Machine Learning

COMP5611M

Coursework2: Reinforcement Learning

Daolin Sheng ml192ds

Experiments

1. All of experiment under the same parameters except for the epsilon greedy(ϵ-greedy):

(1). learning_rate = 0.01

(2). gamma = 0.99

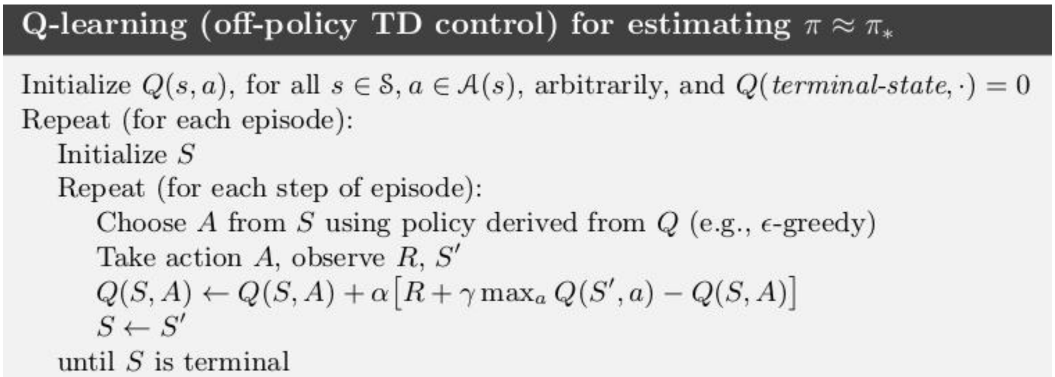
(3). maxStepsPerEpisode = 2500 (max number of steps possible in a single episode)

(4). nbOfTrainingEpisodes = 1100

2. The Q-learning and SARSA algorithm use the same policy: ϵ-greedy.

3. In all experiments, I will use the function 'updateEpsilon(self, episode_counter)' to upadte the epsilon value from the lager to lower(from explorartion to exploitation).

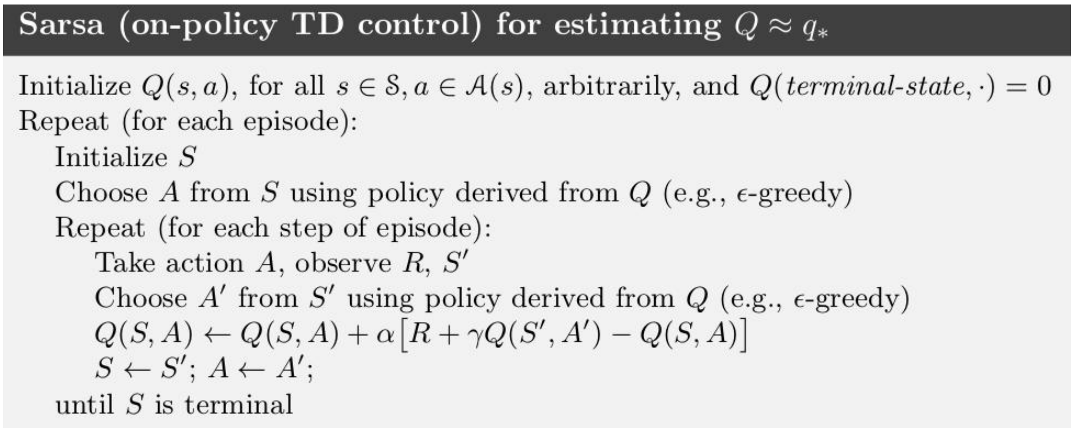
4. The Q-learing algorithm:



Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

```
Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
    Initialize $S$
    Repeat (for each step of episode):
        Choose $A$ from $S$ using policy derived from $Q$ (e.g., $\epsilon$-greedy)
        Take action $A$, observe $R$, $S'$
        $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$
        $S \leftarrow S'$
    until $S$ is terminal
```

