COMP 138 RL: Homework 2

Nanxi Liu

March 5, 2025

1 Introduction

In reinforcement learning, agents can also learn without a model whose knowledge the agent relies on to make decisions. The Monte Carlo (MC) method achieves exactly this goal. It only provides the agents with experiences, and learning happens in episodes. We may think of the agent as a player in a game. The player plays a game many times, and even though the player does not have any prior knowledge of the game, it takes actions based on the instructions of an experienced player. The agent may or may not choose to take on the experiences of the other player based on its setting. The agent's learning from each episode is only updated after an episode has ended and not when it's in the middle of an episode. Due to the ongoing process of action learning, the MC problems are non-stationary. For instance, the agent may run into new states it has not seen before in each episode. There are several different types of MC algorithms. In the textbook by Sutton and Barto, the authors first introduced the First-Visit MC prediction algorithm, where the agent learns the values of a state in policy PIE by averaging the returns of its first visits to that state in each episode. Another useful way to use MC algorithms is to estimate the Q table, or the values of state-action pairs. To maintain exploration of all states, the idea of the Exploring Start (ES) MC algorithm is introduced. This algorithm ensures that every state-action pair has a possibility to be chosen. It also updates the Q table by averaging the returns of a state-action pair in every step of the episode to make sure that the values are updated effectively. Another algorithm explored in this paper is the Off Policy MC Control algorithm, where the episodes are generated by a soft (behavior) policy, but the actual updates of the Q table apply to that of a target policy. Due to the structure of only examining the actions in an episode that matches an action in the target policy, the off policy method has a relatively slow learning, as the updates begin at the tail of an episode and may not be able to reach the starting states as often. Both the ES and Off Policy Control algorithms will be discussed in the following sections.

2 Goal

The author is curious about the convergence speed of the ES and Off Policy algorithm, as well as their effectiveness during implementation. The goal of this paper is to demonstrate the learning abilities of both algorithms in practice, and discuss any outstanding properties that the author finds interesting.

3 Problem Description

The problem studied in this paper is a racetrack problem, where the agent is a racecar that tries to cross a right turn efficiently to reach the finish line. The racetrack is given as grids, and the car may move vertically and horizontally. Therefore, the car has two velocity components - vertical and horizontal. Since the speed of the car can't be negative, it can only move in the up or right direction. Each velocity component is an integer number that represents the number of grids it crossed in a time step. The velocity components can't both be zero at the same time except at the starting line and neither can exceed the value of 4. If the car crashes into the edges during the generation of an episode, it is simply put back to a random starting position with both velocity components 0. An episode always ends with the car reaching the finishing line. The velocities may change by 1, -1, or 0 at each time step, giving rise to a total of 9 actions. The reward at each step is -1. There is also a 0.1 chance that the velocity increments at any given time step to be zero, regardless of the actions taken.

4 Experiment Design

The experiment tested two MC algorithms - ES and Off Policy Control. Each algorithm learns 30000 episodes, and each experiment is repeated 20 times. While the author was hoping to train the agent on more episodes and run the experiment for more than 100 times, the training has proven to take quite long, therefore its implementation is quite limited. There are two racetracks that the agents have been trained on. The racetracks are very similar to the ones given in exercise 5.12 of the Sutton and Barto textbook. The behavior policy used in the training are epsilon greedy algorithms with a decaying epsilon. The epsilon remains 0.5 for the first half of the episodes, then gradually decays by 0.001 at each episode till it reaches 0.1. The discount factor used in the algorithms is 0.9. See below for the pseudocode for the ES and Off Policy Control algorithms from the Sutton and Barto textbook.

5 Hypothesis

The author believes that the Off Policy Control method may not explore as much as the ES algorithm in a given time period due to its bias towards tail episodes. This may also lead to higher variance in the learning returns.

