EL2805 Lab1 Report

Jiaying Yang
jiayingy@kth.se

Yiping Xie
yipingx@kth.se

1 Problem 1: The Maze and the Random Minotaur

1.1 Formulate the problem as a MDP

State Space:

$$S = \{S_0, S_1, \ldots, S_{899}\} \cup \{Dead\} \cup \{Win\}, \quad S_i = (P_x, P_y, M_x, M_y)$$
$$0 \leq P_x \leq 5,\ 0 \leq P_y \leq 4;\ \ 0 \leq M_x \leq 5,\ 0 \leq M_y \leq 4;$$

where $S_i = (P_x, P_y, M_x, M_y)$, P_x is the person's position along the x-axis, P_y is the person's position along the y-axis , M_x is the minotaur's position along the x-axis, M_y is the minotaur's position along the y-axis. The "Dead" state denotes the next state of person being at the same position of the minotaur, and the "Win" state denotes the next state after the person at the exit.

Actions:
The person can choose to move up(0), down(1), left(2), right(3) or stay(4) at every time.

Time-horizon and Objective Function: The time-horizon is T. The finite-horizon total reward function is:

$$\mathbb{E}\{\sum_{t=0}^{T-1} r_t(S_t, a_t) + r_T(S_T)\}$$

Rewards:
Here I model the reward as this: Every time the player meets the minotaur, he collects reward -1, every time he is at the exit while the minotaur is not, he collects a reward 1, and every time he wants to go through the wall in the maze, he collects a reward -100 to ensure he will not choose that action.
Terminal rewards:

$$r_T(S = Dead) = 0$$
$$r_T(S = Win) = 0$$

Rewards:

$$r_t(S = Dead, a = \cdot) = 0$$
$$r_t(S = Win, a = \cdot) = 0$$
$$r_t(S = S_{\{P_x = M_x, P_y = M_y\}}, a = \cdot) = -1$$
$$r_t(S = S_{\{P_x = B_x, P_y = B_y\}}, a = \cdot) = 1$$
$$r_t(S = S_w, a = A_w) = -100$$
$$r_t(S = S_{no}, a = A_{no}) = 0$$

where S_w is the state when the player is next to wall or edge in any direction, where A_w is the action which will hit wall or edge at state S_w. S_{no} denotes nothing particularly happens at this state, and A_{no} denotes that nothing particularly will happen if taking this action at this state.

Transition Probabilities:
The non-zero transitions $P_t(S'|S,a)$ are:

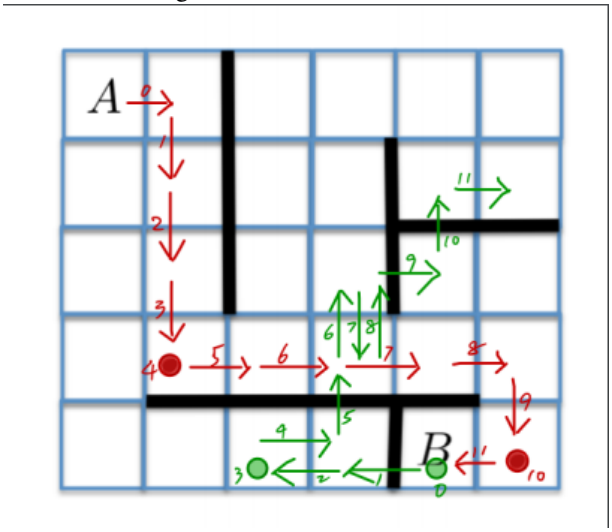
$$P_t(S' = Dead|S = Dead, a = \cdot) = 1$$
$$P_t(S' = Win|S = Win, a = \cdot) = 1$$
$$P_t(S' = Dead|S = S_{\{P_x=M_x, P_y=M_y\}}, a = \cdot) = 1$$
$$P_t(S' = Win|S = S_{\{P_x=B_x, P_y=B_y\}}, a = \cdot) = 1$$
$$P_t(S' = S'_N|S = S_N, a = a_N) = \frac{1}{N}$$

where N denotes the number of minotaur's possible actions without going out of the maze at state S_N. If the minotaur is at the corner $N = 3$, if the minotaur is next to the edge except for the corner $N = 4$ and otherwise $N = 5$ (in this case we assume the minotaur can stand still). Here we did not include the restriction of wall for the player, since we have already set the action of trying to go through it with reward -100, which will guarantee the player not hit the wall.

