CS6700 : Reinforcement Learning
Programming Assignment-3 Report
HRL and DQN

Name: Pragnesh Rana **Roll number:** ME17S301

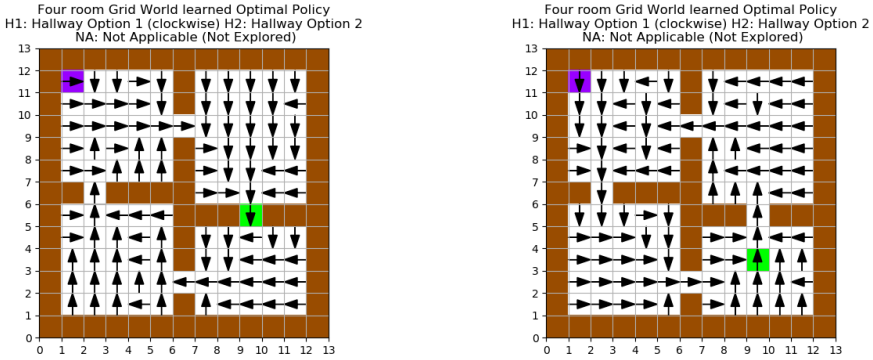
- Part-1 is on Hierarchical Reinforcement Learning
- Part-2 is on Deep Reinforcement Learning

Hierarchical Reinforcement Learning

The SMDP and Intra option Q learning has been used to solve the four room grid world problem. Let's start with brief introduction about the problem.

Answers-1: Grid World of Four Rooms and Visualization the learned Q values

The defined grid world is divided into four rooms. The upper left is room is define as Room-1. Numbering of room follows the clockwise notation. The agent is defined in upper left corner of room as given in fig-1 by blue colour cell. In image In the fig.-1, the brown colour indicates wall and green colour indicates terminal state.The study is conducted for two terminal state which is defined as G1 as in fig.-1(a) and G2 as in fig.-1(b). In the grid, each room has two hallways which can take agent from one room to another. The hall-way option follows policy π such that the agent get transferred to terminal state with shortest possible path and least possible obstacles.



(a) Grid world with goal G1 (b) Grid world with goal G2

Figure 1: Grid world of four rooms. Blue:Agent, Green:Terminal The grid world-1(a) has terminal state G1 and grid world-1(b) has terminal state G2. Arrow indicates the optimal policy. The policy in fig-1(a) is obtained using option-1 where as same in fig.-1(b) by option-2

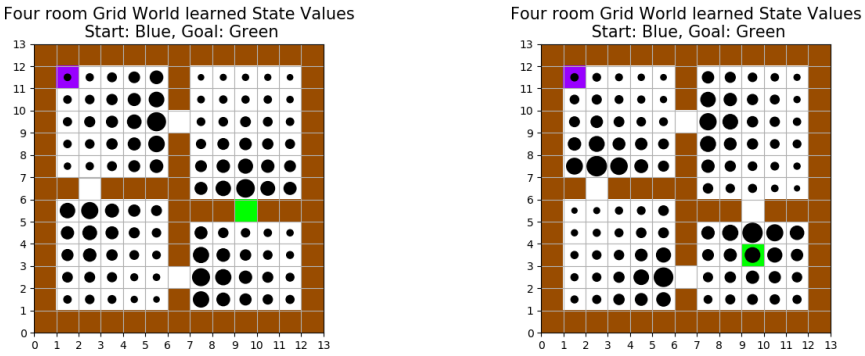
For one move, agent is rewarded with 0. For terminal state reward is +1. With Pr $= \frac{2}{3}$, the agent take correct action and other actions are performed with Pr $= \frac{1}{9}$. The vale of discounted factor is $\gamma = 0.9$. Each hallway option has termination condition 0 for states lies within room and 1 if outside. The initiation of state includes room as well as hallway. The initiation state is

defined only inside the room which makes the world deterministic. To take agent from one room to another, there are two possible options. Option-1 follows the clockwise notation which take agent from room-1 to room-2. Option-2 follows anti-clockwise direction which can take agent from room-1 to room-4.

To solve the defined problem, SMDP and Intra-option Q learning is utilized. The optimal policy is the fundamental thing for used methods. To obtain the optimal policy using Q-learning, the initial states were randomly selected and goal is directed to hallway. likewise, eight policy is obtained. The state values are obtained using the maximum return.

$$V(s) = \underset{Q(s,a)}{\operatorname{argmax}} \tag{1}$$

The obtained optimal policy for goal-G1 and G2 is given in fig.-1. The visualization of state values are given in fig.-2. For goal both goals, higher state values are obtained near hallways and terminal state. High state values are indicated by bigger size of circle.



