CS4013/5013 Assignment 6

Fall 2024

Due Date: Nov 24, 2024.

In this assignment, we implement a Markov decision process algorithm which is policy iteration plus Monte Carlo. See detailed algorithm in the last page of '1104_mod7a.pdf'.

Apply your algorithm to estimate the utility of each state in the grid world in Figure 1 (left) based on the stochastic move in Figure 1 (right). This move means, for example, if the agent takes action "up", 60% chance it will end up moving up, 20% chance moving left, 10% chance moving right and 10% chance moving down.

Remember, however, if an agent hits the wall or rock, it stays in the current state. For example, if it takes action UP at (2,1), 60% chance it will stay (by trying to go up but hit the rock), 20% chance it will move to (1,1), 10% chance it will move to (3,1) and 10% chance it will stay (by trying to move down but hit the wall) – so the total chance to stay is 60% + 10% = 70%.

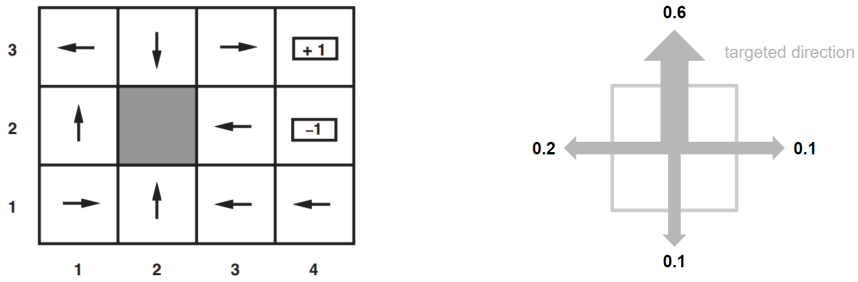

Figure 1: Grid World and Stochastic Move

Task 1. Use Monte Carlo to Estimate Utility

Implement first-visit Monte Carlo and use it to estimate the utility of each non-terminal state based on the policy in Figure 1 (left) and stochastic move in Figure 1 (right). Set discount rate $\gamma = 0.8$ and the reward at each non-terminal state $r = -0.04$. To estimate the utility of each state, run at least 10 experiments. Report your estimated utilities in the following table.

State	(1,1)	(1,2)	(1,3)	(2,1)	(2,3)	(3,1)	(3,2)	(3,3)	(4,1)
Utility									

Task 2. Optimize policy based on your estimates.

Based on your estimated utilities, optimize your policy at each state using Bellman's equation. (See details in the page 31/35 of '1104_mod7a.pdf'.) You can write a program to automatically identify the optimal policy or manually identify it. If you choose to write a program, upload the code. If you choose to manually identify, elaborate the analysis for state (1,2). In either case, report your updated policy in the following table (R: right, L: left, U: up, D: down).

State	(1,1)	(1,2)	(1,3)	(2,1)	(2,3)	(3,1)	(3,2)	(3,3)	(4,1)
Old Policy	R	U	L	U	D	L	L	R	L
New Policy									

Note 1: If you repeat Task 1 and Task 2, utility estimates will converge; so will the policy.

Note 2: Although we introduce policy iteration plus Monte Carlo as an algorithm for Markov decision process, it can also be used for passive reinforcement learning problem (less efficient than temporal difference learning). See page 5-6 of '1104_mod7b.pdf'.

Task 3. Some quiz questions will be posed on Canvas. Complete them.

Submissions Instructions

You should place the utility estimate table and policy update table in a pdf file named 'hw6.pdf' and upload it to Canvas through the submission page for hw6. You can generate the pdf file using any tool, although you are encouraged to generate it using Overleaf. A latex template 'hw6_Latex.txt' will be provided to you.

You also need to upload the code that generate the utility estimate table. Name this code as hw6_task1.py and set the state as (1,1). If you choose to write a program to update optimal policy, upload the code and name it as hw6_task2.py.

You need to implement the policy iteration plus Monte Carlo algorithm by yourself, and not to use any existing policy iteration or Monte Carlo algorithm from any Python library.