EL2805 LAB1 Report

Wei Wang, Yuxia Wang

Problem 1. The Maze and the Random Minotaur

(a) MDP formulation

State space S: S = { (x, y, a, b), where (x, y) is not an obstacle}

We consider the player and the minotaur together in the maze, (x,y) is the position of the player and (a,b) is the position of the minotaur.

Action space A : A = {stay ,left, right, up, down}

We allow the player to choose to either move left, right, down, up or not move at all (stay).

Transition probabilities P

When Minotaur and the person are in the same position, then the person will be eaten, whenever given which action, the state will not change any more.

When the person reaches the point of (6,5), while Minotaur is not there, existing is successful, the state will not change either.

While stepping, when there are N neighbour positions can be reached, the probability for one possible move will be 1/N.

p(s'=s |s{x=a, y=b}, a=·) = 1

p(s'=s |s{(x,y)=(6,5), (a,b)!=(6,5)}, a=·) = 1

p(s'=s_next |s=s_curr, a=a_curr) =1/N

Rewards R

The objective is to maximize the probability of exiting the maze before time T, so we define:

r(s= s{(x,y)=(6,5), (a,b)!=(6,5)}, a=·) = 0

r(s = s{x=a, y=b}, a=·) = -100

r(s'=s_next |s=s_curr, a=a_curr)=-1

r(s'= wall or obstacle| s=s_curr, a=go for impossible) = -100

(b) Solution for the problem

Given T= 20, one possible path for the person and Minotaur is shown as Figure 1, the blue one is for the person, and the red one is for Minotaur. This is from the condition where Minotaur can not stand still, and the assumption is that Minotaur can walk in the wall. The path for Minotaur will change each time due to the randomness.

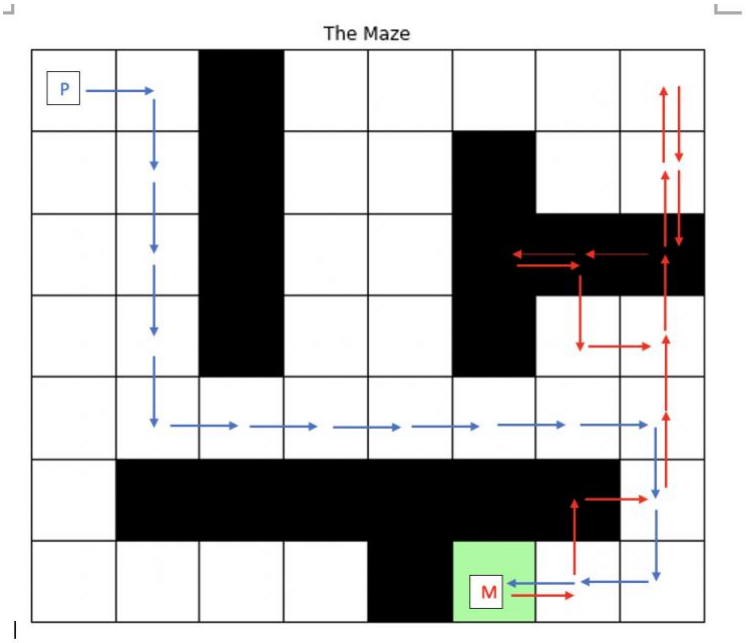

Figure 1. one possible path for person and Minotaur

Considering the two situations of Minotaur can stand still and can not stand still, as shown in Figure 2, when Minotaur can stand still, it will be harder for the person to exit alive, therefore, the maximum probability will increase from 0 to T = 15, but it will take more steps to reach 1 (after T=20).

