AI 3000 / CS 5500 : Reinforcement Learning

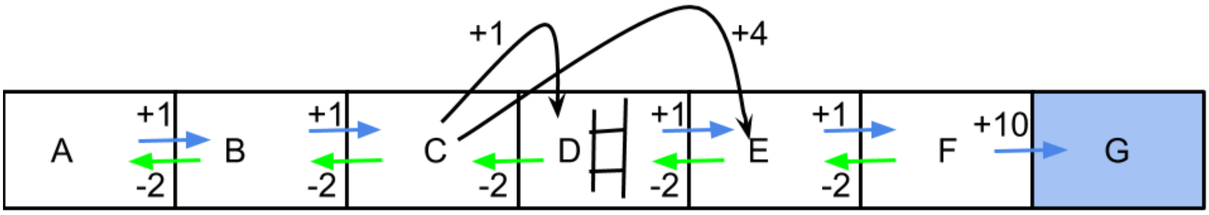
Assignment № 3

Due Date : 06/10/2023

Couse Instructor : Easwar Subramanian 21/09/2023

Problem 1 : Model Free Methods

Consider the MDP shown below with states $\{A, B, C, D, E, F, G\}$. Normally, an agent can either move *left* or *right* in each state. However, in state C, the agent has the choice to either move *left* or *jump* forward as the state D of the MDP has an hurdle. There is no *right* action from state C. The *jump* action from state C will place the agent either in square D or in square E with probability 0.5 each. The rewards for each action at each state s is depicted in the figure below alongside the arrow. The terminal state is G and has a reward of zero. Assume a discount factor of $\gamma = 1$.



Consider the following samples of Markov chain trajectories with rewards to answer the questions below

- $A \xrightarrow{+1} B \xrightarrow{+1} C \xrightarrow{-2} B \xrightarrow{+1} C \xrightarrow{+1} D \xrightarrow{+1} E \xrightarrow{+1} F \xrightarrow{+10} G$
- $A \xrightarrow{+1} B \xrightarrow{+1} C \xrightarrow{+1} D \xrightarrow{+1} E \xrightarrow{+1} F \xrightarrow{+10} G$
- $A \xrightarrow{+1} B \xrightarrow{+1} C \xrightarrow{+4} E \xrightarrow{+1} F \xrightarrow{+10} G$
- $A \xrightarrow{+1} B \xrightarrow{+1} C \xrightarrow{+4} E \xrightarrow{-2} D \xrightarrow{+1} E \xrightarrow{+1} F \xrightarrow{+10} G$
- $A \xrightarrow{+1} B \xrightarrow{+1} C \xrightarrow{+4} E \xrightarrow{-2} D \xrightarrow{+1} E \xrightarrow{+1} F \xrightarrow{-2} E \xrightarrow{+1} F \xrightarrow{+10} G$

(a) Evaluate $V(s)$ using first visit Monte-Carlo method for all states s of the MDP. (2 Points)

(b) Which states are likely to have different value estimates if evaluated using every visit MC as compared to first visit MC ? Why ? (2 Points)

(c) Fill in the blank cells of the table below with the Q-values that result from applying the Q-learning update for the 4 transitions specified by the episode below. You may leave Q-values that are unaffected by the current update blank. Use learning rate $\alpha = 0.7$. Assume all Q-values are initialized to -10. (2 Points)

s	a	r	s	a	r	s	a	r	s	a	r	s
C	jump	4	E	right	1	F	left	-2	E	right	+1	F

	Q(C, left)	Q(C, jump)	Q(E, left)	Q(E, right)	Q(F, left)	Q(F, right)
Initial	-10	-10	-10	-10	-10	-10
Transition 1						
Transition 2						
Transition 3						
Transition 4						

(d) After running the Q-learning algorithm using the four transitions given above, construct a greedy policy using the current values of the Q-table in states C, E and F. (1 Point)

