EL2805 Reinforcement Learning
Computer Lab 1

Núria Casals
casals@kth.se
950801-T740
Robin de Groot
robindg@kth.se
981116-T091

Problem 1: The Maze and the Random Minotaur

(a) Formulate the problem as an MDP

The state representation was subdivided in the state of the player and the state of the minotaur since both the player and the minotaur can be in any of the spaces inside the maze as long as the square is not part of the wall. In our setting, the Minotaur can jump walls as long as it is a one length jump. This results in the following state space representation:

$$\mathcal{S} = \mathcal{S}_p \times \mathcal{S}_m = \{(i, j) : \text{such that the cell (i,j) is inside the maze and not a wall}\}^2$$

The action space was defined by the assignment, and we note it down as follows:

$$\mathcal{A} = \{\text{Stay}, \text{Move left}, \text{Move right}, \text{Move up}, \text{Move down}\}$$

The transition probabilities are a bit more involved as we implemented them through multiple condition checks. First, we check if the player and the minotaur are in the same position, or if the player is in the goal state (in our case position (6,5) in the maze, as defined by the assignment). In both scenarios, the player and the minotaur will stay at the same position, therefore

$$\begin{aligned}\mathbb{P}((s_p, s_m)|(s_p, s_m), \cdot) = 1, \qquad & \forall s_p = s_m \\ \mathbb{P}((s_p, s_m)|(s_p, s_m), \cdot) = 1, \qquad & \forall s_p = (6, 5)\end{aligned}$$

If neither these conditions hold, we move on to the next transition probabilities assuming that the previous once did not happen. The transition probabilities also assume that the player will not move into a wall, as we have not defined those as possible states in the maze, and we assume that the player does not try to move to a non-adjacent state, as it is not part of the action space. This restriction is part of the assignment description. Let (s_p, s_m) and $(s'_p, s'_m) \in \mathcal{S}$,

$$\begin{aligned}\mathbb{P}((s_p, s'_m)|(s_p, s_m), Stay) &= \frac{1}{\text{\# possible } s'_m} \\ \mathbb{P}((s'_p, s'_m)|(s_p, s_m), a) &= \frac{1}{\text{\# possible } s'_m} \qquad \forall a \in \mathcal{A} \setminus \{Stay\}\end{aligned}$$

Our end goal is to maximize the probability of exiting the maze, or in other words, reach position (6,5) in the maze without being eaten. Therefore we should define the reward function as follows:

$$r(s,a) = r((s_p, s_m), a) = \mathbb{1}_{\text{Player exiting the maze}} = \begin{cases} 0 & s_p \neq \text{exit cell} \\ 1 & s_p = \text{exit cell} \end{cases}$$

Where s_p is the player's position in the maze. With this MDP setting, we will maximize $\mathbb{E}[\sum_{t=0}^{T} r(s_t, a_t)] = \mathbb{E}[\mathbb{1}_{\text{Player exiting the maze}}] = \mathbb{P}(\text{Player exiting the maze})$ as we wanted.

(b) Solve the problem, and illustrate an optimal policy for $T = 20$. Plot the maximal probability of exiting the maze as a function of T. Is there a difference if the minotaur is allowed to stand still? If so, why?

You can see the illustration of the optimal policy for $T = 20$ by clicking in the following links:

- When the minotaur cannot stay
  https://drive.google.com/file/d/1oI8o2_VNryXkh-EomWWPFLZwTiOTNbsj/view?usp=sharing
- When the minotaur can stay
  https://drive.google.com/file/d/1AoDKMFyFMPvKOEzVnX_hJPs52Qhz01s7/view?usp=sharing

The probabilities of escaping are plotted for the case if the minotaur is not allowed to stay and for the case in which the minotaur is allowed to stay, in figure 1 and figure 2 respectively.

We can see that the probability of escaping is not the same at the same time horizon T. This is because a strategy can be much more simple in the case where the minotaur is not allowed to stay. In that case, if the player and minotaur are in adjacent states, the player can move to the minotaur state and have certainty that it will not be eaten. This way the player can easily pass the minotaur, he just has to make sure that he is not in a possible next state of the minotaur.

When the minotaur is allowed to stay, this trick does not work anymore. This means that there is a larger chance of being eaten and that the player has to be more careful with its strategy. This means that the player thus has a smaller chance of being able to escape at the same time horizon.

(c) Assume now that your life is geometrically distributed with mean = 30. Modify the problem so as to derive a policy minimizing the expected time to exit the maze. Motivate your new problem formulation. Estimate the probability of getting out alive using this policy by simulating 10 000 games.