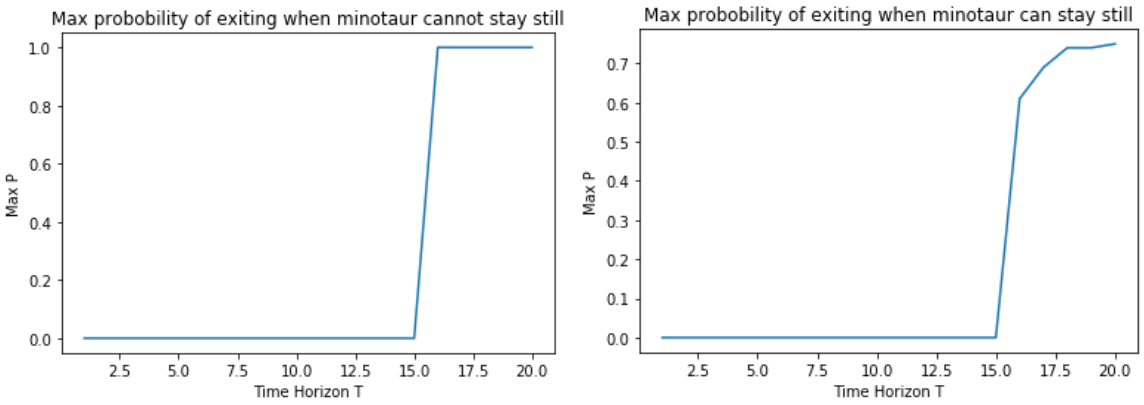

Figure 2. the maximum probability of existing for both cases

(c) Solution for new situation

Suppose life is geometrically distributed with mean 30, then it is an MDP with discounted infinite time horizon, we solve this problem by using the method of value iteration. From the geometry distribution of the life, we can get:

$$E[T] = 1/(1-\lambda) = 30$$

λ= 29/30

By setting $\xi = 0.0001$, after 10000 iterations, the probability of getting out alive is 1.

Problem 2: Robbing Banks

(a) Formulate the problem as an MDP

Here we formulate this problem in the same way as in Problem 1.

States space S = { (x, y, a, b) } , where (x,y) is the position of robber, and (a,b) is the position of the policeman.

Action space A = {stay, left, right, up, down}, we suppose the robber can stay still, and the police can not stay still too.

Transition probabilities P

When the robber and the policeman are in the same position, the robber will be caught, whenever given which action, the state will not change any more.

When the robber reaches any one of the four banks, while the policeman is not there, robbing is successful, the state will not change either.

While stepping, when there are N neighbour positions can be reached, the probability for one possible move will be 1/N.

We set bank = {(0,0), (2,0), (0,5), (2,5)}

p(s'=s |s{x=a, y=b}, a=·) = 1

p(s'=s |s{(x,y)=any of bank, (a,b)!=(x,y)}, a=·) = 1

p(s'=s_next |s=s_curr, a=a_curr) =1/N

Rewards R

The objective is to maximize the average discounted reward, so we define:

r(s= s{(x,y)= any of bank, (a,b)!= (x,y)}, a=·) = 10

r(s = s{x=a, y=b}, a=·) = -50

r(s'=s_next |s=s_curr, a=a_curr) = 0

(b) Solution and the valuation function

As shown in Figure 3, we set two intervals for lambda, both cases show that the initial value increases with the increasing lambda value. Because we want to maximize the average discounted reward, a bigger lambda means consider more on long-term reward. This shows that the long-term reward will increase when lambda approaches to 1.

