CS 747: Programming Assignment 2

Dipesh Tamboli - 170070023

October 23, 2020

1 Task 1 - MDP Planning Algorithm

- I have used initiated all the matrices(π, V, T, R) as numpy zero array for all the cases.
- For all sorts of tie-breaking, I have preferred value with the minimum argument(that's what **np.argmax** returns).
- I am calculating π^* from V^*.

Some specific points corresponding to my implement:

1.1 LP

- I an using list for storing the variables and not dict.

1.2 HPI

- I have used value iteration for getting V after choosing every next policy
- Also, I am specifically using argmax over Q for choosing next set of actions(π), that is, choosing the action which has maximum Q(s,a).
- Taking a lot of time compared to LP and VI for the episodic MDP where discount is 1.

2 Task 2 - Solving a maze using MDPs

I have considered following points while encoding maze as a MDP.

- I didn't encode the boundary walls as any states as it will just increase the number of states without really helping in solving the MDP.
- I have used following reward scheme:
  - "-1" for going to an empty state from an empty state

  - ”-10” if it is going to bump any wall and next state will be the same as the current state(as it will help agent to get out of the pitfalls where bumping to the wall maybe an equi-probable option to get less reward rather than exploring the grid and thus higher penalty for that action)
  - ”10” for reaching to the final end state

- Also, I haven’t define any reward nor any probabilities for the occuoied states(wall) and eventually it is set to zero as I have initialize T and V from numpy zeros array.