ECE 276C Assignment 2 Report: Tabular Methods

Jingpei Lu
Jacob School of Engineering
University of California, San Diego

Date: October 23, 2019

1 Model Based Methods

1.1 The Environment

The FrozenLake environment is a 4 by 4 grid-world, with states numbered as follows:

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

and each state is either a frozen surface F or a hole H. The start point S is at state 0 and the goal G is at state 15. Thus, the state space is $\mathcal{S} = \{0, 1, 2, ..., 14, 15\}$.

The agent has 4 potential actions: LEFT = 0, DOWN = 1, RIGHT = 2, UP = 3. So the action space is $\mathcal{A} = \{0, 1, 2, 3\}$. The reward is 0 for every step taken, 0 for falling in the hole, 1 for reaching the final goal. Given a state and an action, there is a probability that you land in a different state from the one you intent to move, so the environment is stochastic.

1.2 The Value Function

Given the state $s \in \mathcal{S}$ and policy π, the state value function $V^\pi(s)$ can be described as:

$$V^\pi(s) = \mathbb{E}_\pi[G_t|s_t = s] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t = s] \tag{1}$$

where G_t is future cumulative discounted reward, r is the reward, and $\gamma \in [0, 1]$ is the discount factor. If we pull out the first reward from the sum, we can rewrite it as:

$$V^\pi(s) = \mathbb{E}_\pi[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2}|s_t = s] \tag{2}$$

The expectation can be computed by summing over all possible actions and all possible returned states. Therefore, the expectation can be written as:

$$\mathbb{E}_\pi[r_{t+1}|s_t = s] = \sum_a \pi(a|s) \sum_{s'} p(s'|s,a) r(s,a,s') \tag{3}$$

$$\mathbb{E}_\pi[\gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2}|s_t = s] = \sum_a \pi(a|s) \sum_{s'} p(s'|s,a) \gamma \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2}|s_{t+1} = s'] \tag{4}$$

where $a \in \mathcal{A}$, $s, s' \in \mathcal{S}$, $p(s'|s,a)$ is state transition probability, $r(s,a,s')$ is the reward function, and $\pi(s,a)$ is the policy. By expanding the equation (2) using (3) and (4), we get

$$V^\pi(s) = \sum_a \pi(a|s) \sum_{s'} p(s'|s,a)[r(s,a,s') + \gamma \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2}|s_{t+1} = s']] \tag{5}$$

Since we have deterministic policy, we always choose the optimal action a to take in each state, such that $\pi(a|s) = 1$. And by observing the equation (1), the value function then can be written as:

$$V^\pi(s) = \sum_{s'} p(s'|s,a)[r(s,a,s') + \gamma V^\pi(s')] \tag{6}$$

1.3 Test Policy

For experiment, I generated 100 random policies, and their mean successful rate is 0.0133. The performance of each random policy shows on the Figure 1.

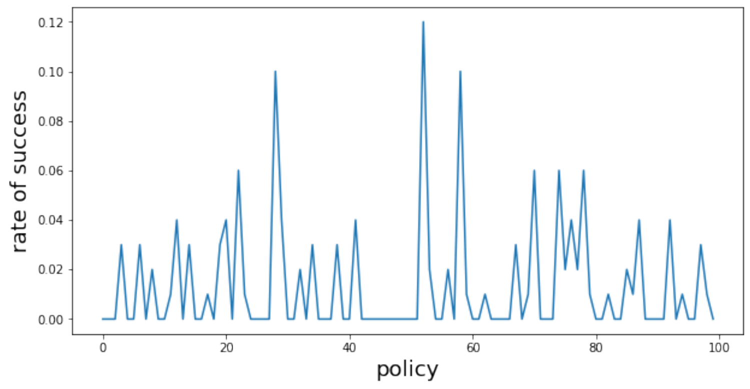

Figure 1: The performance of each random policy

1.4 Learn Model

After running 100000 episodes, I fond that the transition probability model $T(s,a,s')$ and the reward $R(s,a,s')$ always converge to some number.

