

Programming Assignment 2 : CS 747

Bhaskar Gharu, 180050023

September 2020

Implementation details and assumptions for Task 1 (T1)

- Parsed the mdpFile and created 3 dimensional numpy arrays of shape (numStates, numActions, numStates) for storing transition rewards and probabilities
- Transitions which are not present in the mdpFile are assumed to occur with 0 reward and 0 probability
- The threshold used to test for the convergence of **Value Iteration** is **1e-12**, i.e. if the sum of absolute difference of value for each state is less than equal to threshold then the algorithm is stopped
- Implemented **Linear programming** algorithm for solving bellman equations using **Pulp** in python
- For **Howard's Policy iteration** initialised the initial policy as 0 for all states, for each iteration solved bellman equations using numpy's linear algebra module to get the value function for given policy and improved upon the current policy for each of the improvable states

Implementation details and assumptions for Task 2 (T2)

- Encoded the maze such that all the cells for a given grid except the cells with value 1 are valid states
- Assumed that there is only one endstate (3) in the grid, and the grid is padded by walls (1s)
- For a given state there are 4 actions - [0,1,2,3] as West, East, North and South respectively
- all transitions are encoded with reward, probability as -1, 1 respectively
- also encoded self looping transitions for states if surrounded by wall for each action
- Evaluated the correctness for the encoded design using **Value Iteration** only
- Once the optimal policy is evaluated for each state, started traversing on the grid from the start state until the end state is reached or if the value function of current state is found to be zero