EL2805 HT22 Lab1 Report

Philipp Katterbch, 20000510-T472
philipp.katterbach@rwth-achen.de

Markus Pietschner, 19990814-T378
markus.pietschner@rwth-achen.de

Abstract

Report for Lab1 of EL2805 Reinforcement Learning.

1 Task 1

1.a

MDP:

$$S = (x, y, \tilde{x}, \tilde{y}) \in \mathbb{Z}^4 \mid 0 \leq x, \tilde{x} \leq 6 \;\; 0 \leq y, \tilde{y} \leq 7$$
$$A = (\text{"}up\text{"}, \text{"}down\text{"}, \text{"}left\text{"}, \text{"}right\text{"}, \text{"}stay\text{"})$$
$$r_t = \begin{array}{cc} -100 & | \quad (x_{t+1}, y_{t+1}) = (\tilde{x}_{t+1}, \tilde{y}_{t+1}) \\ -100 & | \quad \text{"action leads into wall or outside map"} \\ 0 & | \quad (x_{t+1}, y_{t+1}) = (5, 6) \\ -1 & | \quad \text{else} \end{array}$$

If no wall in next state:

$$P((x_{t+1}, y_{t+1}) = (x_t + 1, y_t) | A = \text{"}right\text{"}) = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t - 1, y_t) | A = \text{"}left\text{"}) = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t, y_t + 1) | A = \text{"}up\text{"}) = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t, y_t - 1) | A = \text{"}down\text{"}) = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t, y_t) | A = \text{"}stay\text{"}) = 1$$

If wall in next state:

$$P((x_{t+1}, y_{t+1}) = (x_t, y_t) | A = \text{"}right\text{"}) = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t, y_t) | A = \text{"}left\text{"}) = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t, y_t) | A = \text{"}up\text{"}) = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t, y_t) | A = \text{"}down\text{"}) = 1$$

Minotaur transitions, not at edge of map or corner:

$$P((\tilde{x}_{t+1}, \tilde{y}_{t+1}) = (\tilde{x}_t + 1, \tilde{y}_t)) = 0.25$$
$$P((\tilde{x}_{t+1}, \tilde{y}_{t+1}) = (\tilde{x}_t - 1, \tilde{y}_t)) = 0.25$$
$$P((\tilde{x}_{t+1}, \tilde{y}_{t+1}) = (\tilde{x}_t, \tilde{y}_t + 1)) = 0.25$$
$$P((\tilde{x}_{t+1}, \tilde{y}_{t+1}) = (\tilde{x}_t, \tilde{y}_t - 1)) = 0.25$$

Minotaur transitions i.i.d. between allowed movements

Minotaur always stays in map, may enter walls.

Episode ends when in goal or eaten.

Preprint. Under review.

1.b

In the former scenario the agent was able to always evade the Minotaur by not standing still. This is now no longer possible. Rewards must be updated twice, once for moving the agent and once for moving the Minotaur.

MDP:

$$S = (x, y, \tilde{x}, \tilde{y}) \in \mathbb{Z}^4 \mid 0 \leq x, \tilde{x} \leq 6 \;\; 0 \leq y, \tilde{y} \leq 7$$

$$A = (\text{"}up\text{"}, \text{"}down\text{"}, \text{"}left\text{"}, \text{"}right\text{"}, \text{"}stay\text{"})$$

$$r_t = \begin{array}{cc|c} -100 & | & (x_{t+1}, y_{t+1}) = (\tilde{x}_t, \tilde{y}_t) \\ -100 & | & \text{"action leads into wall or outside map"} \\ 0 & | & (x_{t+1}, y_{t+1} = (5, 6) \\ -1 & | & \text{else} \end{array}$$

$$\tilde{r}_t = \begin{array}{cc|c} -100 & | & (\tilde{x}_{t+1}, \tilde{y}_{t+1}) = (x_{t+1}, y_{t+1}) \\ 0 & | & \text{else} \end{array}$$

$$r_{\text{total}} = r_t + \tilde{r}_t$$

If no wall in next state:

$$P((x_{t+1}, y_{t+1}) = (x_t + 1, y_t) | A = \text{"}right\text{"}) = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t - 1, y_t) | A = \text{"}left\text{"}) = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t, y_t + 1) | A = \text{"}up\text{"}) = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t, y_t - 1) | A = \text{"}down\text{"}) = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t, y_t) | A = \text{"}stay\text{"}) = 1$$

If wall in next state:

$$P((x_{t+1}, y_{t+1}) = (x_t, y_t) | A = \text{"}right\text{"}) = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t, y_t) | A = \text{"}left\text{"}) = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t, y_t) | A = \text{"}up\text{"}) = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t, y_t) | A = \text{"}down\text{"}) = 1$$

