Reinforcement Learning HW 1

Wenzhe Liu S1631854

February 26, 2017

1 Multi-armed Bandits

1.1 The effect of the temperature parameter on the function

According to a soft-max distribution (i.e., Gibbs or Boltzmann distribution) as follows:

$$Pr(A_t = a) = \frac{e^{\beta H_t(a)}}{\sum_{b=1}^{k} e^{\beta H_t(b)}},$$

Where the denominator is the normalization part, and k denotes the total number of actions, and a denotes the taken action and $\beta = \frac{1}{\tau}$ and $\beta > 0$.
Taking two actions as an example, the formula can be written as follows:

$$Pr(A_t = a_1) = \frac{e^{\beta H_t(a_1)}}{e^{\beta H_t(a_1)} + e^{\beta H_t(a_2)}},$$

Then the formula is divided by $e^{H_t(a_1)\beta}$ for both numerator and denominator, and it becomes:

$$Pr(A_t = a_1) = \frac{1}{1 + e^{-\beta(H_t(a_1) - H_t(a_2))}},$$

When β is small, the probability for each action is distributed equally, so it means more exploration should be taken. When β is large, the policy is greedy, so it means more exploitation should be taken.

1.2 The optimistic initial value example

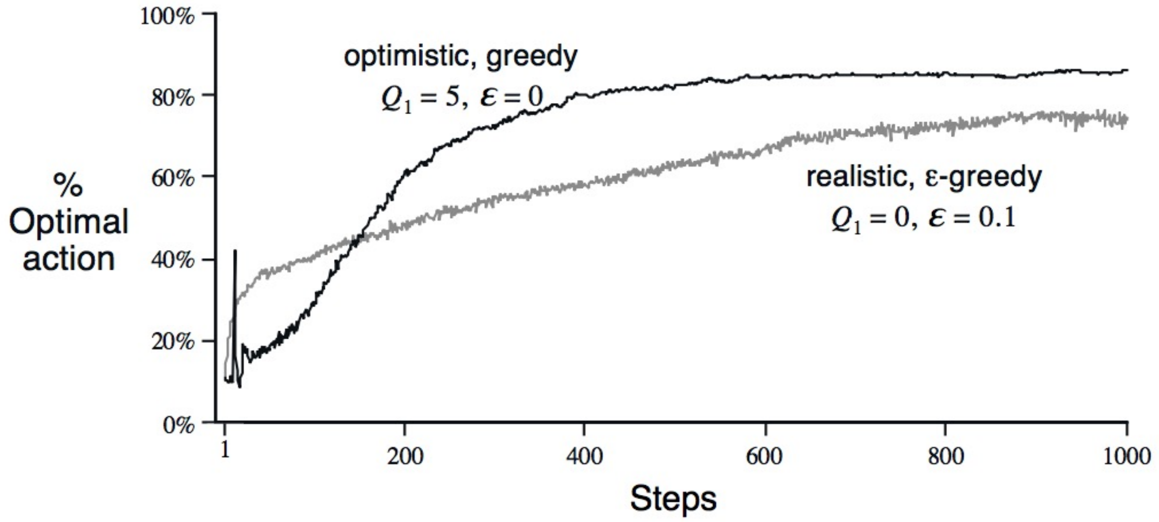


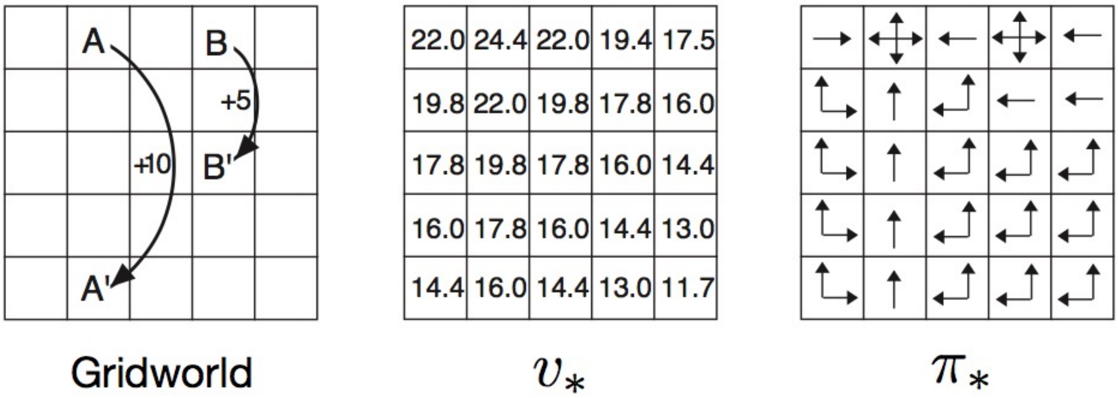
Figure 1: The effect of optimistic initial action-value estimates on the 10-armed testbed.

1. How do you explain the oscillations and spikes in the early part of the curve for the optimistic method?
   (1)According to Figure 1, there are some oscillations in the early part of the curve. It is because this optimism encourages action-value methods to explore. Whichever actions are initially selected, the reward is less than the starting estimates; the learner switches to other actions, being "disappointed" with the rewards it is receiving. The result is that all actions are tried several times before the value estimates converge(Cited). The system does a fair amount of exploration even if greedy actions are selected all the time.
   (2)10 actions each leads to a reduction, but the optimal action often leads to the least reduction, so greedy will choose this action with high probability. This greedy algorithm leads to spike at 20. But later, with the decrease of exploration, the spike is disappeared.

2. What makes this method perform differently on particular early plays?