1.2 Solve the Problem for T=15

Here is a simulated game showed in figure 1:
The arrows illustrate the moves of the player/minotaur along the time, the dots denotes stay, the red

Figure 1: A Simulated result



one denotes the player's trajectory and the green one denotes the minotaur's trajectory. As we can see, at $t = 11$ the player is at the exit, which indicates the win.
The figure 2 of maximal probability of exiting the maze as a function of T is as follows.
The result below 2 is the case where minotaur is not allowed to stand still. As we can see, before $T = 11$, since there is not enough steps to get to the exit, the probability of exiting within 11 is zero. If $T \geq 12$, the probability of getting out is 1. There will be a difference if minotaur is allowed to stand still, since the transition probabilities will be different therefore the optimal policy will be different as well.
When minotaur is not allowed to stand still, he has to move in some direction every step, which gives the player information to decide which action is the best. This action usually would try to be far away from minotaur but close to the exit at the same time. And the case like moving up and then moving down usually will not happen, which is a waste of steps. However, if minotaur stands still at curtain step, the player would be more cautious about next move, sometimes choosing standing still unnecessarily, which is a waste of steps.

The figure 3 shows the case that minotaur is allowed to stand still. When $T \geq 11$ and as T grows, the probability of exiting is larger and larger until becoming nearly 1.

Figure 2: Minotaur not allowed to stand still

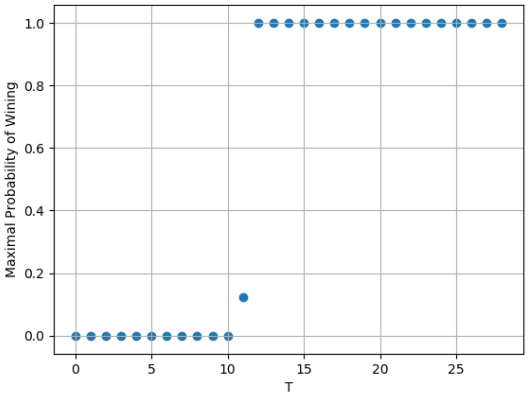
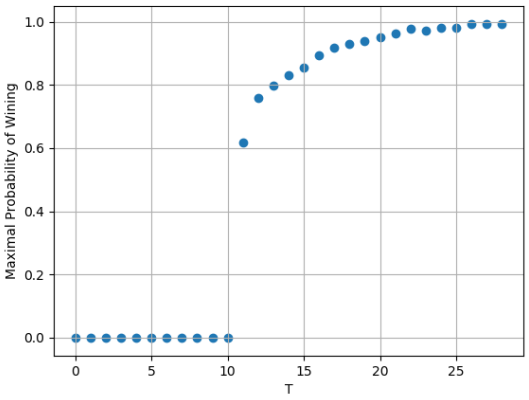


Figure 3: Minotaur allowed to stand still



1.3 Infinite MDP with Discounted Expected Reward

Assume the player's life is geometrically distributed with mean 30, we can model the problem as an infinite MDP with $\mathbb{E}[T] = \frac{1}{1-\lambda} = 30$, where λ is the discounted factor. Then the objective function becomes:

$$\max_{\pi} \lim_{T \to \infty} \mathbb{E}[\sum_{t=1}^{T} r_t(S_t^{\pi}, a_t^{\pi})]$$

The Transition Probabilities and the Reward Functions remain the same except for there is no terminal reward in this case.
Since we can not initialize Backward Deduction lacking of the final state, we use Value Iteration (VI) algorithm with precision $\epsilon = 0.01$. After the value converge, we run the simulation $1e4$ times to estimate the probability of getting out alive using this policy, where we get the probability 1. The result illustrates the optimal policy is indeed good enough to guarantee getting out alive.

2 Problem 3: Bank Robbing (Reloaded)

State Space:

$$S = \{S_0, S_1, \ldots, S_{255}\}, \quad S_i = (R_x, R_y, P_x, P_y)$$
$$0 \leq R_x \leq 3,\ 0 \leq R_y \leq 3;\ \ 0 \leq P_x \leq 3,\ 0 \leq P_y \leq 3;$$

where $S_i = (R_x, R_y, P_x, P_y)$, R_x is the bank robber's position along the x-axis, R_y is the bank robber's position along the y-axis , P_x is the police's position along the x-axis, P_y is the police's position along the y-axis.

Actions:
The bank robber can choose to move up(0), down(1), left(2), right(3) or stay(4) at every step.

Time-horizon and Objective Function: The time-horizon is T. The finite-horizon total reward function is:

$$\mathbb{E}\{\sum_{t=1}^{T} \lambda^{t-1} r(S_t^\pi, a_t^\pi) | s_1^\pi = s\}$$

For a given discount factor $\lambda \in [0, 1)$, from the data, find a policy $\pi^* \in MD$ maximizing (over all possible policies).