Minotaur transitions, not at edge of map or corner:

$$P((\tilde{x}_{t+1}, \tilde{y}_{t+1}) = (\tilde{x}_t + 1, \tilde{y}_t)) = 0.25$$
$$P((\tilde{x}_{t+1}, \tilde{y}_{t+1}) = (\tilde{x}_t - 1, \tilde{y}_t)) = 0.25$$
$$P((\tilde{x}_{t+1}, \tilde{y}_{t+1}) = (\tilde{x}_t, \tilde{y}_t + 1)) = 0.25$$
$$P((\tilde{x}_{t+1}, \tilde{y}_{t+1}) = (\tilde{x}_t, \tilde{y}_t - 1)) = 0.25$$

Minotaur transitions i.i.d. between allowed movements

Minotaur always stays in map, may enter walls.

Episode ends when in goal or eaten.

1.c

Problem was solved using Dynamic Programming. The optimal policy is independent of the Minotaur. The policy is depicted in Figure 1.

1.d

If the Minotaur is not allowed to stand still, the optimal policy is to take the shortest path. The probability to exit turns to 1 when $T \geq 15$. This is plotted in 2a.

This changes if the Minotaur is able to stay in his field as now the agent actually has to evade the Minotaur. The probability of success over 1000 runs per time horizon is plotted in 2b.

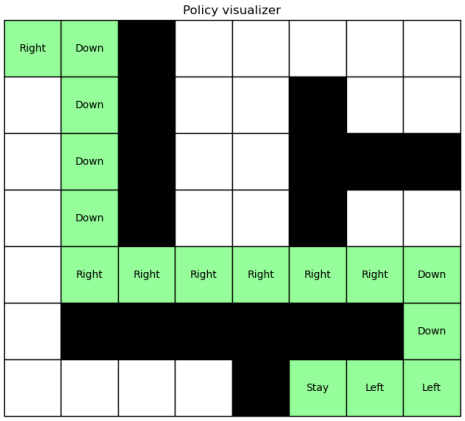
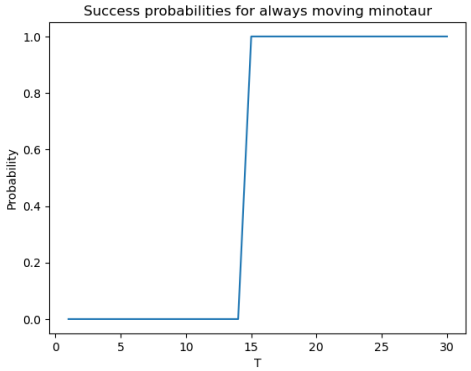
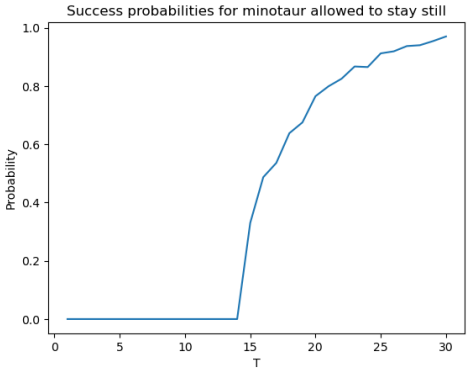


Figure 1: Optimal policy for 1c





(a) Probability of exit when the Minotaur must move. (b) Probability of exit when the Minotaur may stay.

Figure 2: Probability of exit

1.e

The Minotaur cannot stand still and we want to exit the maze as fast as possible, therefore we don't want to stand still. As the starting distance between Minotaur and us is odd, the Minotaur can never stand on the same field as us, if we don't stand still. Therefore, the Minotaur will be ignored and we dismiss "stay" as an action to simplify the MDP. As trying to walk in a wall resulted in "stay" in the previous MDP and is very suboptimal, we only allow actions, which don't enter walls or leave the map. We introduce a new state modeling death by poison.

The simulation still simulates the Minotaur and allows the action "stay". The policy however converges still to take the shortest path independent of the Minotaur.