5. The Sarsa algorithm:

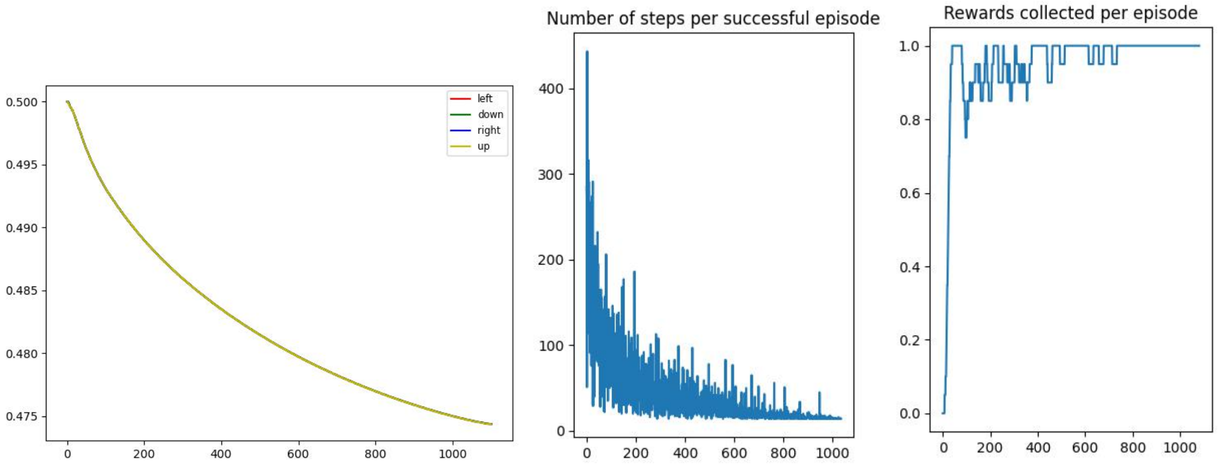


Sarsa (on-policy TD control) for estimating $Q \approx q_*$

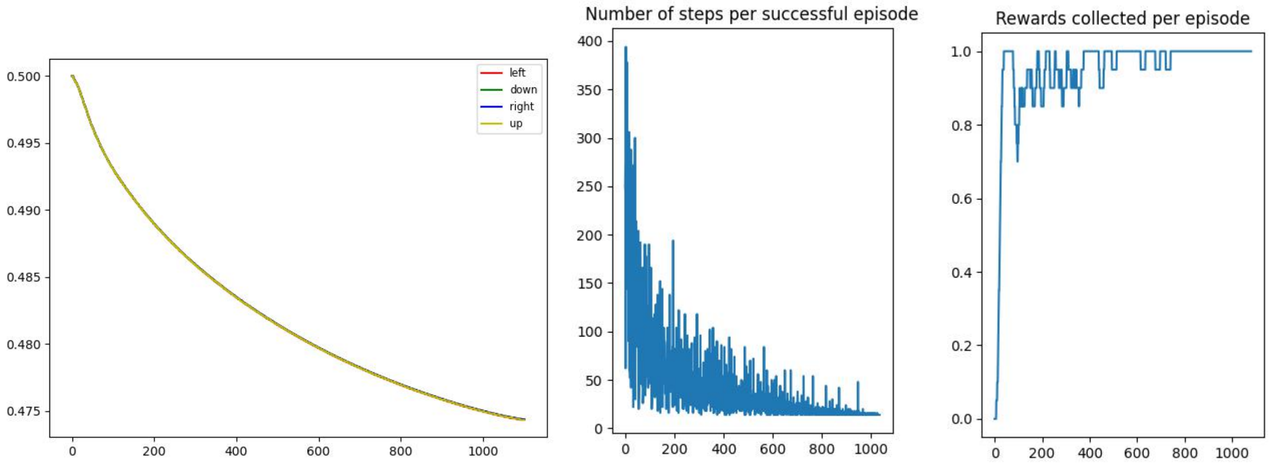
```
Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
    Initialize $S$
    Choose $A$ from $S$ using policy derived from $Q$ (e.g., $\epsilon$-greedy)
    Repeat (for each step of episode):
        Take action $A$, observe $R$, $S'$
        Choose $A'$ from $S'$ using policy derived from $Q$ (e.g., $\epsilon$-greedy)
        $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma Q(S',A') - Q(S,A)\big]$
        $S \leftarrow S'$; $A \leftarrow A'$;
    until $S$ is terminal
```

* *References : the two Images comes from the https://www.google.com.*

1, Q-learning: ϵ = 0.8:

