Reinforcement Learning Lab01

Chieh-Ju Wu, Bix Eriksson

November 29, 2020

Contents

1 Problem 1: The Maze and the Random Minotaur

1.1 Formulate the problem as an MDP

1.1.1 State Space

The state space is encoded as two tuples, the (x, y) coordinates of the player p and the Minotaur m with the player's origin being in the top left (0, 0) and Minotaur's origin at (6, 5). The player can only be in blank cells and can not walk through walls whereas Minotaur is available to do so. Both of them can not walk diagonally nor can them be out of boundary of the gird world. In our map, $0 \leq$ x ≤ 7 and $0 \leq$ y ≤ 6. Thus, we can derive this system into a MDP as described below.

$$s_t = \{(r, p) \mid p = (P_x, P_y), m = (M_x, M_y)\} \in S \cup \{Dead\} \cup \{Win\}$$

$$S = [0, L]X[0, W]X[0, L]X[0, W]$$

where L, W be the length and the width of the maze

1.1.2 Action Space

The valid actions for the player in particular state s is a subset A_s of A which is defined as:

$$A = \{UP(0),\ DOWN(1),\ LEFT(2),\ RIGHT(3),\ STAY(4)\}$$

We assume that Minotaur can not stay in the cell. If there is a wall on the right or the boundary is on the right, then RIGHT is an invalid action. Thus, we can deduct that actions depends on the current state. Furthermore, we can say that the valid action exists if and only if the action results into another state which is a member of the State Space S.

1.1.3 Time Horizon and Objective Function:

The time horizon is T. The finite horizon total reward function is

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

1.1.4 Rewards:

The player receives +1 reward upon entering the goal state. The player receives -100 reward each time hitting the wall or going out of edge or getting eaten the Minotaur. Minotaur can only eat you if and only if the player and it are located at the same position at the same time. The player receives +0 reward each time he makes a valid action.

Non terminal rewards:

$$\begin{aligned}
r_t(S = S_{\{P_x=G_x,\ P_y=G_y\}},\ a = \cdot) &= +1 \\
r_t(S = S_{wall},\ a = A_{wall}) &= -100 \\
r_t(S = S_{edge},\ a = A_{edge}) &= -100 \\
r_t(S = S_{\{P_x=M_x,\ P_y=M_y\}},\ a = \cdot) &= -100 \\
r_t(S = S_{normal},\ a = \cdot) &= 0 \\
r_t(S = Win,\ a = \cdot) &= 0 \\
r_t(S = Dead,\ a = \cdot) &= 0
\end{aligned}$$

Terminal rewards:

$$\begin{aligned}
r_T(S = Dead) &= 0 \\
r_T(S = Win) &= 0
\end{aligned}$$

S_{wall} : states where the player are next to a wall in either direction

A_{wall} : actions which results in hitting the wall at states S_{wall}

S_{edge} : state where the player are next to an edge in either direction

A_{edge} : actions which results in going out of edge at states S_{edge}

S_{normal} : valid states which results into nothing

$G_{x,y}$: coordinates of the goal

1.1.5 Transition Probabilities:

The state transition probabilities for the player is certain while Minotaur moves randomly. The non-zero transition probabilities $P_t(S'|S, a)$ are:

$$\begin{aligned}
P_t(s' = Dead|s = Dead,\ a = \cdot) &= 1 \\
P_t(s' = Win|s = Win,\ a = \cdot) &= 1 \\
P_t(s' = Dead|s = s_{\{P_x=M_x,\ P_y=M_y\}},\ a = \cdot) &= 1 \\
P_t(s' = Win|s = s_{\{P_x=G_x,\ P_y=G_y\}},\ a = \cdot) &= 1 \\
P_t(s' = s_{(p_{t+1},\ m_{t+1})}|s = s_{(p_t,\ m_t)},\ a = a) &= \frac{1}{A}
\end{aligned}$$

a : the valid actions to move from p_t to p_{t+1}

A : number of valid moves for Minotaur at its current state

1.2 Finite time horizon - T = 20

1.2.1 Minotaur cannot stay

Figure 1 illustrates the game played with an optimal policy for time horizon 20 when the Minotaur is not allowed for action "STAY". The green arrows with numbers denotes the move taken by the player while the purple ones represent the Minotaur's moves. As we can see at T = 15, the player reaches the exit, which indicates win.

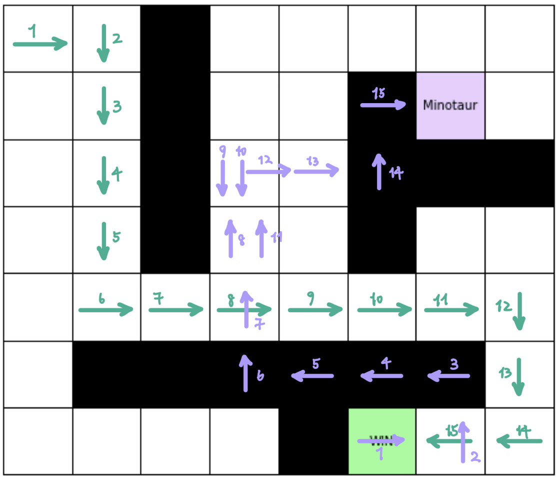

Figure 1: Policy Simulation (Minotaur cannot stay)

In figure 2, we can see that the maximal probability of escaping the maze w.r.t. time horizon T is 1.0 at T = 15 (shortest path), this means that taking the shortest path would always be safe, hence for time horizon 20 as well.