   In the early steps, the optimistic method performs worse because whichever actions are initially selected, the reward is less than the starting estimates. Therefore,it explores more, but eventually it performs better because its exploration decreases with time. However, for realistic $\epsilon-$greedy method, whichever actions are initially selected, the reward is more than the starting estimates. So it exploits more since it will always choose the action with high probability.

2 Value Functions

2.1 The grid world example



22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

Figure 2: Optimal solutions to the gridworld example.

Optimal value function is as follows:

$$V_*(s) = \max_a E_{\pi^*}\Big[\sum_{k=0}^{\infty} \gamma^k R_{r+k+1} | S_t = s, A_t = a\Big].$$

According to Figure 2, from state A, all four actions yield a reward of +10 and take the agent to A'. From state A', after taking 4 up operations, it reaches A again. Repeat the procedure again and again. So according to the procedure above, the value can be expressed as follows:

$$V_*(s) = 10 + 0 * \gamma^1 + 0 * \gamma^2 + 0 * \gamma^3 + 0 * \gamma^4 + 10 * \gamma^5 + 0 * \gamma^6 + ...$$

Therefore,

$$V_*(s) = 10 \sum_{n=0}^{\infty} \gamma^{5n}$$

Calculate the sum of geometric progression,

$$V_*(s) = 10 \lim_{n \to \infty} \frac{1 - \gamma^{5n}}{1 - \gamma^5}$$

Assumes that $\gamma = 0.9$ and take the limit as n goes to infinity, and the formula can be expressed as follows:

$$V_*(s) = \frac{10}{1-\gamma^5},$$

so the optimal value of the best state is 24.419.

3 Q-Learning to play the Enduro game

3.1 Discretisation

If we use the continuous state spaces, value iteration becomes impractical as it requires to compute, for all states s belongs to S. Using Markov chain approximation, continuous state space dynamics model can be turned into the discretized MDP so that the action space can be reduced to a finite set. Policy and value function for the discrete states are optimal for the discrete MDP, but typically not for the original MDP. That is to say, the action that the agent takes in each time is not the optimal, but it is close to the original optimal action.