MDP:

$$S = (x, y, d)) \in \mathbb{Z}^3 \mid 0 \leq x \leq 6 \;\; 0 \leq y \leq 7 \;\; 0 \leq d \leq 1$$

$$A = \begin{array}{c|l} "up" & \text{if } (x_t, y_t + 1) \text{ is in map and not in wall} \\ "down" & \text{if } (x_t, y_t - 1) \text{ is in map and not in wall} \\ "left" & \text{if } (x_t - 1, y_t) \text{ is in map and not in wall} \\ "right" & \text{if } (x_t + 1, y_t) \text{ is in map and not in wall} \end{array}$$

$$r_t = \begin{array}{c|c} -100 & d_{t+1} = 1 \\ 0 & (x_{t+1}, y_{t+1}, d_{t+1}) = (5, 6, 0) \\ -1 & \text{else} \end{array}$$

$$P((x_{t+1}, y_{t+1}) = (x_t + 1, y_t) | A = "right") = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t - 1, y_t) | A = "left") = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t, y_t + 1) | A = "up") = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t, y_t - 1) | A = "down") = 1$$
$$P(d_{t+1} = 1 | d_t = 0) = \frac{1}{30}$$
$$P(d_{t+1} = 0 | d_t = 0) = \frac{29}{30}$$
$$P(d_{t+1} = 1 | d_t = 1) = 1$$

Episode ends when in goal: $(5, 6, 0)$ or when dead: $(\cdot, \cdot, 1)$

The policy maximizing the reward is to take the shortest path to the exit taking 15 steps.

1.f

The probability of getting out alive is $P = \frac{29}{30}^{15} = 0.601$. Our simulation approaches this value.

1.g

Off policy:

- Agent picks the action a_t and subsequent actions depending on Q^t (or other estimated quantity) using the policy $\pi_t(Q^t)$. Often ϵ-greedy.
- Agent assumes it will move according to its own state in the future.
- This action is most likely very bad at the start of training.
- Essentially learning by doing.
- Own policy needs to explore.

On policy:

- Agent assumes it will use a given behaviour policy in the future.
- Eventually, the agent learns the value of the given behaviour.
- Essentially learning by watching another person act.
- Behaviour needs to explore.

Q Learning Convergence:

- $\sum_0^\infty \alpha_t \to \infty$
- $\sum_{t=0}^\infty \alpha_t^2 < \infty$
- Behaviour policy π_b visits every (state,action) pair infinitely often.

=> Q^t converges almost surely to the optimal value for any $\lambda \in (0, 1)$

Sarsa Convergence:

- $\sum_0^\infty \alpha_t \to \infty$

- $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$
- π_t follows ϵ-greedy policy relative to Q^t

=> Q^t converges almost surely to $Q^{\bar{\pi}}$ for any $\lambda \in (0, 1)$, where $Q^{\bar{\pi}}$ tends to Q as $\epsilon \to 0$

1.h

Same assumptions as e. Furthermore, a state for the key is introduced.

MDP:

$$S = (x, y, d, k) \in \mathbb{Z}^4 \mid 0 \leq x \leq 6 \;\; 0 \leq y \leq 7 \;\; 0 \leq d \leq 1 \;\; 0 \leq k \leq 1$$

$$A = \begin{array}{cc|l} & "up" & \text{if } (x_t, y_t + 1) \text{ is in map and not in wall} \\ & "down" & \text{if } (x_t, y_t - 1) \text{ is in map and not in wall} \\ & "left" & \text{if } (x_t - 1, y_t) \text{ is in map and not in wall} \\ & "right" & \text{if } (x_t + 1, y_t) \text{ is in map and not in wall} \end{array}$$

$$r_t = \begin{array}{cc|c} & -100 & d_{t+1} = 1 \\ & 0 & (x_{t+1}, y_{t+1}, d_{t+1}, k_{t+1}) = (5, 6, 0, 1) \\ & -1 & \text{else} \end{array}$$

$$P((x_{t+1}, y_{t+1}) = (x_t + 1, y_t) | A = "right") = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t - 1, y_t) | A = "left") = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t, y_t + 1) | A = "up") = 1$$
$$P((x_{t+1}, y_{t+1}) = (x_t, y_t - 1) | A = "down") = 1$$
$$P(k_{t+1} = 1 | (x_t, y_t) = (6, 1), a = \text{right}) = 1$$
$$P(k_{t+1} = 1 | (x_t, y_t) = (7, 2), a = \text{up}) = 1$$
$$P(k_{t+1} = k_t | \text{not one of the two conditions above}) = 1$$
$$P(d_{t+1} = 1 | d_t = 0) = \frac{1}{50}$$
$$P(d_{t+1} = 0 | d_t = 0) = \frac{49}{50}$$
$$P(d_{t+1} = 1 | d_t = 1) = 1$$

Episode ends when in goal with key: $(5, 6, 0, 1)$ or by death: $(\cdot, \cdot, 1, \cdot)$

Shortest path is 29. Getting out with probability $P = \frac{49}{50}^{29} = 0.557$ using the shortest path approach.