Second Project

Sara Ghavampour
9812762781
Reinforcement Learning Spring2023

Abstract **– The goal of this project is to implement and compare three learning algorithms in RL, Two on-policy and one off-policy.**

Index Terms – Monte Carlo control, TD, Q-Learning, Sarsa

INTRODUCTION

Monte Carlo methods choose different approach than DP for learning. These methods unlike DP methods do not need a complete model of environment and update their estimates by averaging outcomes of random samples. On the other hand, TD methods are a combination of these two extremes. These methods inherent bootstrapping from DP to address variance of MC and sampling, but they don't wait for the end of episode to update estimates.

CODE

Wrapper is a class that inherents from gym.warpper. It changes the action space size to 10 and creates extra diagonal moves using two consequent base moves. For each step function call, It checks for availablity of the path using P matrix and then implements the move using step functions.

EXPERIMENTS

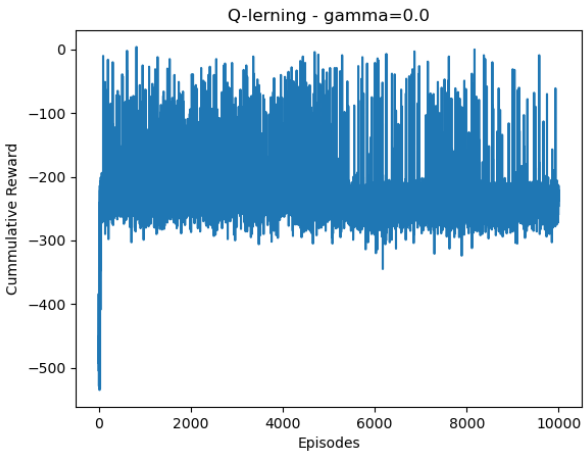

Figure 1

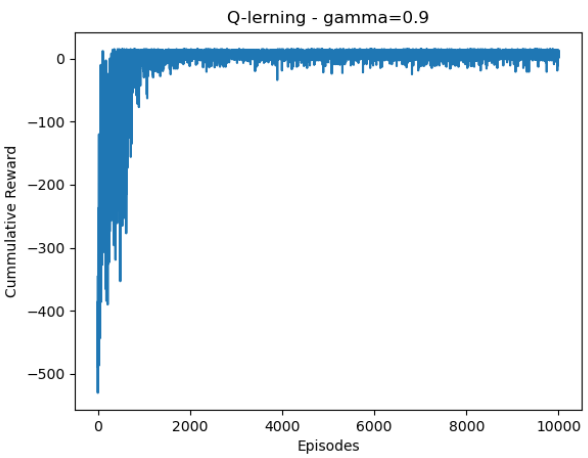

FIGURE 2

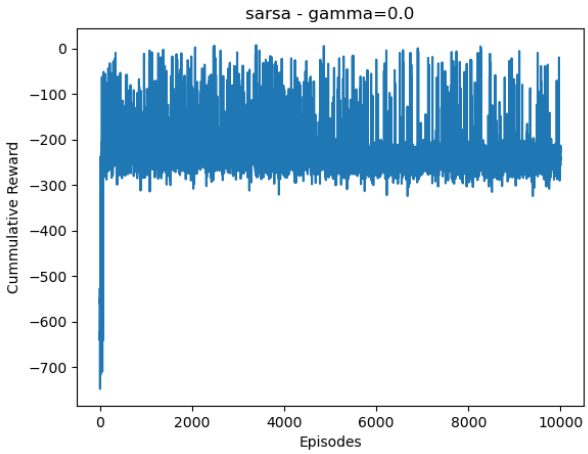

Figure 3

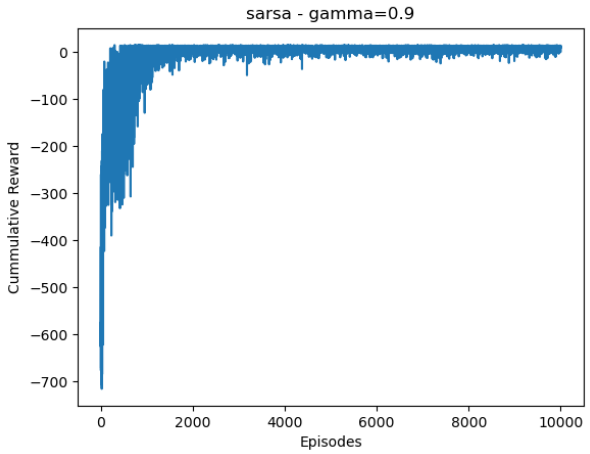

Figure 4

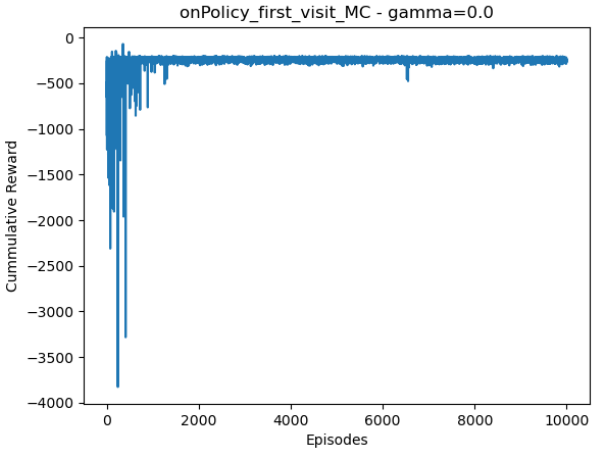

Figure 5

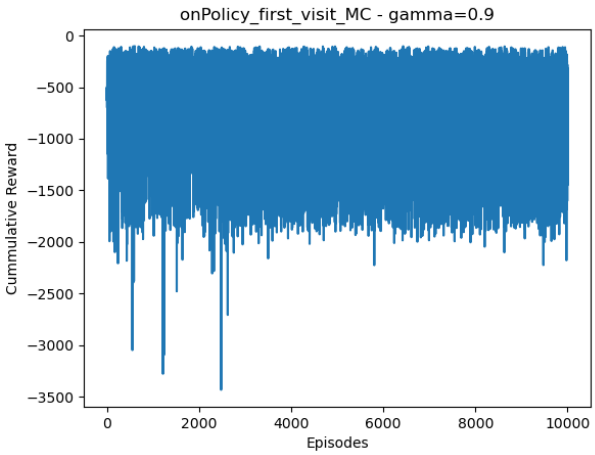

Figure 6

Figures 5 and 6 show that MC has high variance and even after 10,000 episodes, It wasn't able to learn a sufficient policy anی rewards are high also choice of gamma does not effect much on the performance.

By comparing figure 3 with 4 and 1 with 2, the effect of gamma is obvious. Gamma equal to zero caused more variance ans lower rewards in both sarsa and q-learning. The reason for this is that low gamma mostly considers immediate rewards, but high gamma tries to take future rewards into account which is helpful for reducing variance also training is faster with 0.9 because policy is getting improved faster.

Usually sarsa chooses longer but safer paths than q- learning like walking on the cliff example. This is because sarsa takes randomness of choosing actions (epsilon greedy) into account. However performance of sarsa and q-learning in these tests are mostly similar. I believe the answer holds in the fact that in this particular environment choosing actions randomly is not followed by any high risk state like cliff example so their difference isn't obvious in these tests.

Sarsa is on-policy and simultaneously chooses actions and improve a same policy. In contrast q-learning is off policy, meaning that it follows an exploratory policy called behavioural policy and improves another policy called target policy. One significant advantage of this is that it tries to handle exploration-exploitation dilemma.