1. Introduction

Off-policy Watkins's Q(λ) control methods using function approximation is applied at part 2 experimentation. Watkins's Q(λ) is defined as following function:

$$Q^{\pi}(s_t, a_t) = R(s_t, a_t) + \sum_{\tau=t+1}^{T} \gamma^{\tau-t} R(s_\tau, \pi(s_\tau))$$

This function can be transformed as following:

$$Q^{*}(s_t, a_t) = R(s_t, a_t) + \gamma \max_{a} Q^{*}(s_{t+1}, a)$$

Which is called the Bellman Equations.

2. Q(λ) control methods implement

2.1 Q(λ) learning process

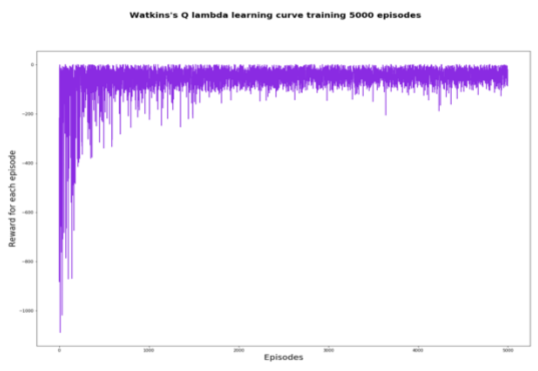
1) Initialize the action-state value function Q (s, a) = 0
2) Initialize the state, action random. initialize action with epsilon greedy policy.
3) Agent chooses an action based on Q value. Observe the reward R and the next state.
4) Update Q based on Reward R
5) Repeat 3) to 4) until the terminal state is reached.
6) Repeat 2) to 5) until Q is fully updated.

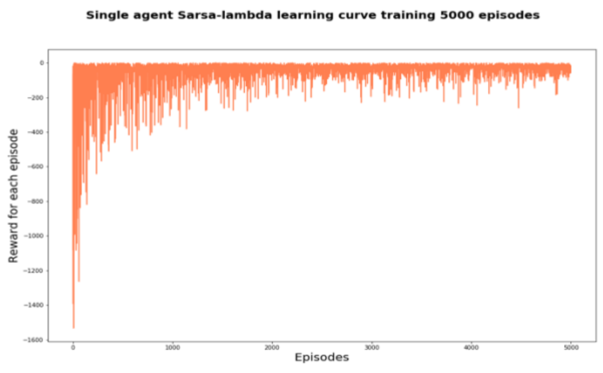
After receiving the reward R_{t+1}, the agent updates Q (s_t, a_t) by the following formula.

$$Q(s_t, a_t) \leftarrow \alpha \left\{ R_{t+1} + \gamma \max_{a} Q(s_{t+1}, a) \right\} + (1 - \alpha) Q(s_t, a_t)$$

It is known that Q is converged to the solution of the Bellman equation by this updated equation. if α is a sufficiently small number. If α is a large number, Q may not converge, but if α is too small, convergence will be delayed. Because replacing traces cannot extend to the use of function approximation straightforward, accumulating trace is used at this lab.

2.2 results compare





Both Q(λ) and Sarsa(λ) have good learning performance.
Q(λ) play better than Sarsa(λ) before the first 1000 episodes, because it can be observed that beginning these two methods need large numbers of steps to reach the target. Then the steps decrease with episodes training. However, Q(λ) have less probably at some episodes which need more steps, this can be obtained from the sparse degree on the graph. Therefore, this can prove Q(λ) learns quicker than Sarsa(λ). At the following episodes, Q(λ) performance is more stable than Sarsa(λ), Sarsa(λ) have a larger standard deviation. Watkins's Q(λ) initialize action and eligibility with epsilon greedy policy. With probability 1-epsilon to choose action based on maximum Q value function and initialize $\gamma\lambda\vec{e}$, or initialize random action choice and make $\vec{e}$ equal to 0. Next action is choosed from the maximum Q value. However, Sarsa(λ) initialize action randomly and choose the next action with epsilon greedy policy, which is with probability epsilon choose next action randomly or take next action based on maximum Q value. Moreover, Sarsa(λ) initialize $\vec{e}$ equal to 0 each time.

3 Other approaches

Deep Q-Network can be applied to mountain car as well. In Q-learning, it is impossible to learn a large-scale environment because the condition set is finite, so Q-learning is functionally approximated so that Q-learning can be performed even in continuous state space. Realize the function approximation of Q learning with neural network and apply deep learning technology