Homework Assignment 4

Zhaoyang Xu

September 29, 2020

Contents

1 Q-learning and SARSA (50 points)

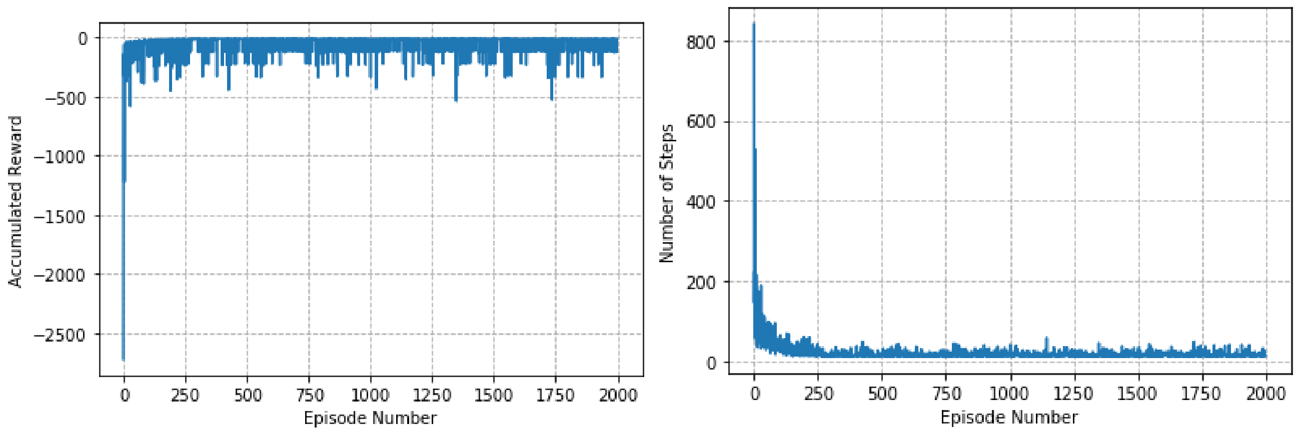
Sarsa is on-policy algorithm, we update Q by

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1} - Q(s_t, a_t))]$$

Q-learning is off-policy algorithem, we update Q by

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t))]$$

1.1



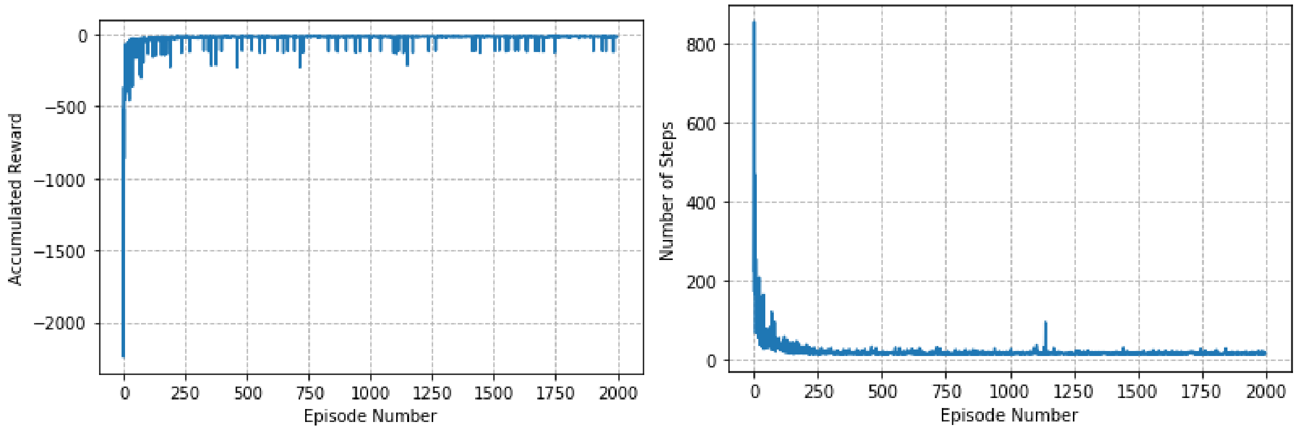
(a) accumulated reward versus episode number (b) steps versus episode number

Figure 1: **Apply Q-learning with** $\epsilon = 0.1$ **and** $\alpha = 0.1$

-12	-12	-11	-10	-9	-8	-7	-7	-6	-5	-4	-3
-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2
-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
-13	0	0	0	0	0	0	0	0	0	0	0