6 Results

The following figures are the learning returns comparison between the ES and Off Policy Control algorithm for both tracks. Attached below are also comparisons of the learned best route of both algorithms for the same track. As we can see, since the Off Policy Control method (drawn in red) learns from the behavior policy, it has a higher return than the ES method to begin with. Then, the learning curve spiked around 2500 time steps, and dropped when it approached 5000 time steps. The curve then slowly but steadily increases and converges around 25000 time steps. For the ES algorithm (drawn in blue), it started with a lower return than the red curve but steadily increased and converged around 22500 time steps. The same is true for both tracks. Compared with the racetrack trajectory of the ES algorithm, the trajectory produced by the Off Policy Control algorithm seems to be smoother.

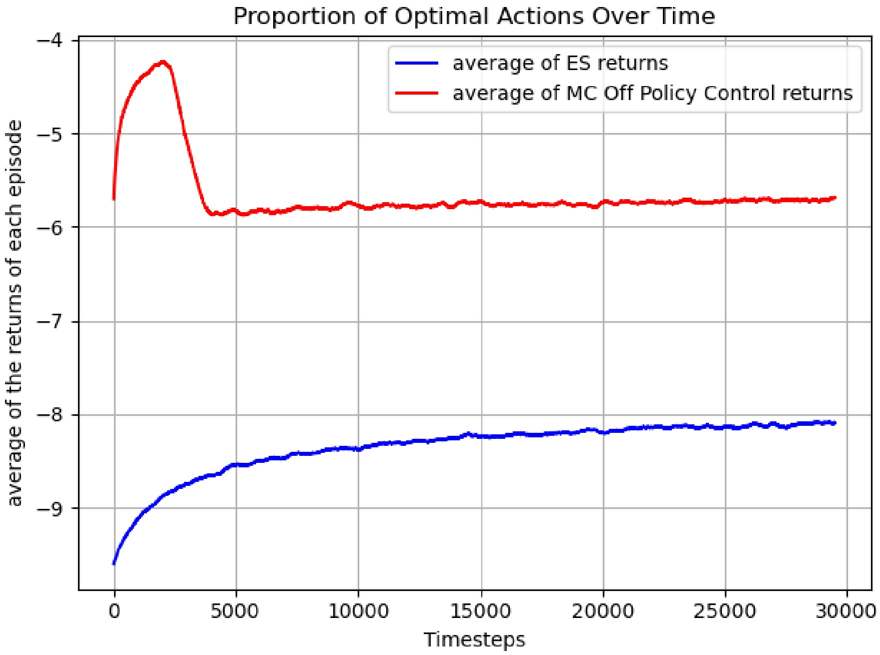

