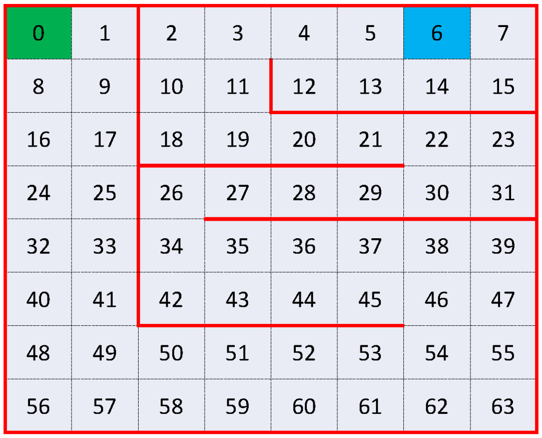
CPT_S 580 – Reinforcement Learning

Homework 2

Due: 11:59 PM on Feb. 16

1. A Maze Game in Gridworld [10pt, 10pt, 10pt]

A 8×8 rectangular gridworld map is shown below:



0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Figure 1: Gridworld map.

Imagine an agent moving within this 8×8 gridworld, where the grids are numbered from 0 to 63, the red lines represent walls, and the green grid (grid 0) and the blue grid (grid 6) are the starting point and the destination, respectively. The agent can move up, right, down, or left for each step except that the walls block the agent's path. That is, if there is a wall in the direction that the agent plans to move, the agent will remain in the current cell.

If the agent arrives at grid 6 (the destination), the agent will receive a reward of $+10$ and the process will terminate. Otherwise, the agent will receive a reward of -1 with probability 0.5 and a reward of -2 with probability 0.5 for each step (including hitting the wall).

The agent's goal is to find the optimal policy that maximizes the expected discounted total reward starting from grid 0 to grid 6. The discount factor is 0.9.

Formulation:

- *State* s:

The state is defined as the grid where the agent is located. $s \in \{0, 1, ..., 63\}$. The initial state is 0 and the terminal state is 6.

- *Action a*:

  $a = 0$: the agent plans to move *up*;

  $a = 1$: the agent plans to move *right*;

  $a = 2$: the agent plans to move *down*;

  $a = 3$: the agent plans to move *left*.

- *Transition*:

  Examples:

  If $s = 0$ and $a = 0$, then the next state will be $s' = 0$ (The agent hits the wall);

  If $s = 0$ and $a = 1$, then the next state will be $s' = 1$;

  If $s = 0$ and $a = 2$, then the next state will be $s' = 8$;

- *Random reward $r(s, a)$*:

  $r(5, 1) = 10$, $r(7, 3) = 10$, $r(14, 0) = 10$. Otherwise, $r(s, a)$ is equal to -1 with probability 0.5 and -2 with probability 0.5.

- *Objective*: Maximize the expected discounted total reward:

$$\mathbb{E}\left[\sum_{t=0}^{\tau-1} \alpha^t r(s_t, a_t) | s_0 = 0\right]$$

  where the subscript t denotes the time slot, τ is the time slot when the agent reaches the terminal state 6 starting from s_0, and $\alpha = 0.9$ is the discount factor.

Note that in this problem the state transition is deterministic. Define a deterministic state transition function f such that $s' = f(s, a)$ where s is the current state, a is the current action, and s' is the next state. For example, $f(0, 0) = 0$, $f(0, 1) = 1$.

Let $V^*(s)$ denote the optimal value function, defined by

$$V^*(s) = \max_{\mu} \mathbb{E}\left[\sum_{t=0}^{\tau-1} \alpha^t r(s_t, \mu(s_t)) | s_0 = s\right]$$

(1) Write down the Bellman equation in terms of V^*, f, r, and α. Hint: Consider two cases, $s = 6$ and $s \neq 6$.

(2) Assume $\mathbf{V}_0 = \mathbf{0}_{64}$, where $\mathbf{0}_n$ denotes a column vector of all zeros with length n. Please calculate $\mathbf{V}_1$ under the value iteration algorithm, i.e., the value function after the first iteration of the value iteration algorithm.

(3) Assume initial policy $\mu_0(s) = 0$ for any $s \in \{0, 1, ..., 63\}$. Please calculate $\mathbf{V}_{\mu_0}$, the value function under policy μ_0, and then identify μ_1 under the policy iteration algorithm (i.e. the policy after the first iteration of the policy iteration algorithm).

2. Q-Learning [10pt]

Consider a Markov chain with three states $\{1, 2, 3\}$. In each state, we can choose one of the two possible actions $\{1, 2\}$. The transition probability matrices under the two actions are given below:

$$\mathbf{P}(1) = \begin{pmatrix} 0.5 & 0.3 & 0.2 \\ 0.1 & 0.4 & 0.5 \\ 0.3 & 0.3 & 0.4 \end{pmatrix} \quad \text{and} \quad \mathbf{P}(2) = \begin{pmatrix} 0.3 & 0.3 & 0.4 \\ 0.5 & 0.1 & 0.4 \\ 0.2 & 0.5 & 0.3 \end{pmatrix}. \tag{1}$$

The cost for a given (state, action) pair is a Bernoulli random variable. The mean costs are given below

$$\mathbf{C} = \begin{pmatrix} 0.1 & 0.9 \\ 0.8 & 0.1 \\ 0 & 0 \end{pmatrix}. \tag{2}$$

We are interested in solving the following discounted cost problem

$$\min_{\mu} \lim_{N \to \infty} E \left[\sum_{k=0}^{N} 0.9^k c(x_k, u_k) \middle| x_0 = 1, u_0 = 1 \right]$$

where x_k is the state at time k, u_k is the action at time k, and μ denotes a policy.

Assume we do not know the model but are given the following trace $(x_k, u_k, c(x_k, u_k))$ instead:

$$(1, 1, 0) \to (2, 1, 1) \to (3, 2, 0) \to (2, 2, 1). \tag{3}$$

Consider the Q-learning algorithm with $Q_0 = \begin{pmatrix} 0 & 0.5 \\ 0.3 & 0 \\ 0.2 & 0.1 \end{pmatrix}$ and step size $\epsilon = 0.1$. Please calculate the sequence of Q-values under Q-learning with the trace given above.

3. A Maze Game in Gridworld (coding part) [30pt]

Please download the files `hw_2.ipynb` and `gridworld_maze.py` from Canvas and complete the codes in `hw_2.ipynb` following the instructions in the file. You can run and debug your codes using Google Colab. After completing the tasks, save `hw_2.ipynb` as `hw_2.py` by clicking "File → Download → Download .py". Then upload `hw_2.py` to Gradescope. The autograder on Gradescope will automatically grade your coding homework.