1.5 Policy Iteration

Figure 2 shows the average rate of success of the learned policy at every policy iteration.

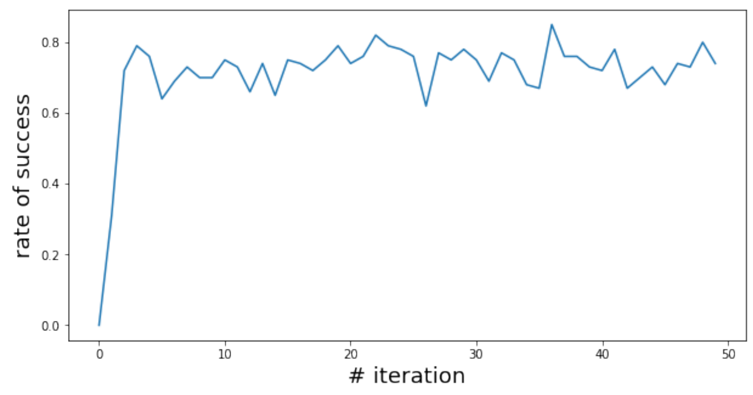

Figure 2: Average rate of success over numbers of iterations.

1.6 Value Iteration

Figure 3 shows the average rate of success of the learned policy at every value iteration.

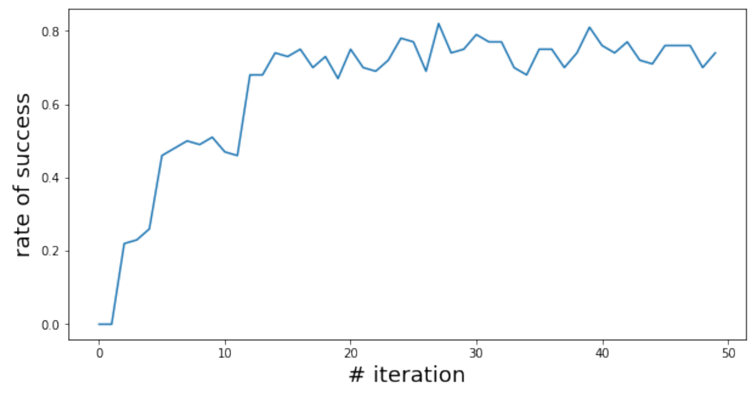

Figure 3: Average rate of success over numbers of iterations.

2 Model Free Methods

2.1 Q-learning

The figures below are showing the success rate of the learned policy over 5000 episodes using Q-learning. Figure 4 shows the performance with different learning rates with fixed discount factor. From my observation, as the learning rate increasing, the variance of the success rate is increasing. With the large learning rate, the learned Q value is biasing towards the demonstration, which might not be the true Q value. Figure 5 shows the performance of different discount factors with fixed learning rate. As the discount factor increasing, the success rate converges faster, because the the algorithm will weight more on the long term reward, which in this case is to arrive the goal.