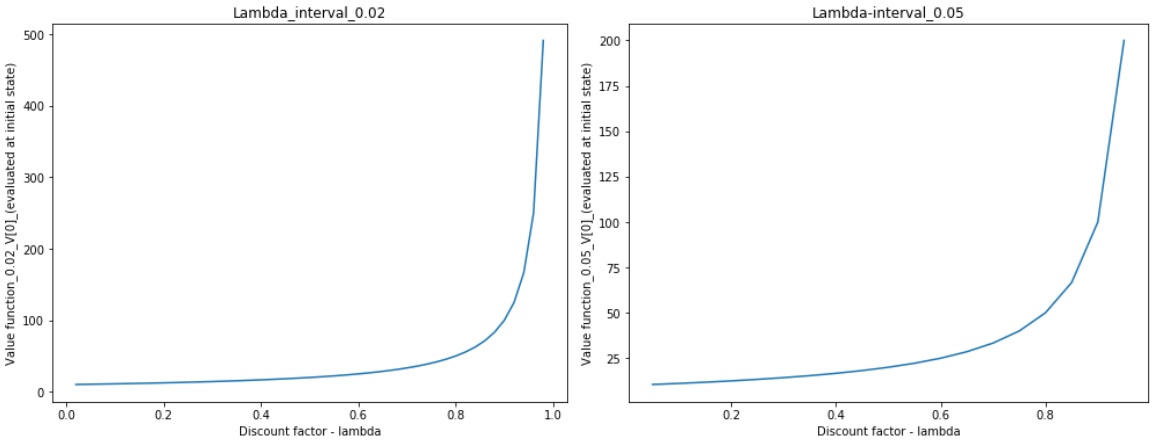

Figure 3 the valuation function changes with lambda

Problem 3. Bank Robbing (Reloaded)

(a) Q-learning

As shown in Figure 4, the value function increases over time and the Q-learning algorithm converges well given 10000000 interactions.

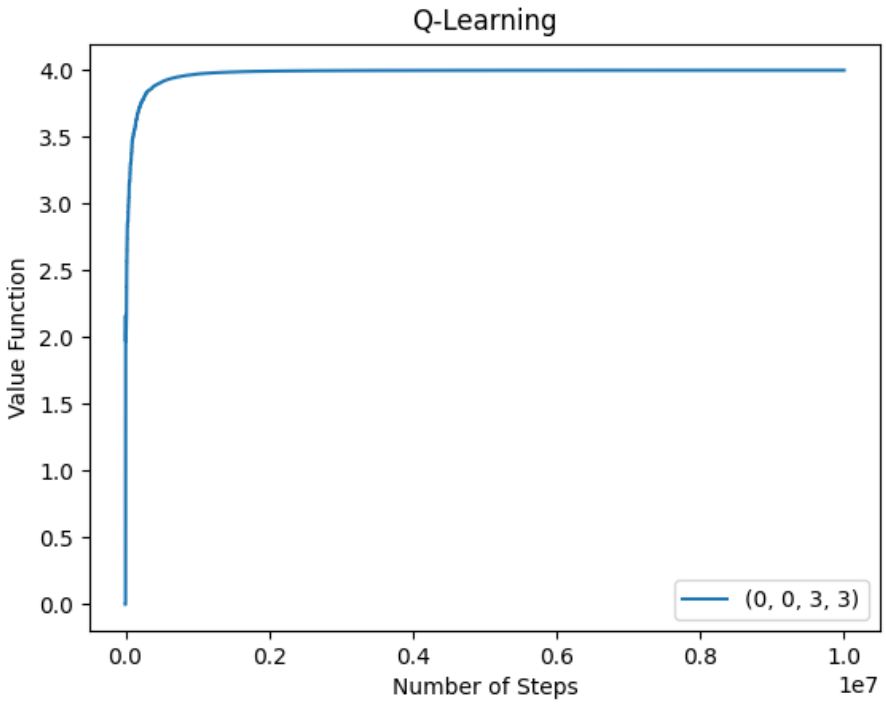

Figure 4 the value function over time

(b) SARSA

As shown in Figure 5, when Epsilon equals to 0.1, the value function changes dynamically first and then converges to a certain range. And this range of value is smaller than Q-learning, which is the optimal policy.

Considering different Epsilon values, Figure 6 gives the learning process of value functions. Similarly, Figure 5, all conditions converge to a certain range, and usually smaller Epsilon values give higher Q values, which makes sense because the Epsilon-greedy policy will explore more by (1-epsilon) terms.