To complete this part of the assignment, we used the Value Iteration algorithm to compute the optimal policy instead of Dynamic Programming, since we are now dealing with an infinite horizon discounted problem. Since we are now assuming that the time is geometrically distributed with mean $\mu = 30$, and the first moment of a geometry distribution is $\mu = \frac{1}{1-\lambda}$, we set the discount factor $\lambda = 1 - \frac{1}{\mu} = 0.967$.

We estimated the probability of getting out alive using the computed policy by simulating 10000 games, and the resulting value was $\mathbb{P}(\text{Exiting alive} \mid \text{The minotaur cannot stay}) = 0.602$ and $\mathbb{P}(\text{Exiting alive} \mid \text{The minotaur can stay}) = 0.516$.

You can see the simulation of the resulting optimal policy in the following links.

- When the minotaur cannot stay
  https://drive.google.com/file/d/1p9AFPWVcLVLm3qiJzRMsN2_w6A6djtuj/view?usp=sharing
- When the minotaur can stay
  https://drive.google.com/file/d/1SA9maZCF4QgDytAjSOKpeQVSPVUkfr1v/view?usp=sharing

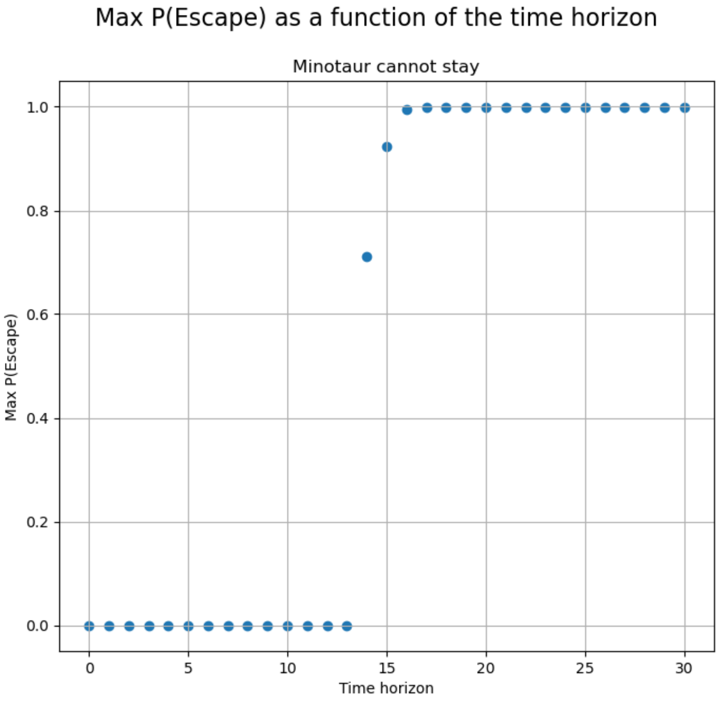

Figure 1: Probability of escaping if the minotaur cannot stay

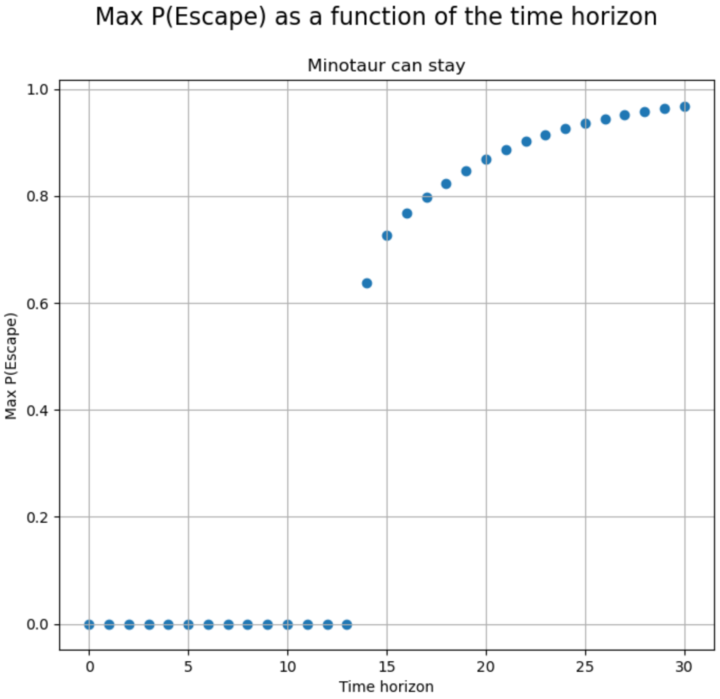

Figure 2: Probability of escaping if the minotaur can stay

Problem 2: Robbing Banks

(a) Formulate the problem as an MDP.