Figure 1: average returns for track 1

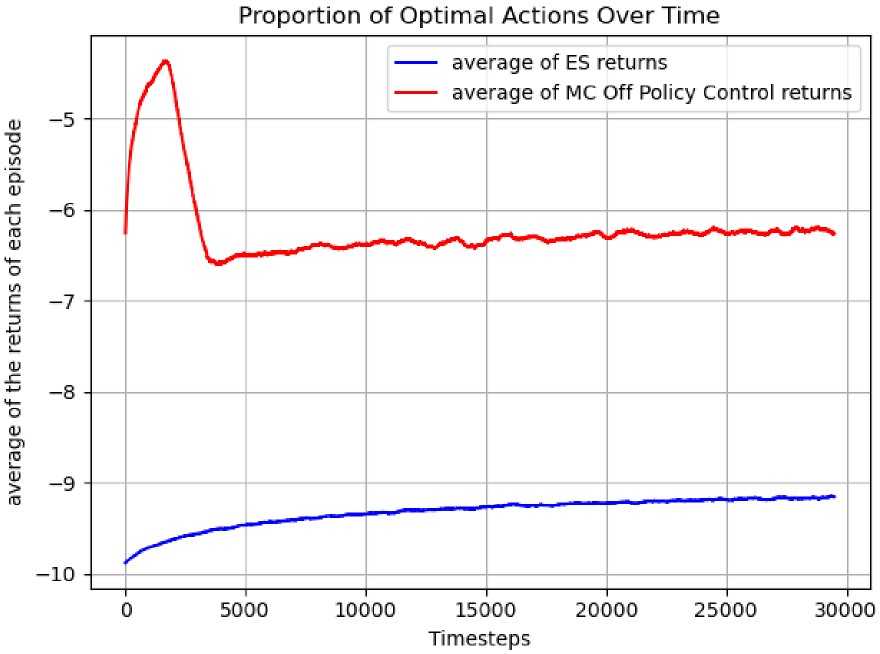

Figure 2: average returns for track 2

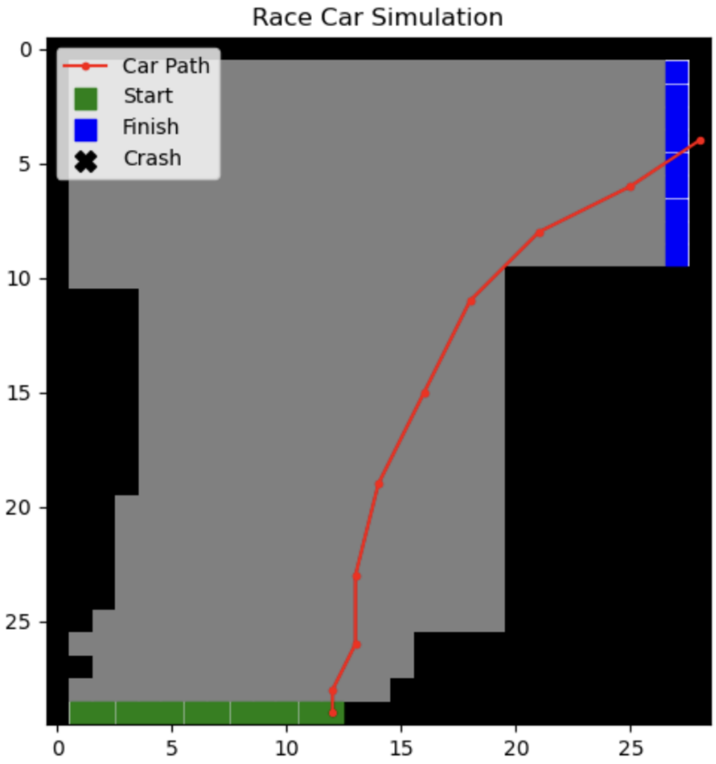

Figure 3: ES trajectory for track 1

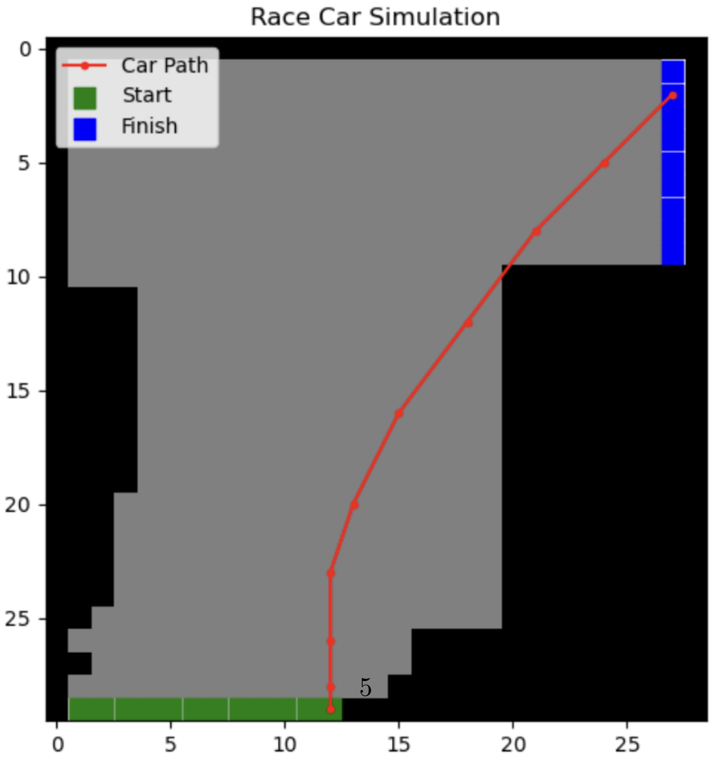

Figure 4: Off Policy Control trajectory for track 1

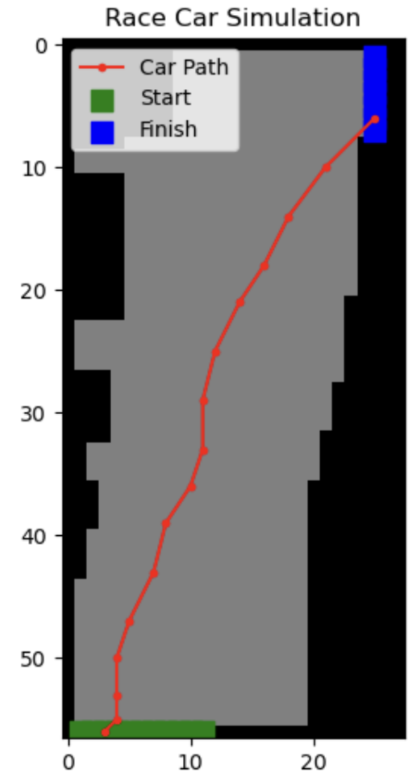

Figure 5: ES trajectory for track 2

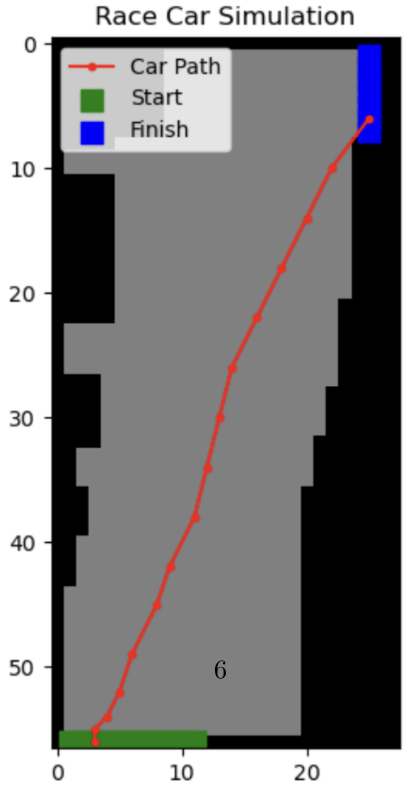

Figure 6: Off Policy Control trajectory for track 2

7 Discussion

The result of the experiment has confirmed the author's assumption about the learning of both algorithms. Due to the instability of importance sampling, the Off Policy Control algorithm spiked at the beginning of the learning process, but quickly plateaued. The ES algorithm, on the other hand, has steadily increased throughout the learning process due to more thorough exploration.

8 Conclusion

In this paper, the author demonstrated the properties of both the ES and Off Policy Control MC algorithms through analysis of the returns and agent trajectory of each method. The Off Policy Control method doesn't cover exploration as much as the ES algorithm due to importance sampling, but it is able to use the behavior policy to plan out smoother routes.

References

9 Appendix

See the Python script and instructions on how to replicate the experiment using the following link: `https://github.com/Nanxi-Flaneuse/Reinforcement-Learning/tree/f725ddf5889d609117c8b6b35c083b3bead0da95/assignments/Monte%20Carlo%20off-policy%20race%20track%20problem`

Due to limited epochs used in Off Policy learning, some start states are not properly updated. Therefore, when running the code to generate the trajectory, there is a small chance that the agent will crash into a wall. In that case, just run the image generation again, and you will probably obtain a working trajectory.