Figure 2: **Final state value by Q-learning**

1.2



(a) accumulated reward versus episode number

(b) steps versus episode number

Figure 3: **Apply Sarsa with** $\epsilon = 0.1$ **and** $\alpha = 0.1$

-15	-14	-13	-12	-11	-10	-8	-7	-6	-5	-4	-3
-15	-14	-13	-11	-10	-10	-8	-7	-6	-4	-3	-2
-16	-15	-14	-12	-12	-10	-9	-8	-7	-6	-2	-1
-17	0	0	0	0	0	0	0	0	0	0	0

Figure 4: **Final state value by Sarsa**

1.3

For the last 100 episodes, Sarsa gave a higher average accumulated reward. Sarsa is an on-policy algorithm, it follows the policy which is learning. Compared with Q-learning, it is more "safe". For the last 100 episodes, Sarsa has less probability to get rewards -100. So, Sarsa will give a higher average accumulated reward. Also, it is clear that Sarsa is higher in figure 5.

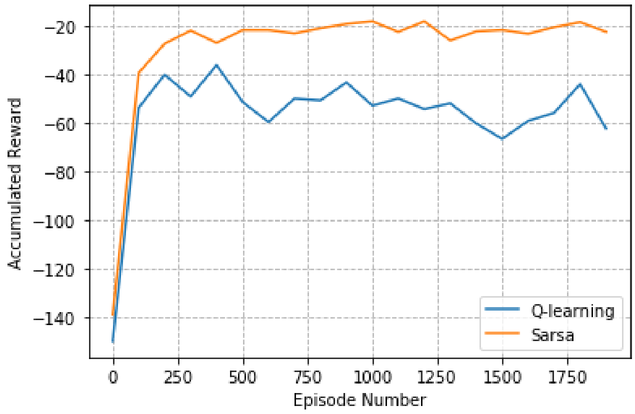


Figure 5: **Average accumulated reward (over 100 episodes) versus episode number**

1.4

Both of two algorithms will give a good results. But they are still not same. SARSA is on-policy, it learns action values relative to the policy it follows. While, Q-Learning is off-policy, it relatives to the greedy policy. When $\epsilon = 0$, actions will be choosed by policy derived from Q. For Sarsa, it will choose the next action then update Q. But for Q-learning, it updates the Q first, then select the next action.

2 Introduction of New Products (25 points)

Define two products as $a_1,\ a_2$. And from question, $\mu(a_1) = 0.5$, $\mu(a_2)$ is unknown. Now, we want to maximize the number of sales. It is obviously that the more rounds we offer the a with greater μ, the more numbers we sell. So, my strategy is exploring a_2 first, which to get the probability $\hat{\mu}(a_2)$. Then exploiting a which with a bigger μ. We write the algorithm down explicitly

```
Algorithm
Input : k
for t≤k do
    Sell a_2
end for
for t≥k+1 do
    Sell a = arg max μ(a_i)
```

We start with ϵT exploration rounds followed by $(1-\epsilon)T$ exploitation rounds. And let $\delta(\epsilon)$ denote the probability that we sell the wrong product in exploitation rounds.

$$
\begin{aligned}
\overline{R}_T &= \max_a \mathbb{E}[\sum_{t=1}^{T} r_t^a] - \mathbb{E}[\sum_{t=1}^{T} r_t^{A_t}] \\
&= \sum_a \Delta(a)\mathbb{E}[N_T(a)] \\
&\leq \Delta\epsilon T + \delta(\epsilon)\Delta(1-\epsilon)T
\end{aligned}
$$

where the first term is a bound by exploration phase and the second is by exploitation phase.

If $\mu(a_2) \leq 0.5$, we know that the a_1 is the best product.

$$\begin{aligned}\delta(\epsilon) &= \mathbb{P}(\hat{\mu}_{\epsilon T}(a_2) \geq 0.5) \\ &= \mathbb{P}(\hat{\mu}_{\epsilon T}(a_2) \geq \Delta + \mu(a_2)) \\ &= e^{-2\epsilon T \Delta^2}\end{aligned}$$

where the last line is by Hoeffding's inequality.

$$\overline{R}_T \leq \Delta\epsilon T + \delta(\epsilon)\Delta(1-\epsilon)T \leq \Delta\epsilon T + \delta(\epsilon)\Delta T = (\epsilon + \delta(\epsilon))\Delta T = (\epsilon + e^{-2\epsilon T\Delta^2})\Delta T$$