Using a similar setting as the first problem, we will define the states as a cartesian product of the Player and Police possible positions inside the predifined town:

$$\mathcal{S} = \mathcal{S}_{Player} \times \mathcal{S}_{Police} = \{(i,j) \in [0,2] \times [0,5]\}^2$$

The action space was defined by the assignment, and we note it down as follows:

$$\mathcal{A} = \{\text{Stay}, \text{Move left}, \text{Move right}, \text{Move up}, \text{Move down}\}$$

The reward function is also defined by the assignment. Let $(s, s_p) \in \mathcal{S}$ denote the respective player and police positions and $a \in \mathcal{A}$:

$$r((s,s_p),a) = \begin{cases} 10 & s \neq s_p \text{ and } s_p \in Bank \\ -50 & s = s_p \\ 0 & otherwise \end{cases}$$

In order to define the transition probabilities, we take into account that the player's moves are deterministic (i.e. the probability of moving towards the selected action is 1 unless it falls outside the town). Therefore

$$\mathbb{P}((s',\cdot),(s,\cdot),a) = 1 \text{ if } s' \in \mathcal{S}_{Player} \text{ and } a \in \mathcal{A}.$$

Moreover, if the police catches the player, the game is reinitialised.

$$\mathbb{P}((s_{Initial}, s_{p,Initial}),(s,s_p),a) = 1 \text{ if } s = s_p \text{ and } a \in \mathcal{A}.$$

On the other hand, the police moves randomly in the player's position. Let $(s, s_p) \in \mathcal{S}$,

$\mathbb{P}((\cdot, s'_p),(s,s_p),\cdot) = 0$ if the randomly chosen action goes in the opposite direction of the player (s).

$\mathbb{P}((\cdot, s'_p),(s,s_p),\cdot) = 1/k$ if the randomly chosen action does not go in the opposite direction of the player, and where $k = \#$Possible directions from s_p to s.

(b) Solve the problem, and display the value function (evaluated at the initial state) as a function of λ. Illustrate an optimal policy for different values of λ – comment on the behaviour.

You can see the value function evaluated at the initial state as a function of λ in figure 3. We can see that the larger the value of λ is, the bigger is the value function. This is an expected outcome because λ is the discount factor in our MDP formulation. Hence, for small values (close to 0) we are discounting a big part of the expected reward, and for values closer to 1 the time horizon gets longer and then the expected reward is bigger.

- Robber and Police chase for lambda=0.25 `https://drive.google.com/file/d/1nt1hOm7a0uoQ6Ratx7DDCCRN18enkOMy/view?usp=sharing`
- Robber and Police chase for lambda=0.75 `https://drive.google.com/file/d/1IOBsEqw6Le4ZE_kasGFwkbEYHMy1J4cX/view?usp=sharing`
- Robber and Police chase for lambda=0.9 `https://drive.google.com/file/d/12JO92BZzsIphWZKO_3DaZSdJDrMkPEkV/view?usp=sharing`

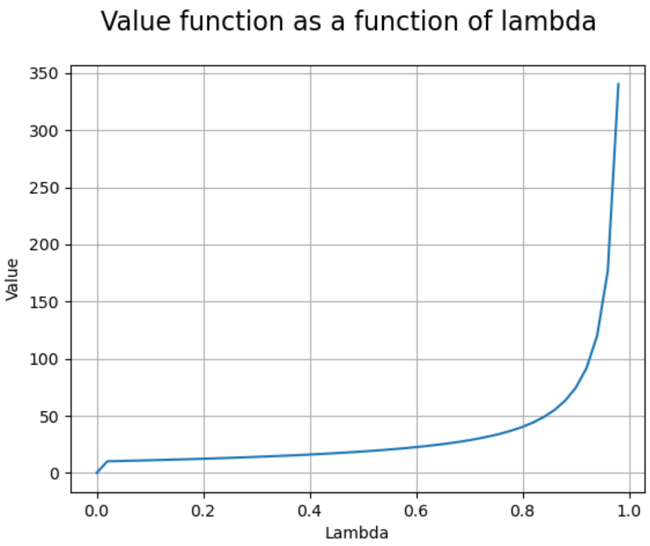

Figure 3: Value function as function of λ

- Robber and Police chase for lambda=0.98 `https://drive.google.com/file/d/1SwhxTTt27lbaYWmmtwYCRC5dXKpuDf86/view?usp=sharing`

From the previous illustrations we see that, as we discussed earlier, for small λ values the game finishes quite fast. For values closer to 1 the game takes longer to finish and the player can move to other cells while the police is randomly trying to chase him. It is visible that the police strategy is very unintelligent since the player can just stay in a position and move to the police's position if the police is in an adjacent cell (which is certainly safe since the police cannot stay in the same position). This way, the player can just move between the two left banks and never get caught. The player more visibly uses this strategy when the games last longer, or in other words, generally games with a higher lambda value.