive Fork: Markov Decision Process (MDP) Toolbox for Python, https://github.com/hiive/hiivemdptoolbox: Accessed: 11/1/2022.

4 Brockman, G. et al., 2016. Openai gym.

5 Openai gym Frozen Lake Documentation, https://www.gymlibrary.dev/environments/toy_text/frozen_lake/: Accessed: 11/1/2022.