(e) For the Q-Learning algorithm to converge to the optimal Q function, a necessary condition is that the learning rate, α_t, which is the learning rate at the t-th time step would need to satisfy the Robinns-Monroe condtion. In here, the time step t refers to the t-th time we are updating the value of the Q value of the state-action pair (s, a). Would the following values for learning rate α_t obey Robbins Monroe conditions ? (3 Points)

(i) $\alpha_t = \frac{1}{t}$

(ii) $\alpha_t = \frac{1}{t^2}$

(f) A RL agent collects experiences of the form $(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})$ to update Q values. At each time step, to choose an action, the agent follows a fixed policy π with probablity 0.5 or chooses an action in uniform random fashion. Assume the updates are applied infinitely often, state-action pairs are visited infintely often, the discount factor $\gamma < 1$ and the learning rate scheduling is appropriate.

(i) The Q learning agent performs following update

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t[r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)]$$

Will this update converge to the optimal Q function ? Why or Why not ? If not, will it converge to anything at all ? (2.5 Points)

(ii) Another reinforcemnt learning called SARSA agent, performs the following update

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

Will this update converge to the optimal Q function ? Why or Why not ? If not, will it converge to anything at all ? (2.5 Points)

Problem 2 : Game of Tic-Tac-Toe

Consider a 3×3 Tic-Tac-Toe game. The aim of this problem is to implement a Tic-Tac-Toe agent using Q-learning. This is a two player game in which the opponent is part of the environment.

(a) Develop a Tic-Tac-Toe environment with the following methods. (5 Points)

(1) An **init** method that starts with an empty board position, assigns both player symbols ('X' or 'O') and determines who starts the game. For simplicity, you may assume that the agent always plays 'X' and the opponent plays 'O'.

(2) An **act** method that takes as input a move suggested by the agent. This method should check if the move is valid and place the 'X' in the appropriate board position.

(3) A **print** method that prints the current board position

(4) You are free add other methods inside the environment as you deem fit.

(b) Develop two opponents for the Q-learning agent to train against, namely, a random agent and safe agent (5 Points)

(1) A **random agent** picks a square among all available empty squares in a (uniform) random fashion

(2) A **safe agent** uses the following heuristic to choose a square. If there is a winning move for the safe agent, then the corresponding square is picked. Else, if there is a blocking move, the corresponding square is chosen. A blocking move obstructs an opponent from winning in his very next chance. If there are no winning or blocking moves, the safe agent behaves like the random agent.

(c) The Q-learning agent now has the task to learn to play Tic-Tac-Toe by playing several games against **safe** and **random** opponents. The training will be done using tabular Q-learning by playing 10,000 games. In each of these 10,000 games, a fair coin toss determines who makes the first move. After every 200 games, assess the efficacy of the learning by playing 100 games with the opponent using the full greedy policy with respect to the current Q-table. Record the number of wins in those 100 games. This way, one can study the progress of the training as a function of training epochs. Plot the training progress graph as suggested. In addition, after the training is completed (that is after 10,000 games of training is done), the trained agent's performance is ascertained by playing 1000 games with opponents and recording the total number of wins, draws and losses in those 1000 games. The training and testing process is described below. (15 Points)

(1) Training is done only against the random player. But the learnt Q-table is tested against both random and safe player.

(2) Training is done only against the safe player. But the learnt Q-table is tested against both random and safe player.

(3) In every game of training, we randomly select our opponent. The learnt Q-table is tested against both random and safe player.

(4) Among the three agents developed, which agent is best ? Why ?

(5) Is the Q-learning agent developed unbeatable against any possible opponent ? If not, suggest ways to improve the training process.

[**Note** : A useful diagnostic could be to keep count of how many times each state-action pair is visited and the latest Q value for each state-action pair. The idea is that, if a state-action pair is visited more number of times, Q value for that state-action pair gets updated frequently and consequently it may be more close to the 'optimal' value. Although, it is not necessary to use the concept of **afterstate**, it may be useful to accelerate the training process]

ALL THE BEST