(a) State values with goal G1 by option-1 (b) State values with goal G2 by option-2

Figure 2: The grid world-2(a) has terminal state G1 and grid world-2(b) has terminal state G2. Circle size indicates the associated state value. The values in fig-2(a) is obtained using option-1 where as same in fig.-2(b) by option-2

SMDP Q-learning :

Semi Markov Decision Processes are generalized MDPs, which allows policy maker to choose action according to change in state. It also provides the evolution of policy with continuous time while following arbitrary probability distribution. In short, SMDP follows the similar nature of MDP with options. Execution option starts with state $\mathscr{I}$ following policy π and jumps to terminating state s'. The SMDP Q-learning updates for option-value function is given by,

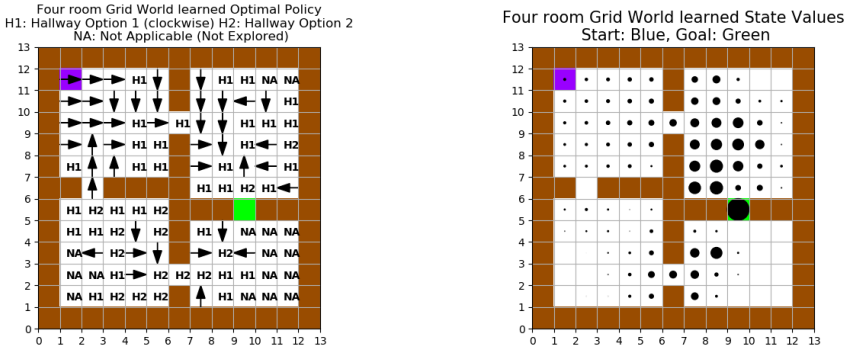
$$Q(s,o) = Q(s,o) + \alpha[r + \gamma^k \max_{o' \in s'} Q(s',o') - Q(s,o)] \tag{2}$$

where,
k - the number of time steps between s' and s
r - Cummulative discounted return over time

For Goal-G1:

The learned optimal policy by SMDP-Q learning for goal-G1 is given in fig.-3(a). **The optimal policy is resultant of defined policy and primitive actions as well.** The primitive action are better than option away from goal and near the start state. Whereas, **near the terminal state Q-values are high and options play crucial role.**

Same can be observed from the state value diagram. **Near the goal state has higher valuer, which decrease as we move away from the goal.** Due to property of learning from experience and bound of wall near terminal state, in near the goal region there might be high chances of stumbling. Some of the states are not explored as it possible to reach toward goal by following any option from those states.

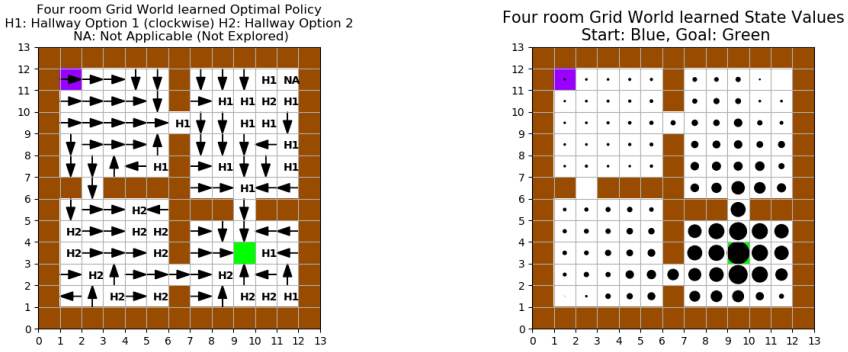


(a) Optimal policy by SMDP-Q (b) State values for SMDP-Q

Figure 3: Grid world of four rooms. Blue:Agent, Green:Terminal. The optimal policy-3(a) and associated state values-3(b) for Goal-G1 after training of 10000 episodes.

