CS747:Programming Assignment 2 Report

Debabrata Biswal
Roll no : 203050024

October 23, 2020

1 Introduction

The report contains the implementation details of the algorithms that I have coded for this assignment including the assumptions,some observations. P.S: All notations are according to the class conventions.So I am not defining them separately.

2 Task 1

Implementation:

The implementation of all algorithms were straightforward.I just had to convert the mathematical expression in the class slides into code.Some thinking was needed for vectorization of some tasks.

For **VI** I kept a tolerance of 1e-12 , and initialized all state value functions to zero.

Similiarly for **HPI** I initialized the with random policies and did policy improvement.I set the tolerance to 1e-12 for this task also.

For **LP** I used pulp and the default CBC solver that comes with it to solve the Linear Programming Problem.

3 Task 2

This task required some thought.The maze is encoded to MDP in the following way.

- The number of states of MDP are the number of valid cells in the maze.(Cells that are not 1's).
- The number of actions are 4.(0,1,2,3 corresponding to N,S,E,W)
- All actions are deterministic.
- Going from a valid cell (i, j) to valid cell $(i \pm 1, j \pm 1)$ corresponds to transition from $st(i, j)$ to $st(i \pm 1, j \pm 1)$ with probability 1 and reward -1.(Here st is mapping from cell i, j to state-space.)
- Going from valid cell to some invalid cell (cells having walls or cells outside maze) incurs a reward of -10 and the state does not change.

- the MDP type is episodic.
- Discount factor is set to 1.

  Intuitively ,setting reward -1 for each move is to ensure the shortest path is found and it does not get trapped in a loop to increase the reward.(Which would have been the case if the reward was some positive number).

  As for which algorithm to use , I tried with all 3 algorithm and among them I found VI to be the fastest for this task.And also HPI was having hard time with gamma = 1 since the matrix sometimes became singular.And although LP was giving correct answer it was slow.

  Algorithm to test TASK 2 : **VI**

  Assumption : The maze has only one end state.Although the code could be easily modified (only encoder.py and decoder.py since planner.py already deals with multiple endstates) to include multiple end states.In that case the algorithm would choose the end-state with minimum distance from start state.