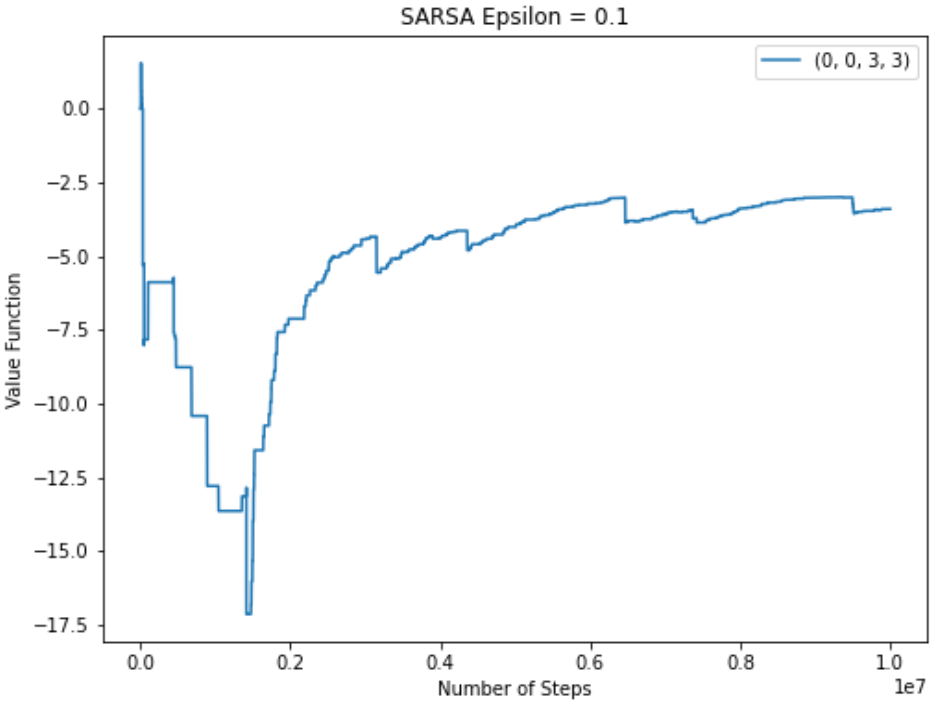

Figure 5 value function from SARSA over time

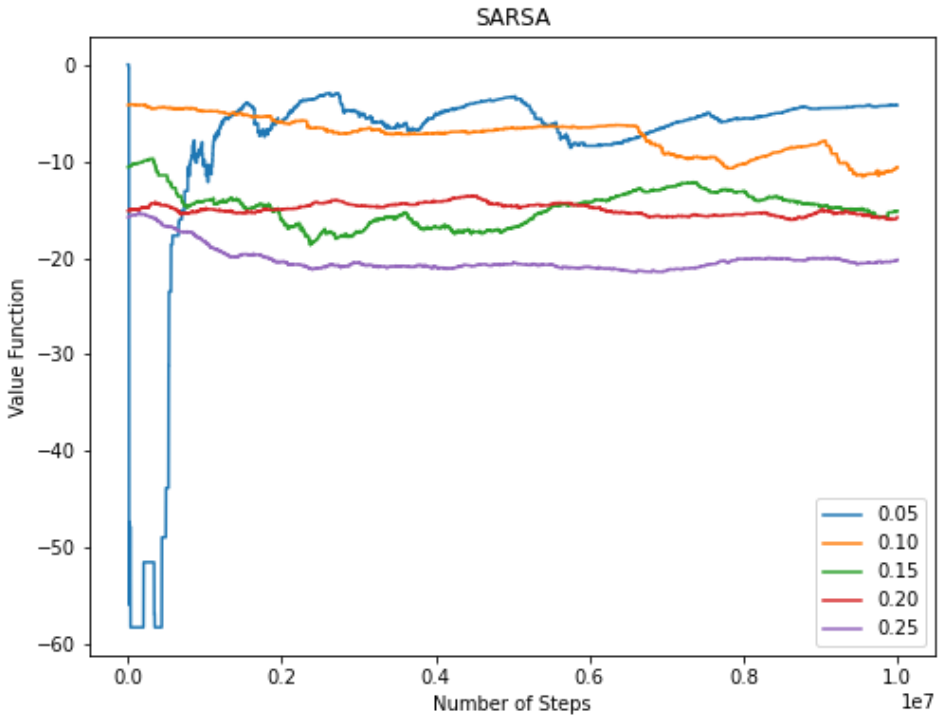

Figure 6 the value functions from SARSA based on different Epsilon values.