3.2 Random agent

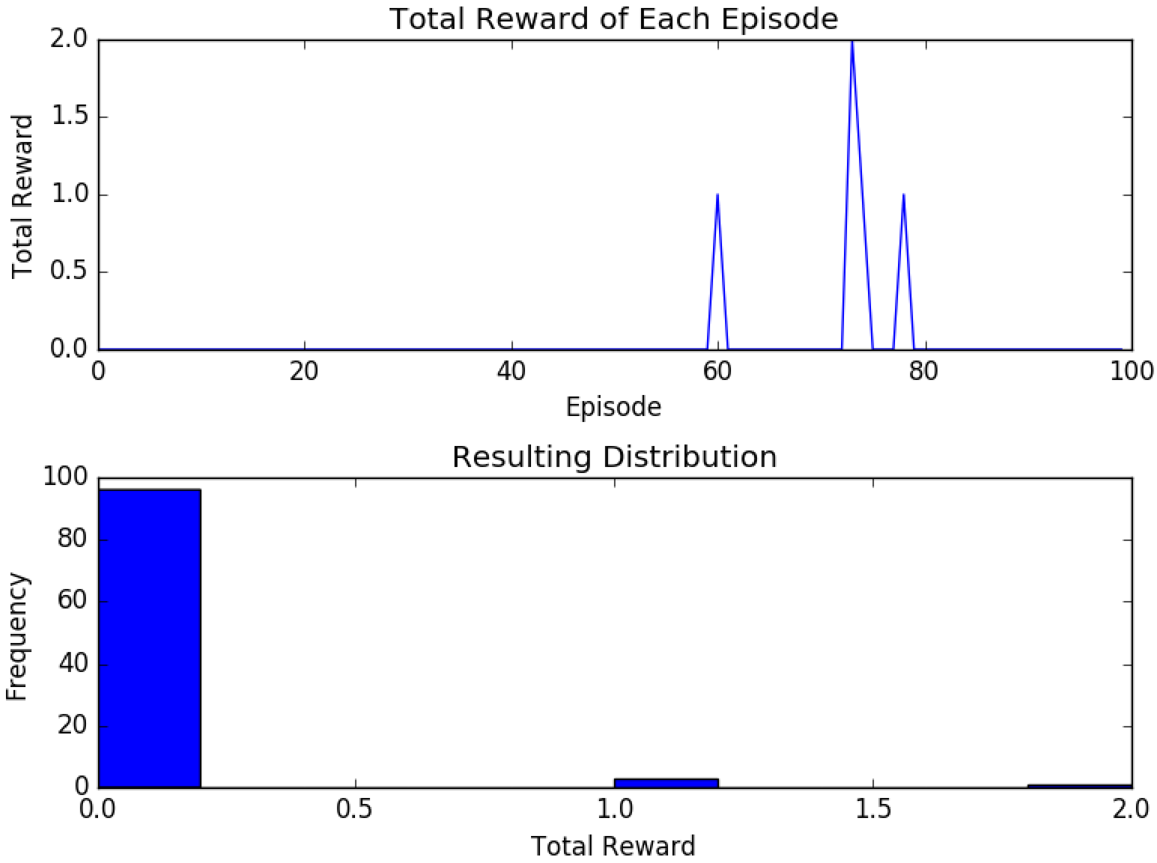


Figure 3: The total reward achieved by random agent.

According to Figure 3, it shows the total reward obtained for each episode, as well as the resulting distribution after running the agent for 100 episodes. The mean and the variance of the total reward obtained per episode is shown in Table 1.

Table 1: Mean and Variance result of random agent(100 episodes)

Items	value
Mean of the total reward	0.04
Variance of the total reward	0.012

3.3 Q-learning Agent

Comparing Figure 3 with Figure 4, a great improvement can be seen. At the first few episodes, the Q-learning agent is trying different actions and tries to learn from the failure. But later, it converges at 52 total reward which shows that agent learns how to drive. In order to show the whole trend of learning, it has 1000 episodes running. But for the result more competitive, the mean and the variance of the total reward is calculated from first 100 episodes. From the Table 1 and 2, the mean of the total reward increases from 0.04 to 31.04, which also shows the great improvement made by Q-learning agent.

