Assignment 2: Report

CS747

Siddharth Saha
170100025

November 20, 2020

1 MDP Planning Algorithms

Design decisions:

- We store the Rewards and Transition probabilities in a 3D numpy array: S*A*S.
- All elements of both arrays are initialized as 0.
- **Value Iteration Solver:**
  - The update of action values is vectorized to speed up implementation: `np.sum (np.multiply (T, R + gamma*V), axis=2)`.
  - The threshold `eps` is defined in order to break when norm of difference between consecutive value functions is negligible.
- **Linear Programming Solver:**
  - This method uses the PuLP solver.
- **Howard Policy Iteration Solver:**
  - The `policyEvaluation` function evaluates value function using the PuLP solver.
  - It chooses `best_action` such that maximum states are improved and returns when there are no more improvable states.

2 Solving a maze using MDPs

Kindly run MazeVerifyOutput using value iteration solver.
Formulation of maze as MDP:

- 4 actions [N, E, W, S] encoded as [0,1,2,3] respectively.
- Each free grid square considered to be a potential state of the agent.
- Transition probability between any adjacent free grid squares is 1.

- We want to minimize total moves, thus reward of -1 for all non-terminal transitions.
- Reward of 0 for transition leading to terminal state.
- Reward of -2 for action that lead into a wall and effectively do not change the state.
- Gamma of 1 as all rewards are equally important and episodic task as it is guaranteed to end.