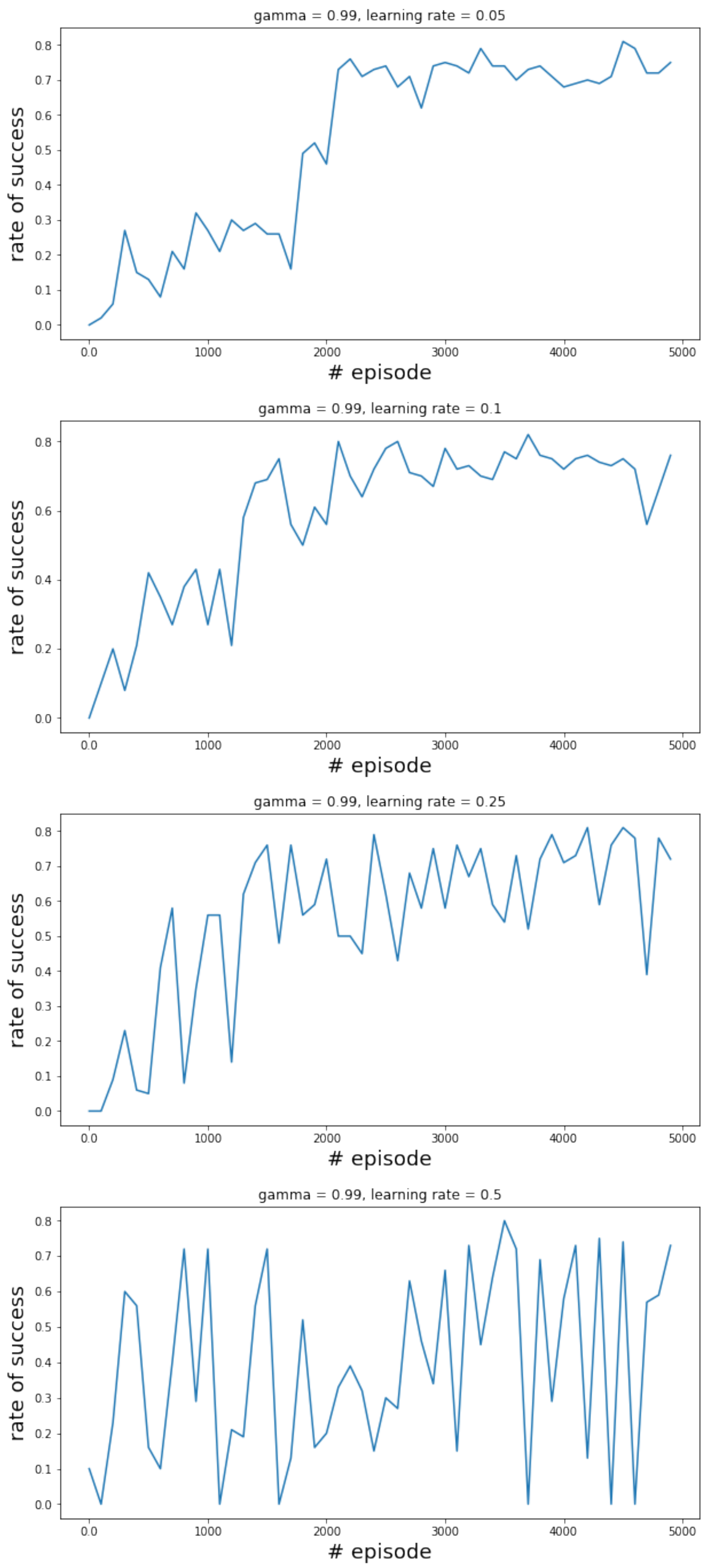

Figure 4: Average rate of success over numbers of episodes. From top to bottom, the learning rates α are 0.05, 0.1, 0.25, 0.5 correspondingly, and the discount factor $\gamma = 0.99$ is fixed.