Table 2: Mean and Variance result of Q-learning agent(100 episodes)

Items	value
Mean of the total reward	31.04
Variance of the total reward	136.39

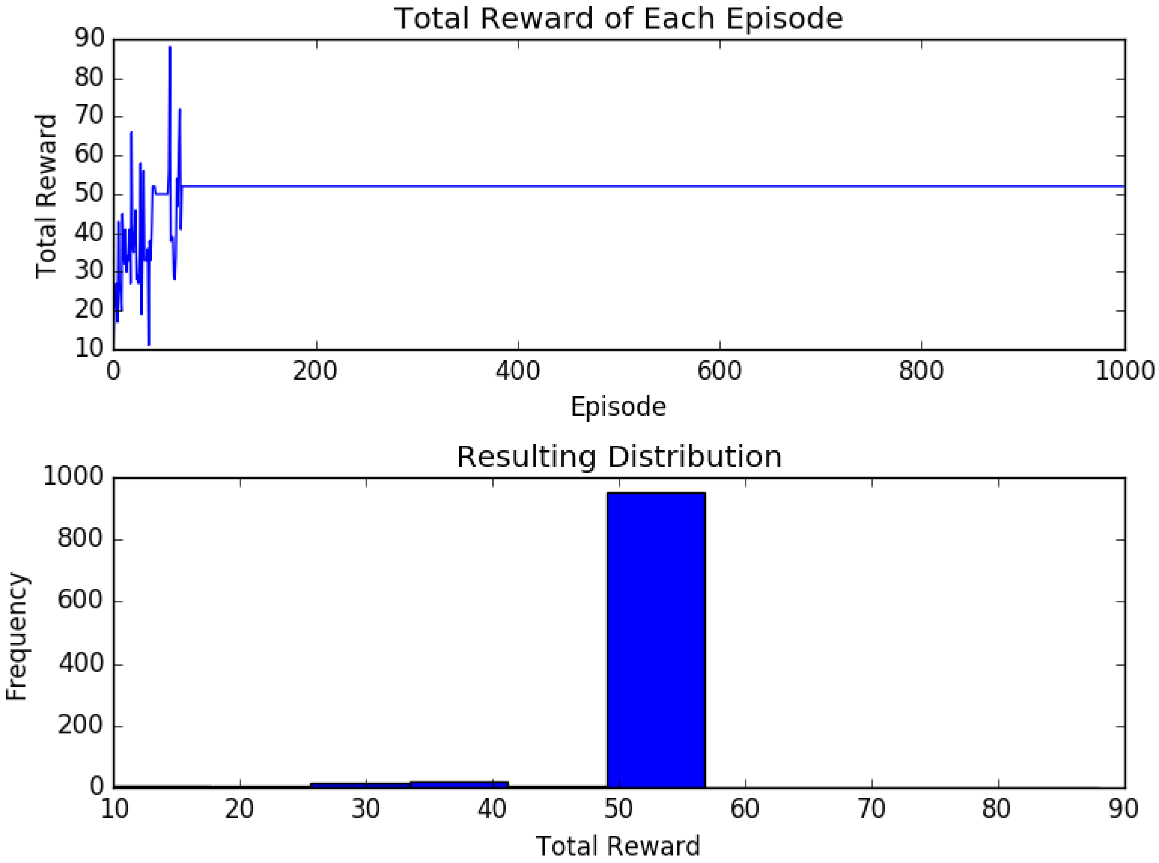


Figure 4: The total reward achieved by Q-learning agent.

3.4 Time horizon

Figure 5 shows the performance of Q-learning low-level agent, which only cares about immediate neighbourhood. The dimensionality of the state space is only 64. However, Figure 4 shows the performance of Q-learning advanced agent, which considers all components in the left, right and ahead of the agent(3 columns). The dimensionality of the state space is 10^4.So comparing Figure 4 and 5, the low-level agent converges at only 34 total reward and it converges quicker than advanced agent. Besides, from the Table 2 and 3, the mean of total reward of the agent increases from 25.69 to 31.04. In conclusion, with the increase of state space, the performance of the agent is getting better and better.