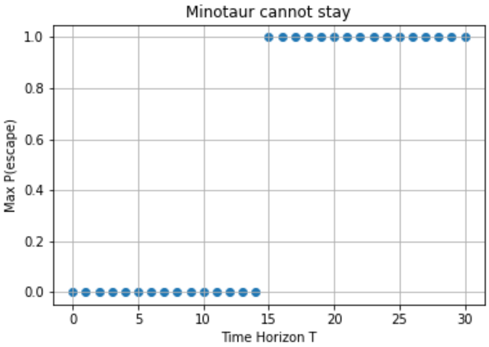

Figure 2: Maximal Probability to exit the Maze, Minotaur starts at (6, 5)

However, if we look closer at figure 1, the maximal probability 1.0 is simply due to the impossibility for the player and the Minotaur to be at the same place if Minotaur starts at (6, 5). If we changed the starting point of Minotaur to (6, 6), then we can see a drop of probability for time horizon 15, then taking the shortest path has less probability to win at T = 15, as figure 3 depicts.

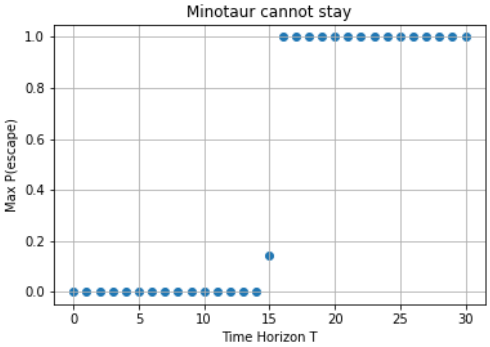

Figure 3: Maximal Probability to exit the Maze, Minotaur starts at (6, 6)

1.2.2 Minotaur can stay

Figure 4 illustrates the game played with an optimal policy for time horizon 20 when the Minotaur is allowed for the action "STAY". The green arrows with numbers denotes the move taken by the player while the purple ones represent the Minotaur's moves. As we can see at T = 19, the player reaches the exit, which indicates win.

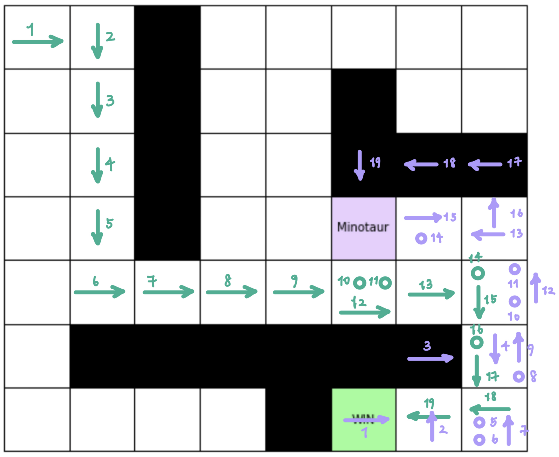

Figure 4: Policy Simulation (Minotaur can stay)

In figure 5, we can see that the maximal probability of escaping the maze w.r.t. time horizon T is much lower since it will give the player a hard time to make decisions. For instance, if the player and the Minotaur are standing side by side (Left: player; Right: Minotaur), in contrast to the previous case where the player should always proceed to the state closer to the goal, either choosing stay or proceed will pose the same threat to the player, hence, the best choice will be taking a backward step. These backward step actions will lead to longer time to escape the Maze, hence the lower probability for each time horizon. If we prolonged the time horizon, we can then see the probability of exiting the maze grows along with the increment.

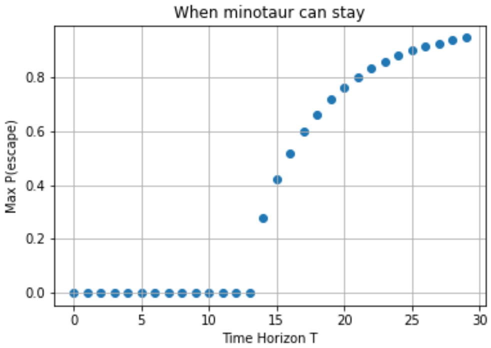

Figure 5: Maximal Probability to exit the Maze, Minotaur starts at (6, 5)

1.3 Infinite time horizon

Assume the player's life is geometrically distributed with mean 30, we can model the problem as an infinite MDP with equation 1.1, where λ is the discounted factor. Then the objective function becomes equation 1.2.

$$\mathbb{E}[T] \;=\; \frac{1}{1-\lambda} \;=\; 30 \tag{1.1}$$

$$\max_{\pi} \;=\; \lim_{T\to\infty} \mathbb{E}[\sum_{t=1}^{T} \lambda^{t-1}\; r(s_t^{\pi},\; a_t^{\pi})] \tag{1.2}$$

The transition probabilities and the reward function remain the same except that the absence of terminal states and terminal rewards. Since dynamic programming is not a possible action here, we use value iteration algorithm with precision $\epsilon = 0{,}0001$. After the value converge, we run the simulation for 10000 times to estimate the probability to escape the exist using this policy. We get probability 1.0. The result illustrates that with the given life distribution, the optimal policy is good enough to guarantee to escape the maze.