For Goal-G2:

The optimal policy for goal-G2 is given in fig.4(a). For this case also, states near the goal selects primitive action over policy as it helps agent to move away from the hallways and direct it towards goal whereas, states away from the goal follows actions. From figure-4, it is clear that SMDP Q-learning has obtained optimal policy using **mix of primitive action as well as options.**



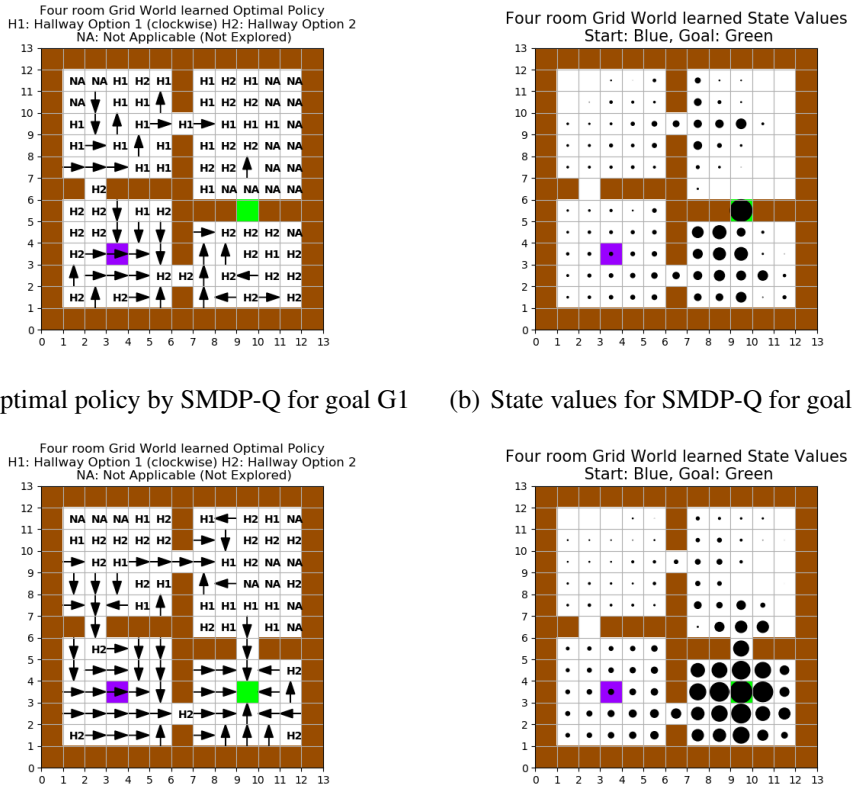
(a) Optimal policy by SMDP-Q (b) State values for SMDP-Q

Figure 4: Grid world of four rooms. Blue:Agent, Green:Terminal. The optimal policy-4(a) and associated state values-4(b) for Goal-G2 after training of 10000 episodes.

For goal-2 compared to goal-1, the major states follow primitive actions where as options are dominant in case of goal-G1. Mainly for room-1 policy plays major role, whereas for room 2 and 4 it requires both. **Near terminal state, dominance of policy make algorithm faster by increasing the learning rate.**

Answers-2 : Changed initial state to the centre of room 4

For the same terminal state, initial state is directed to center of room-4. **The change in state also causes the change in value function and optimal policy.** The optimal policy obtained in these cases are also mixtures of options as well as primitive actions.



(a) Optimal policy by SMDP-Q for goal G1 (b) State values for SMDP-Q for goal G1

(c) Optimal policy by SMDP-Q for goal G2 (d) State values for SMDP-Q for goal G2

Figure 5: Grid world of four rooms. Blue:Agent, Green:Terminal. The optimal policy-5(a) and associated state values-5(b) for Goal-G1 after training of 10000 episodes. Same way, the optimal policy-5(c) and associated state values-5(d) for Goal-G2

Change of state does not affect the role of primitive action and policy near the terminal states as well as far away from it. The major difference in both scenario (policy for same goal and change of initial state) is change in state value function. By comparing the fig.-3(a) and fig.-5(a), it is clear that the **state value of adjutant room of terminal and start state has higher state values. Change of the initial state directly affect the optimal policy, which varies in both scenario and finds the best possible route using primitive action and options.**

The value obtained in case of goal G2 is higher and shows nature of gradient. As goal-G1 is constrained by two wall these varies in this case. Options play vital role near terminal state of goal-G1, whereas, primitive actions play important role in case of goal-G2.

Answers-3: Intra-option Q learning

Deep Reinforcement Learning:

For contentious state and action space. policy gradient algorithm are implemented. [1]

Answers-1: Hyper-parameter Tuning

Answers-2: Report of hyper-parameters tuning

Answers-3: Report of the variation of hyper-parameters like hidden layer sizes, epsilon, mini-batch size, target frequency.

Answers-4: Observations and inferences of removal of the experience replay and/or the target network

References:

[1] “Rl course by david silver - youtube,”