Table 3: Mean and Variance result of Q-learning agent(low-level)

Items	value
Mean of the total reward	25.69
Variance of the total reward	148.99

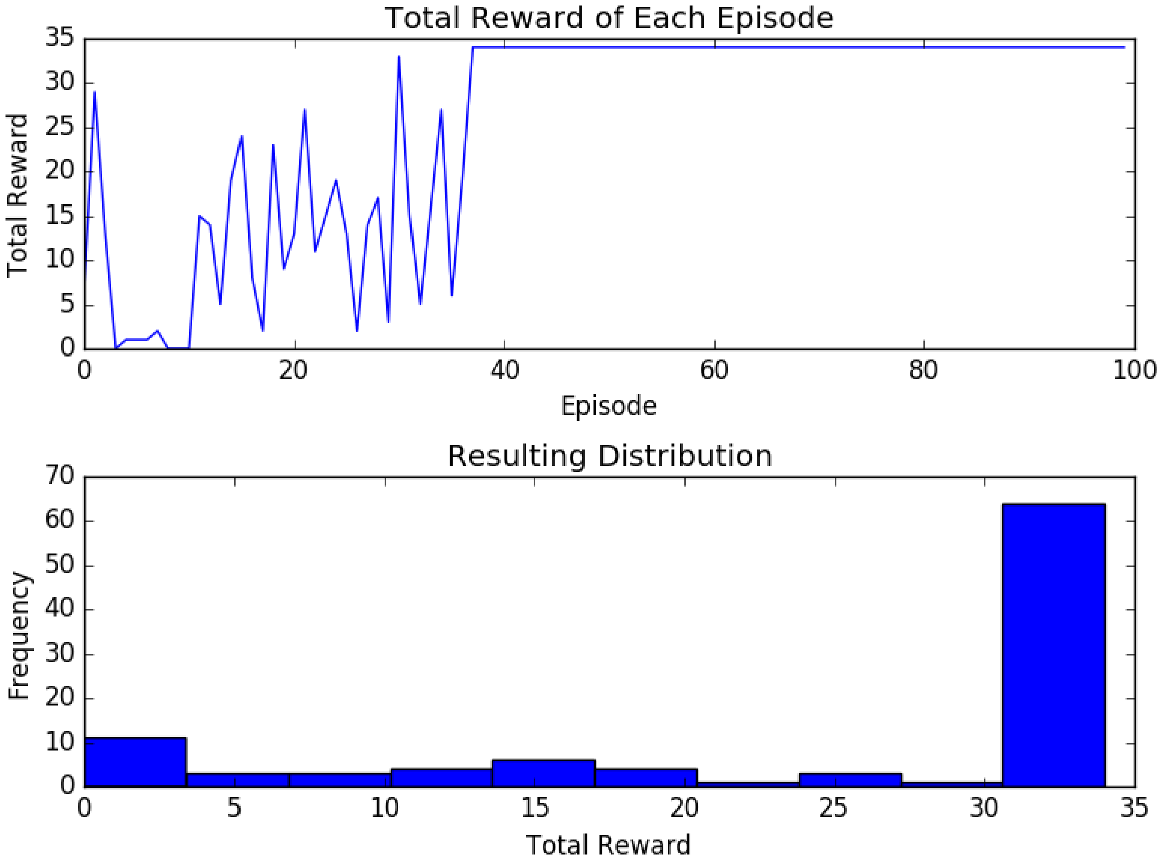


Figure 5: The total reward achieved by Q-learning agent(low-level).

3.5 Additional features

In my opinion, the speed of the agent can be considered as the feature. Considering current position information and speed, the agent can make more wise choice. So the same as before, the continuous state space dynamics model should be turned into the discretized MDP. The state can be expressed by the tuple(a,l,r,f,s), where a,l,r,f represent the position of the agent, the left component, the right component, the components in front of the agent, and s represents the speed of the agent in this situation.