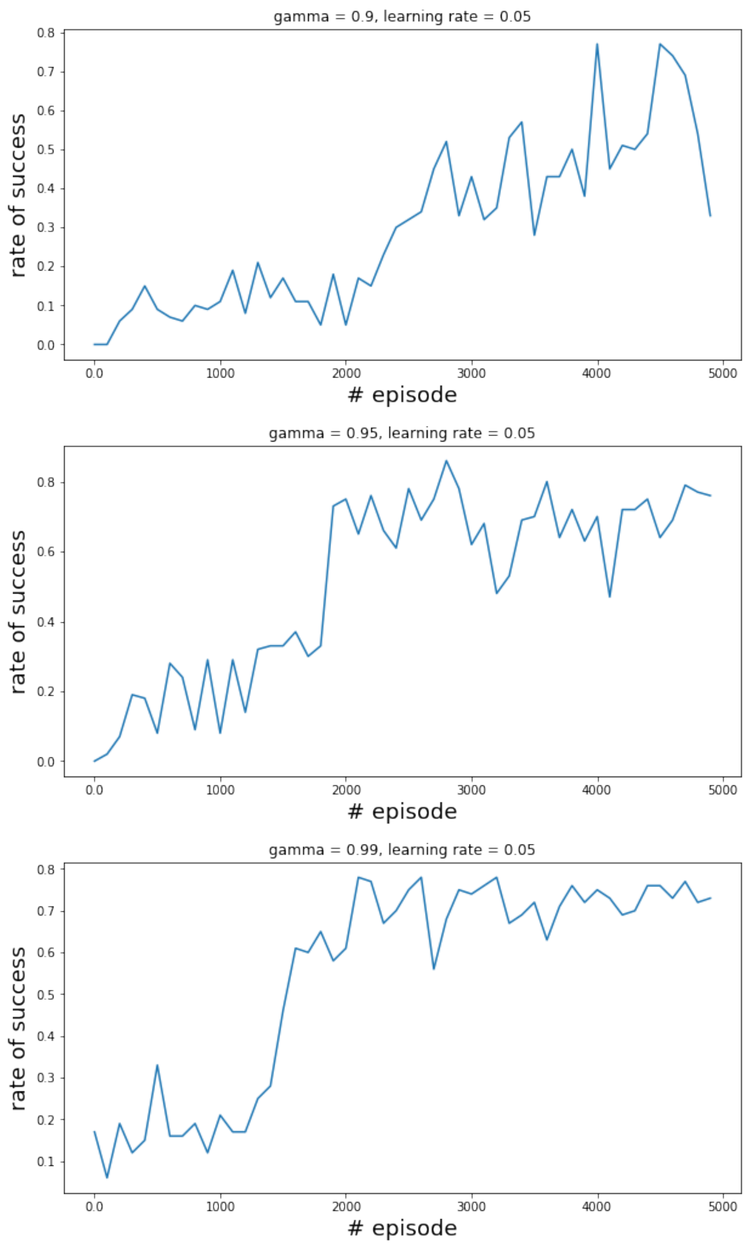

Figure 5: Average rate of success over numbers of episodes. From top to bottom, the discount factors γ are 0.9, 0.95, 0.99 correspondingly, and the learning rates $\alpha = 0.05$ is fixed.

2.2 Sigmoid Exploration

In this section, I tried to decrease the exploration rate in a non-linear way, which is applying the sigmoid function on normalized episode number. The implementation of the exploration strategy is described below.

Algorithm 1 Sigmoid Exploration

```
Input: Current state $s$, current episode number $n$, total episode number $N$
Output: Action $a$
$n \leftarrow \frac{n}{N} * 20 - 10$          ▷ normalize the episode number to $[-10, 10]$
$\epsilon \leftarrow \frac{1}{1+e^{-n}}$          ▷ apply sigmoid function on $n$
Generate $x$ from uniform distribution $\mathcal{U}(0, 1)$
if $x > \epsilon$ then
    Generate a random action $a$
else
    Take action $a$ =argmax$_a$Q(s,a)
end
```

By using this exploration strategy with learning rate $\alpha = 0.06$ and discount factor $\gamma = 0.99$, the training results of the Q-Learning is shown on the figure 6 below.

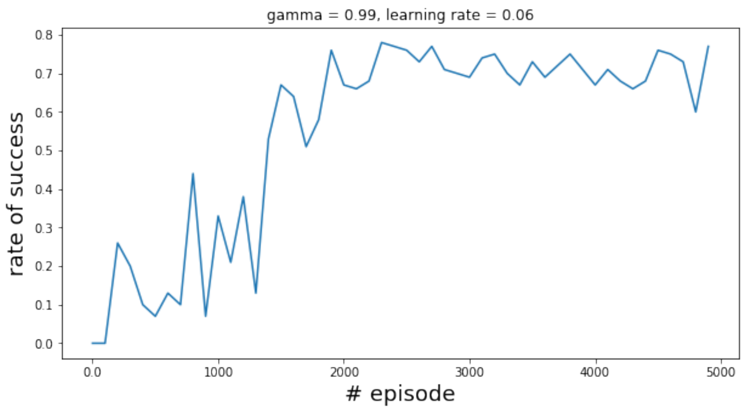

Figure 6: Average rate of success over number of episodes.