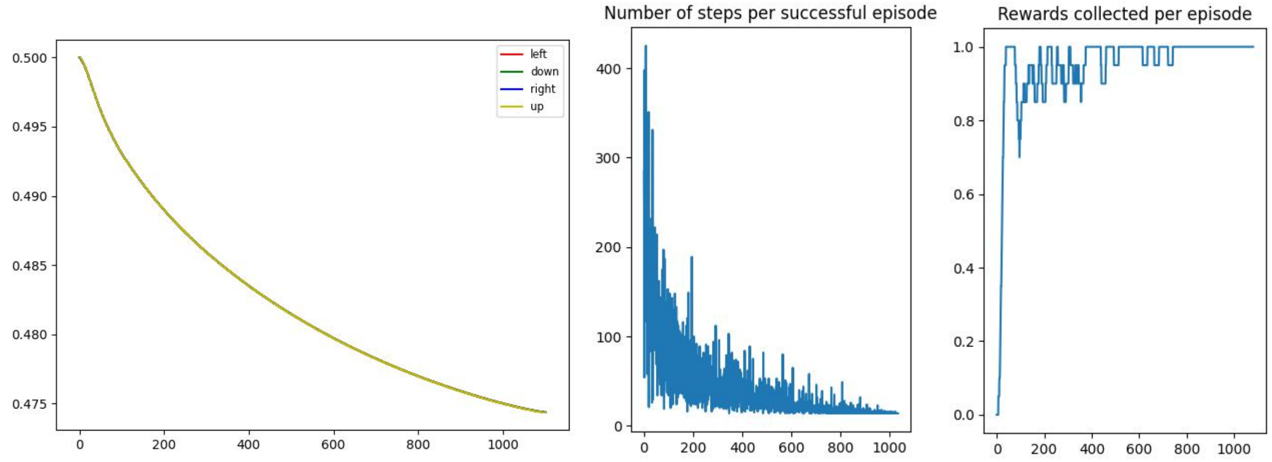


2, SARSA: ϵ = 0.8:

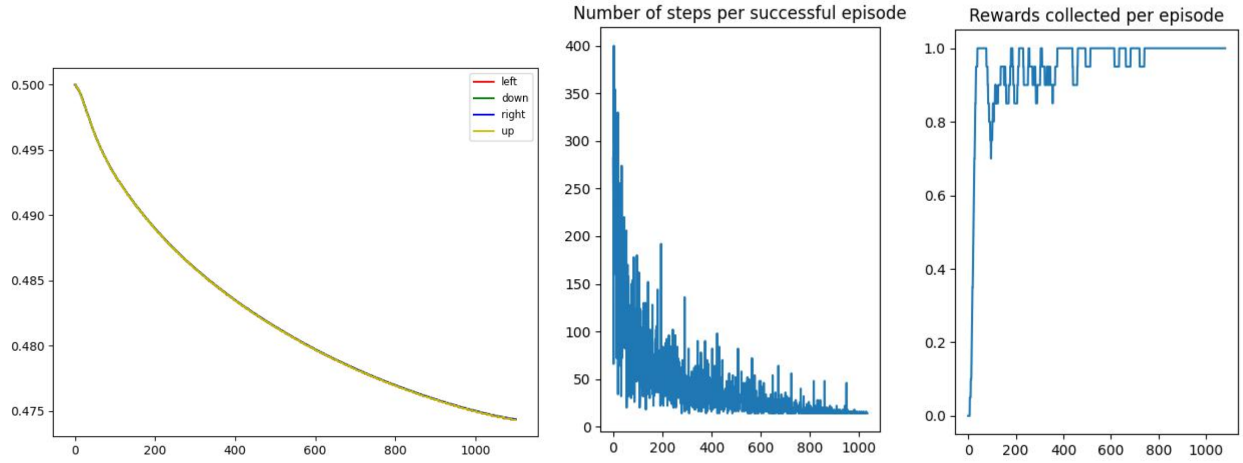


At the same epsilon value 0.8, Q-learning is an off-policy TD control policy , and SARSA is an on-policy TD control method. By comparing plots of these two algorithms, we can the truth that the average of the number of steps per successful episode of Q-learning is less than that of SARSA due to Q-learing always attempts to follow the optiomal path which is the shorted one. By way of contrast, the sarsa algorithm will converge to a much safer route that keeps it well away from the cliffff, even though it takes longer.

3, Q-learning: $\epsilon = 0.1$:



4, SARSA: $\epsilon = 0.1$:



At the same epsilon value 0.1 (lower than 0.8), which means Q-learning and Sarsa use the higher exploitation in this experiment. So, the avarege number of the number of steps per successful episode of these algorithms are less than previous two experiments in epsilon 0.8.