Assignment II - Planning Decision Process

Introduction

This assignment involves finding an optimal policy for a given implementation will focus on iterative techniques such as valu their respective variants. Additionally, the analysis will extend various components defining an MDP on the optimal policy.

Domain Description

Consider an autonomous car in the grid world environment. T allowing it to run on petrol or electricity. It is capable of movir right, and left. Due to construction work, holes have been du Therefore, the autonomous car needs to find an optimal polic every grid cell) to avoid falling into the holes and drive safely

Environment

The environment is represented as a 2D grid world with discr designated as 'H,' indicating the presence of a hole. The gara vehicle is initially parked, and the destination is marked as 'G goal are labelled as 'F,' indicating free cells.

Transition Model

- The car can move in four directions: top ('t'), down ('d'), r state change for each action is:-

$$right(x, y) = (x + 1, y)$$
$$left(x, y) = (x - 1, y)$$
$$top(x, y) = (x, y + 1)$$
$$down(x, y) = (x, y - 1)$$

- Because of rain, the roads in the grid world become slipp effects of actions. In other words, the transitions are stoc intended cell with a probability p and randomly in one of t probability $(1-p)$. Transitions outside the grid boundar transitions resulting in a position outside the grid bounda
- It is important to note that moving into a hole or reaching *state* in this context.

Reward Model

- The car gets a reward after taking an action and reaching
- For each action that does not lead to a hole or the goal, tl `living_reward`. In terminal states, where the agent enco receives the respective rewards, namely `hole_reward` an reward model $R(s, a, s')$ is defined as

$$R(s, a, s') = \begin{cases} \texttt{living\_reward}, & \\ \texttt{hole\_reward}, & \text{if} \\ \texttt{goal\_reward}, & \text{if} \end{cases}$$

- The discount factor, γ, represents the autonomous car's accumulating delayed rewards.

Your Implementation

Part A: Solving for optimal policy

Your task is to use iterative methods to solve the given MDP a implementation should be general enough to be able to work with two maps: `small_map` and `large_map` to test your impl default values for the parameters of the transition and rewarc

- The transition probability p is 0.8, meaning that 80% of t state, while there's a 20% chance of reaching a random r
- The `living_reward` and `hole_reward` are 0. The `goal_`

- The discount factor, γ, is 0.9.

Value Iteration

Implement a vanilla value iteration. As discussed in the le
distance between consequent value updates to determin
different epsilon. Experiment with varying epsilon values
final value estimate yields a sensible optimal policy.

Asynchronous Updates

Implement value iteration with asynchronous updates, wh
in-place within each iteration. Note that the order in whic
influences the convergence rate in the case of asynchron
following sweeping strategies.

- **Row-major sweep**: Cells are updated row by row,
  to the next.
- **Prioritized sweep:** Use the magnitude of Bellmar
  is, the states whose successors change the most
  defined as

$$\left| \max_a \mathbb{E}[R_{t+1} + \gamma v(S_{t+1}) | S_t$$

Compare and report the convergence speed, measured i
synchronous and asynchronous versions of value iteratio
becomes notable for larger grids.

Policy Iteration

Implement policy iteration, assuming a random initial poli
used to determine the convergence of the policy iteration

For all the algorithms, report the following information for

1. Report the wall time taken for convergence of both th iterations (count of policy iteration cycles for policy ite value iteration) till convergence in the default settings scalable for larger environments?
2. Plot the estimated value of the starting state for every plots, one for each map. Each plot should contain four - vanilla value iteration, row-major sweep, prioritized s evolution of the value of the starting state.
3. Generate a heat map of the values obtained after the policy as arrows (denoting the direction of movement) results in a better optimal policy?

Part B: Analysis

In the subsequent parts, you will be tasked to change the valu environment and then observe, analyze and report the resulti and value estimates. Unless otherwise specified, assume tha iteration implementation to derive the optimal policy. Also, re `small_map.`

B1. Living reward analysis

For this part, assume that the car uses your vanilla value itera optimal policy. Also, assume that the transition function is de

- The petrol prices are fluctuating due to demand in the cru expense will increase if it uses petrol to power its engine the living reward). Indicate how your policy will change as -0.9. In both cases observe and explain the differences b that of the default settings.
- To decrease the inflation in petrol prices, the grid world g use electric vehicles. So, using the electric engine will pro +0.001. Assume a discount factor (gamma) of 0.999. Ru the new setting and report your observations along with a

- Plot the heat maps of the values after convergence and t
  above settings. Analyse the policies obtained and report

B2. Changing transition probabilities

- The rain, coupled with snowfall, makes the road exceptio
  navigate. Consequently, every action has an equal probab
  intended direction or in a direction perpendicular to the ir
  is moving right, it may equally likely end up moving right,
  of 1/3.
- Plot the heat map and the policy and highlight the differe
  from that of default settings.

Implementation Guidelines

Maps and Visualizer

- Please download the package from Moodle. The package

```
A2 ├── maps # The layouts of different environm
│   ├── large_map.csv ├── plot.py
```

- **Maps:** There are two maps. `small_map` is a grid of size 4
  size 50×50. Each line in the CSV file corresponds to the
  are separated by commas, and each entry denotes the ty
  (start), 'G' (goal), 'H' (hole), or 'F' (free).
- **Visualizer:** We have provided you with a `plot` function (
  simulate your policy. However, this is only for visualization
  optional. You are also free to make your visualization pip

```
def plot(frames, frame_delay = 500, cell_size = 50
numpy arrays where each array is a state of the gr
between consecutive frame in milliseconds while re
each cell in grid world ''' # color coding:- # cya
green: destination/goal, yellow: hole
```

Evaluation

You will be evaluated based on the correctness of the implem
concepts and the validity of your analysis/findings. A viva will

Submission Guidelines

- This assignment is to be done **individually**.
- Please submit it on Moodle as a single zip file named **<A2**
  should contain a single python file - **A2.py -** containing y
  report.pdf - that details your algorithm, results and the i
  plots to the drive (if required), and include the drive link a
  the report.

```
├── A2.py ├── report.pdf
```

- **The submission date is 6:00 pm on Wednesday, Februa**
  incur a 10% penalty up to a single day.
- The assignment must be done only from your own origina
  previous implementations done by others. Do not violate
  policy on honour code violations discussed in class).