Analysis of Q-Learning, SARSA and Monte-Carlo Algorithms on Taxi Environment

Hamad Abdul Razzaq
hr06899@st.habib.edu.pk
Habib University
Karachi, Pakistan

Muhammad Zain Yousuf
my06200@st.habib.edu.pk
Habib University
Karachi, Pakistan

Laiba Jamil
lj06272@st.habib.edu.pk
Habib University
Karachi, Pakistan

***Abstract*—This paper aims to compare and contrast three different Reinforcement Learning Algorithms on a Taxi Environment. The three algorithms to be compared are Q-Learning, SARSA, and Monte-Carlo Every-Visit Algorithms.**

***Index Terms*—Reinforcement Learning, SARSA, Q-Learning, Monte-Carlo**

I. Introduction to Problem: Taxi Environment

The Taxi Environment is a Two Dimensional Environment in which the problem we have to solve is that a Taxi Driver should pick up passenger from its pick-up place and drop him off at his desired location. While doing so, the driver must make the smallest number of moves (shortest path). This optimal shortest path problem can be solved via Reinforcement Learning framework and in this paper, we will try three different Reinforcement Learning techniques, Monte-Carlo (Every Visit), Q-Learning and SARSA for solving this optimality problem. But first, we would model this problem as an Markov Decision Process (MDP).

II. Modelling the Problem as MDP

A. State Space

In order to determine a state in our environment, we must consider four parameters, the x position of the driver, the y position of the driver, the pick-up position of the passenger and the destination position of the passenger. Since we are operating on a 5×5 Grid, and the passenger pick-up positions are 4 and the passenger can also be in the car so 5 different pick up positions, and the destination positions in total are 4 [1]. So, a state is a 4-tuple and can be represented as:

$$s = (d_x, d_y, p_p, p_d)$$

Where,
d_x: x-position of the driver
d_y: y-position of the driver
p_p: Passenger Pick-up
p_d: Passenger Drop-off

In total, there are $5 \times 5 \times 5 \times 4 = 500$ States.

B. Action Space

The actions that can be taken by the driver is to move Up, Down, Left, Right, Pick up the passenger and Drop Off a passenger. So the action space is:

$$A = \{\text{Up, Down, Left, Right, Pick, Drop}\}$$

C. Reward

Since we want to minimize the distance, therefore, we will model the reward function as follows:

- If a Driver moves in any of the four directions, it gets a reward of -1.
- If a Driver Drops the passenger off at the wrong location, it will get a highly negative reward of -10 since its an highly undesirable result.
- If a Driver Drops the passenger off at the right location, it will get a highly positive reward of 20 since its a fruitful result

III. Training the Agent via Q-Learning

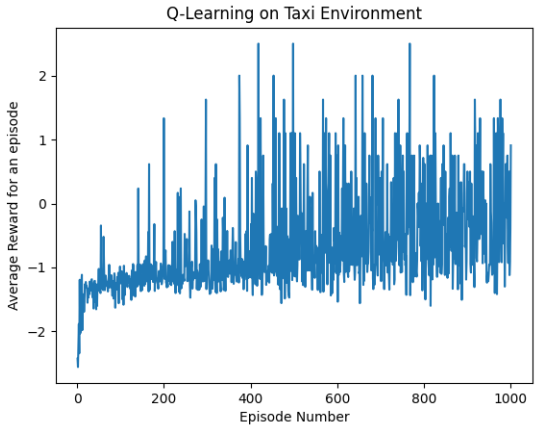
We will first train the algorithm with the help of Q-Learning technique. In this technique, the parameter set-up is:

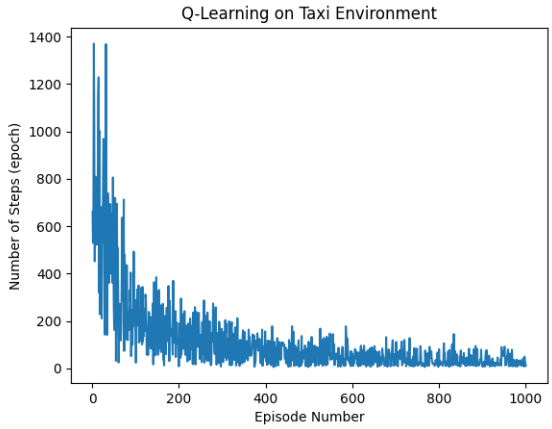
$$\gamma = 0.7$$

$$\alpha = 0.1$$

$$\epsilon = 0.1$$

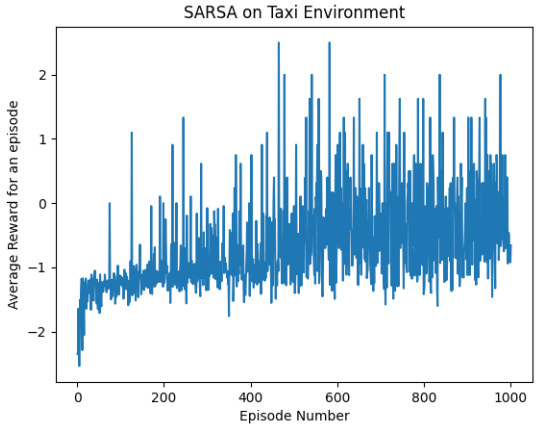
In order to determine the convergence rate, we look the increase in average reward with the increase in episode and the total number of steps in the episode (epochs) as we run episodes. The convergence rate would be faster if we get a higher average reward in as much smaller number of episodes, and similarly, if we the number of steps in each episode decreases in smaller number of episodes, then the convergence rate is fast. For Q-Learning the following two-graphs were obtained:

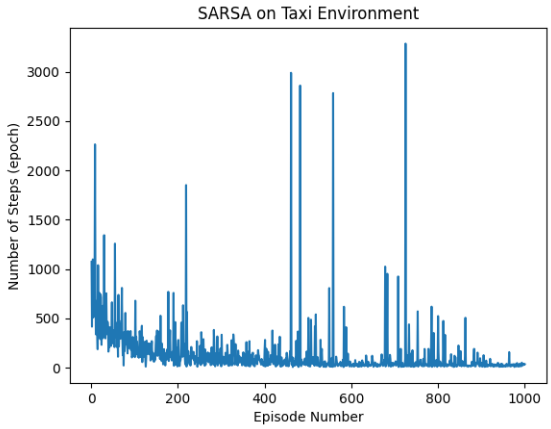




IV. Training the Agent via SARSA

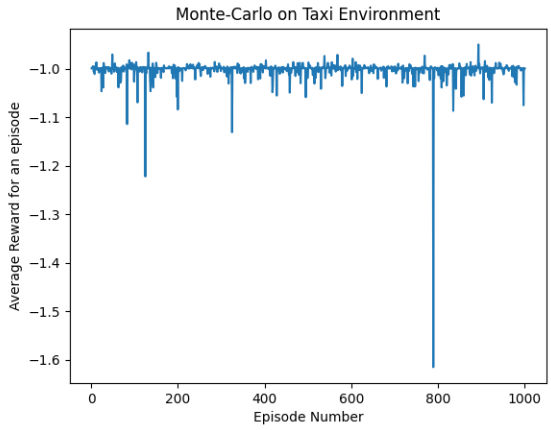
While training the agent with SARSA, the same values for the parameters were taken and the following results were obtained:

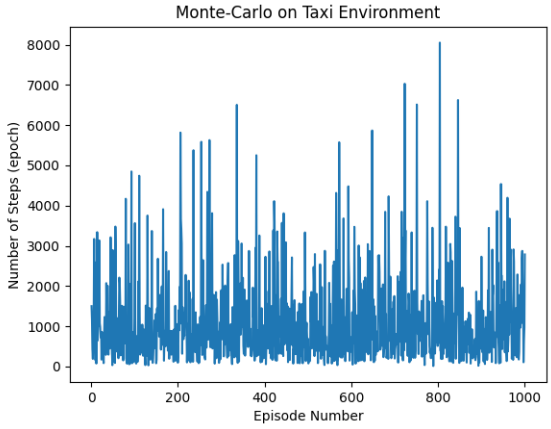




V. Training the Agent via Monte-Carlo

We implemented an Every-Visit Monte-Carlo Algorithm and improve our policy after every episode. The results obtained in case of this technique are as follows:

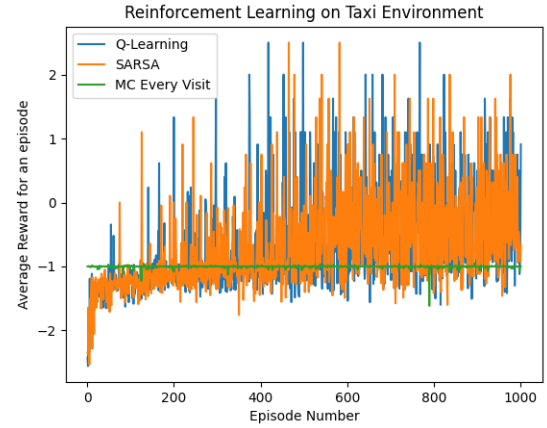


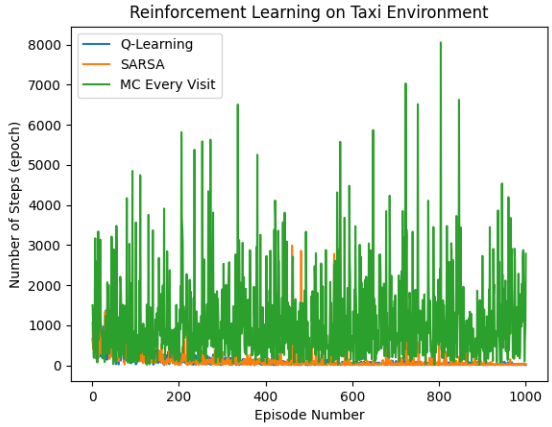


VI. Analysis

For comparing the three given techniques, the results of all the three algorithms were plotted together. We get the

following two graphs:





Also, The average Reward obtained in both the three techniques are:

- Q-Learning = 0.13
- SARSA = -0.09
- Monte-Carlo = -1.002

In the light of the above results, we see that after 1000 episodes, the Monte-Carlo is still showing an ambiguous behaviour, which means that it is yet to converge. SARSA learning on the other hand, increases the average reward after 1000 episodes, but there is still some peaks that can be seen in the epoch, graph, indicating that the convergence sometimes is not optimal. Lastly, for Q-Learning, we see that the average reward is also increasing and the number of steps is also decreasing as we run more episodes. As a result, we see that Q-learning outperforms the other two techniques because of its greedy Nature.

REFERENCES

[1] https://towardsdatascience.com/solving-the-taxi-environment-with-q-learning-a-tutorial-c76c22fc5d8f

[2] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. Cambridge, MA: The MIT Press, 2020.