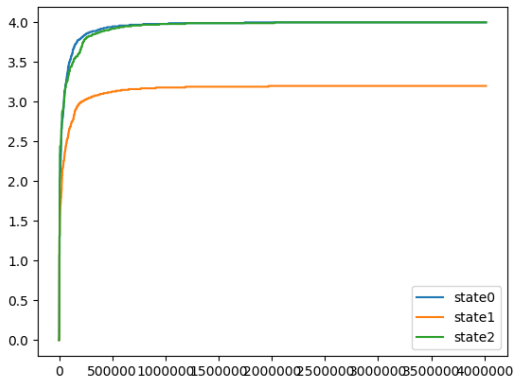
Rewards:
Here I model the reward as this: Every time the bank robber meets the police, he collects reward -10, every time he is at the point B (the bank's position) while the police is not, he collects a reward 1.

$$r(S = S_P, a = \cdot) = -10$$
$$r(S = S_B, a = \cdot) = 1$$

2.1 Q-Learning Algorithm

Algorithm:
Initialize $Q(s, a)$ as a zero matrix
Initialize s
Repeat (for each step) :
Choose a from s randomly
Take action a, observe r, s'
$Q(s, a) \leftarrow Q(s, a) + \alpha\,[r + \gamma max_a Q(s', a) - Q(s, a)]$
$s \leftarrow s'$;

Figure 4: Plot of value function over time

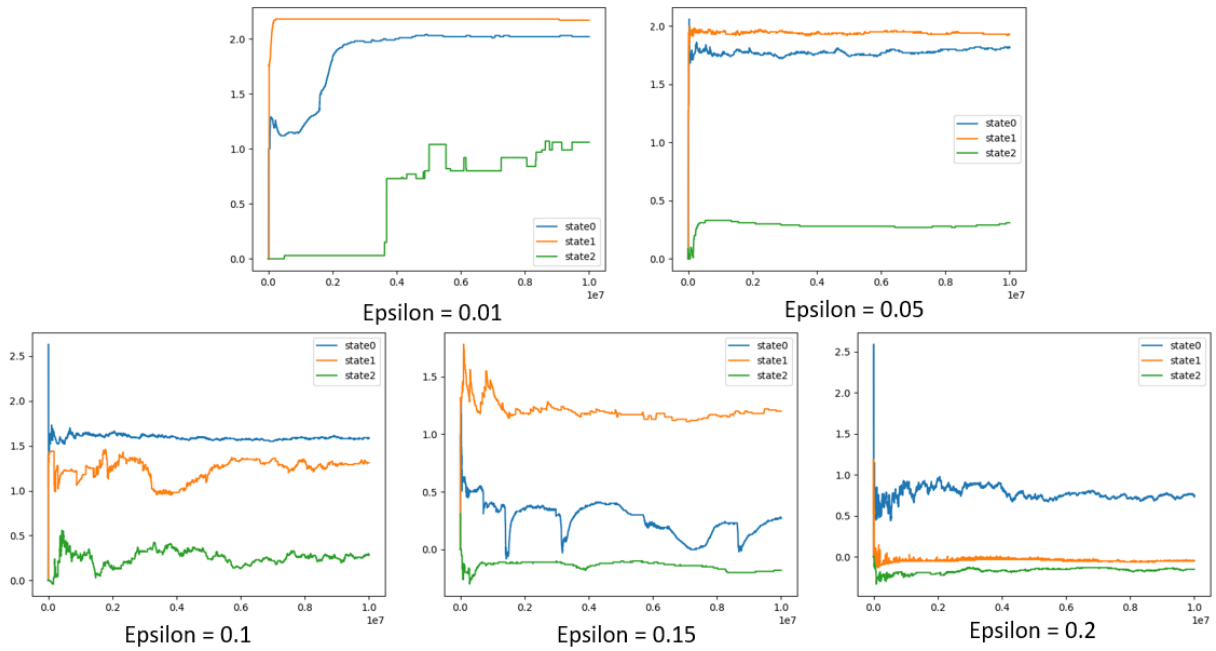


Here we randomly choose three states to plot its value function. We run the Q-learning process for $1e7$ steps, and plot the result of value function according to it. We can find out the value function converges after around $1e7$ steps.

2.2 SARSA Algorithm

Algorithm:
Initialize $Q(s, a)$ as a zero matrix
Initialize s
Choose a from s according to $\epsilon - greedy$
Repeat (for each step) :
Take action a, observe r, s' Choose a' from s' according to $\epsilon - greedy$
$Q(s, a) \leftarrow Q(s, a) + \alpha\left[r + \gamma Q(s', a') - Q(s, a)\right]$
$s \leftarrow s'; a \leftarrow a';$
As we can see in the figure above, we run $1e7$ steps using SARSA algorithm for different ϵ. Since

Figure 5: Plot of value function over time



SARSA algorithm is not as greedy with respect to rewards as Q-learning, the value function over time for SARSA algorithm will not be as good as Q-learning, at least for some states. But with reasonable ϵ, for example, ϵ =0.05, the value function is going to converge after certain steps.