We take a derivative and equate to zero to get the minimize $\overline{R}_T$, which leads to $\epsilon = \frac{\ln 2T\Delta^2}{2T\Delta^2}$. We can also prove that the second derivative is positive. Thus, we obtain:

$$\overline{R}_T \leq \max\{\Delta T, (\frac{\ln 2T\Delta^2}{2T\Delta^2} + \frac{1}{2T\Delta^2})\Delta T\} = \max\{\Delta T, \frac{\ln(2T\Delta^2)+1}{2\Delta}\}$$

If $\mu(a_2) \geq 0.5$, we know that the a_2 is the best product.

$$\begin{aligned}\delta(\epsilon) &= \mathbb{P}(\hat{\mu}_{\epsilon T}(a_2) \leq 0.5) \\ &= \mathbb{P}(\hat{\mu}_{\epsilon T}(a_2) \leq \mu(a_2) - (-\Delta)) \\ &= e^{-2\epsilon T|\Delta|^2}\end{aligned}$$

Because a_2 is the best product, so the pseudo regret during exploration phase is 0.

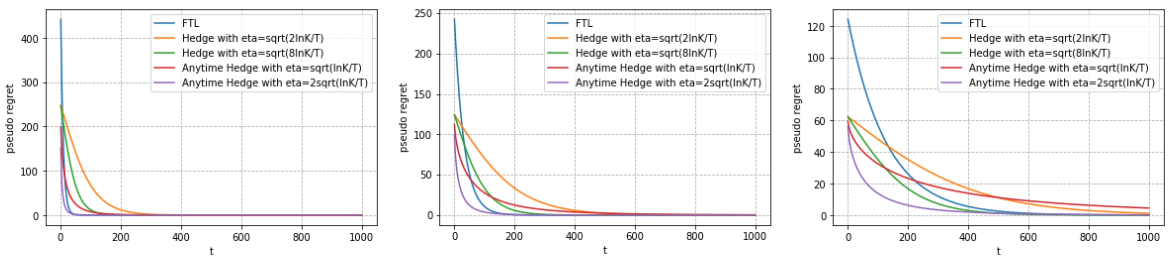
$$\overline{R}_T \leq \delta(\epsilon)|\Delta|(1-\epsilon)T = e^{-2\epsilon T|\Delta|^2}|\Delta|(1-\epsilon)T$$

$e^{-2\epsilon T|\Delta|^2}$ and $1-\epsilon$ are decreasing functions, so the whole term is decreasing with ϵ. Thus,

$$\overline{R}_T \leq e^{-2\epsilon T|\Delta|^2}|\Delta|(1-\epsilon)T \leq |\Delta|T$$

3 Empirical comparison of FTL and Hedge (25 points)

3.1



(a) $\mu = \frac{1}{2} - \frac{1}{4}$ (b) $\mu = \frac{1}{2} - \frac{1}{8}$ (c) $\mu = \frac{1}{2} - \frac{1}{16}$

Figure 6: **Pseudo regret of the five algorithms**

By definition on page 58 in notes,

$$\begin{aligned}\overline{R}_T &= \mathbb{E}[\sum_{t=1}^{T} l_t^{A_t}] - \min \mathbb{E}[\sum_{t=1}^{T} l_t^a] \\ &= \sum_{t=1}^{T}\sum_{a=1}^{K} p_t(a) l_t^a - \min \mathbb{E}[\sum_{t=1}^{T} l_t^a]\end{aligned}$$

where $p_t(a) = \frac{e^{-\eta_t L_{t-1}(a)}}{\sum_{a'} e^{-\eta_t L_{t-1}(a')}}$.

It is obvious that greater η make algorithms tighter. Also, with smaller μ, Anythime Hedge algorithms shows better than Hedge algorithms.

3.2

When $\mu = \frac{1}{2} - \frac{1}{4}$, it leads to higher regret. The μ becomes bigger, the regret becomes lower. From fig.6 we can see that, with a smaller μ, all the algorithms show tigter. It means that the algorithms depend on μ.
With t become greater, the regret become smaller. It is easy to understand that with more round, the algorithm will 'learn' more, and it will choose a better action.

3.3

For follow the leader algorithm,

$$R_T = \sum_t l_t(a_t) - \min \sum_t l_t$$

and each round we play $a_t = arg \min \sum_i^{t-1} l_i$, which is the best choice based on all past rounds. Lets consider a sequence with 0 and 1. If the $X_i = 0$, then $X_{i+1} = 1$. X_i always shows different with the previous one.