Machine, Data and Learning

Assignment 3.2
MDP

Deadline: April 18th 2024

1 Introduction

A Markov Decision Process (MDP) refers to a stochastic decision-making process that uses a mathematical framework to model the decision-making of a dynamic system.
MDPs rely on variables such as the environment, agent's actions, and rewards to decide the system's next optimal action.
Given below is the value iteration algorithm, which uses the Bellman equation.
The Bellman equation updates the value function based on the maximum expected value of all possible actions A from state I.

The algorithm works as follows,

1. Initialize utility of all non-sink states to 0, i.e. $U_0(I) = 0$

2. Utility values for states are calculated as follows

$$U_{t+1}(I) = \max_A \left[C(I,A) + \gamma \sum_J P(J|I,A) U_t(J) \right] \tag{1}$$

   where A is the action chosen, $C(I,A)$ is the cost of taking action A from state I, γ is the discount factor, $P(J|I,A)$ is the probability of reaching state J given the agent was at state I and chosen action A

3. Repeat this algorithm until $U_{t+1}(I)$ is close to $U_t(I)$ for all states.

4. The optimal policy is defined as follows

$$\text{Policy}(I) = \text{argmax}_A \left[C(I,A) + \gamma \sum_J P(J|I,A) U_t(J) \right] \tag{2}$$

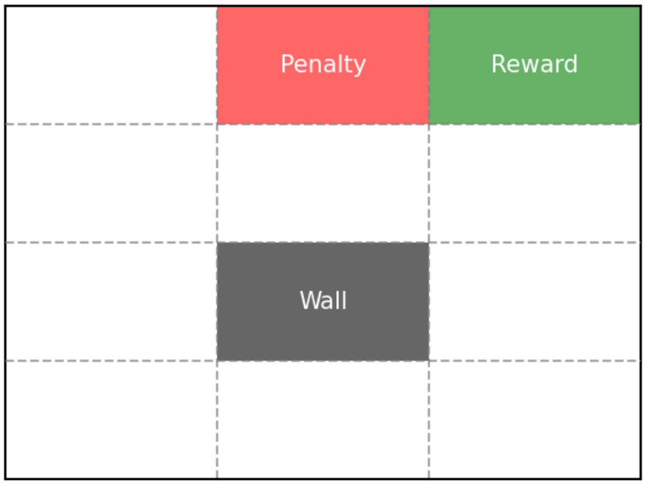

Figure 1: Grid world

2 Question 1 (70 marks)

2.1 Values Required

You will need the following values for the implementation:

1. Reward for reaching the goal state = 1
2. Penalty for reaching the red state = -1
3. Step cost = -0.04
4. Probability of going in the direction of the action = p
5. Probability of going in a direction perpendicular to the action = $\frac{(1-p)}{2}$
6. Discount factor = 0.95

   One thing to note is that if the agent is unable to move according to the action i.e., the move is blocked by the boundaries or the wall, the agent will remain in the same state.

2.2 Task A (20 marks)

Your task is to implement Value Iteration in Python for the given grid world (Fig 1). Take $p = 0.7$ for this task. The code should print the utility value of each cell in the grid after each iteration until the values converge, where convergence is defined by a difference ≤ 0.0001 between utility values.

2.3 Task B (50 marks)

Run the algorithm for p values ranging from 0.1 to 0.9 with steps of 0.1. Print only the final policies that the algorithm converges to in the following format for each p.

```
'right'  'none'  'none'
 'up'     'up'   'down'
 'up'    'none'  'down'
 'up'    'left'  'left'
```

Observe how the policy changes with different p values. Comment on why the changes occur as they do.

3 Question 2 (30 marks)

Implement Task A by hand for two iterations. You need to show the calculation for utility values at each point on the grid. One iteration means one time step. See if your values match the output from your code.

4 Submission Format

Submit a zipped file **roll_no.zip** It should contain a **roll_no.ipynb** containing the implementation along with the results in the format mentioned above and a **roll_no.pdf** containing the handwritten submission.

4.1 General instructions

The deadline for this assignment is April 18, 2024 at 11:59 PM
Please start the assignment early to prevent a clash with endsem preparation
Plagiarism is strictly prohibited, and a straight 0 will be awarded in such cases

Cheers,
MDL TAs