2 Problem 2: Robbing Banks

2.1 Formulate the problem as an MDP

2.1.1 State Space

The state space is encoded as two tuples, the (x, y) coordinates of the player p and the Police p with the robber's origin being in the top left (0, 0) and Police's at (1, 2). We assume that there is only one police in town, which means, if the police is not in the police station, the robber does not get caught. In other words, both the robber and the police can be in any place in the map as long as they don't go out of boundary, and the robber is only caught by the police if and only if they are in the same position at the same time.

$$s_t = \{(R, P) \mid R = (R_y, R_x), P = (P_y, P_x)\} \in S$$

$$S = [0, L]x[0, W]x[0, L]x[0, W]$$

where L, W be the length and the width of the town

2.1.2 Action Space

The valid actions for the player in particular state s is a subset A_s of A which is defined down below. Both the robber and the police cannot walk diagonally.

$$A = \{STAY(0),\ UP(1),\ RIGHT(2),\ DOWN(3),\ LEFT(4)\}$$

The problem is formulated as such that if a move will take either the robber or police out of boundaries of the map, the move will be evaluated as a STAY move, and the police/robber will be left on the same position as before the move.

2.1.3 Rewards:

The robber receives +10 reward when robbing a bank. The player receives -50 reward when being caught by the police. The police can only catch the robber if and only if they are at the same position at the same time. The robber will not get caught if he got into the police station as long as the police is not there. The player receives 0 reward each time it makes a move that neither leads to going into the bank nor getting caught.

$$r_t(S = S_{\{R_x = B_x,\ R_y = B_y\}},\ a = \cdot)\ =\ +10$$
$$r_t(S = S_{\{R_x = P_x,\ R_y = P_y\}},\ a = \cdot)\ =\ -50$$
$$r_t(S = S_{normal},\ a = \cdot)\ =\ 0$$

$B_{x,y}$: coordinates of the banks

S_{edge} : state where the player are next to an edge in either direction

A_{edge} : actions which results in going out of edge at states S_{edge}

S_{normal} : valid states which results into nothing

2.1.4 Transition Probabilities:

The state transition probabilities for the robber is certain while the police moves randomly towards the robber, and so, the probability of going to the next state is a dependent on the number of possible moves that can be done by the police. The exception is the case where the robber is caught, in which it moves to the initial state $(R_y, R_x, P_y, P_x) = (0, 0, 1, 2)$ with probability 1.

$$P_t(s' = s'_{(r_{t+1},\ p_{t+1})} | s = s_{(r_t,\ p_t)},\ a = a_N)\ =\ \frac{1}{N}$$
$$P_t(s' = initial\ state | s = caught,\ a = \cdot)\ =\ 1$$

N : number of valid actions for the police at its current state

2.2 Solution

2.2.1 Value function as a function of alpha

The value function has been evaluated for different discounting factors and can be found in figure 6. The value function has been evaluated for discount factors in the range of 0 to 0.9.

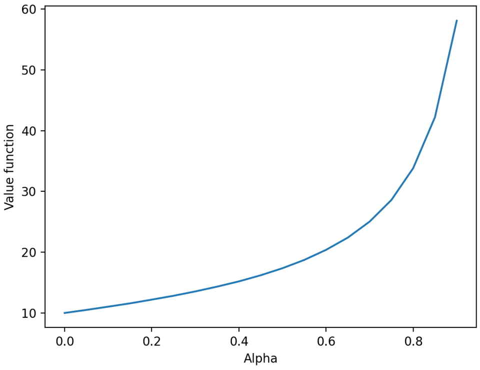

Figure 6: Value function as a function of discount factors, alpha

2.2.2 Optimal policy

The optimal policy for a variation of alpha can be seen illustrated under the link:

https://youtu.be/XJN3Wv4R4oQ

As can be seen, the robber avoids getting caught for all of the discounting factors applied. The most notable difference between the different policies has to do with the policies ability to plan ahead of time. For instance, the more myopic policy(corresponding to alpha = 0) rarely leaves the initial state side of the map(unless it is being chased off by the police) while the more far-sighted policy(corresponding to alpha = 0.9) find more value by changing side depending on the position of the police. The policies with larger discount factors has a better understanding of how far away the potential threat of being caught is, which is also reflected in the value function in figure 6.