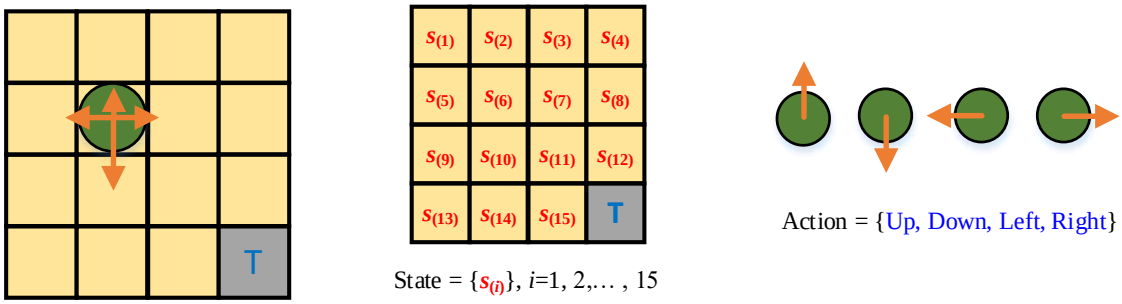
<Reinforcement Learning and Control>

Homework A

1.1 Problem description

Consider an indoor robot cleaner in a gridded room:



(a) Grid environment (b) State space (c) Action space

Figure 1 Cleaning robot in a grid room

The robot is assumed to work in a rectangular grid environment. Each cell represents one state of the grid environment, with only one destination (Cell T). The terminal state is shaded in Figure 1. The robot has four actions: “Up”, “Down”, “Left”, and “Right”, with the purpose of reaching the destination as soon as possible. The state space of the grid environment is

$$\mathcal{S} = \{s_{(1)}, s_{(2)}, \cdots, s_{(15)}, s_{(16)}\}$$

The action space of the robot is

$$\mathcal{A} = \{\text{Up}, \text{Down}, \text{Left}, \text{Right}\}$$

The robot dynamics is described as follows. Each action can move the cleaner to its neighboring cells in a random behavior. After taking any action from $\mathcal{A}$, its next state $s_{t+1} \in \mathcal{S}$ (if available) becomes:

$$\Pr\{s' = \text{Front Cell} \mid s = \text{Cell}, a\} = 0.8$$
$$\Pr\{s' = \text{Neighboring Cells} \mid s = \text{Cell}, a\} = 0.1$$
$$\Pr\{s' = \text{Back Cell} \mid s = \text{Cell}, a\} = 0$$

Note that “Neighboring”, “Front” and “Back” is fixed with the action direction of the robot. The robot dynamics forces the robot to move forward in the probability of 0.8, and move to neighboring cells, either leftward or rightward, in 0.1, respectively. There is no possibility for it to move backward. In addition, the agent must not move outside of the grid such that the agent at the corners or along the boundaries might have an unavailable state at next step. If the next state is not available, the robot keeps its current state. In addition, the terminal state is absorbing which will always return to itself regardless of any action.

The design of reward signals follows the fact that each action consumes a certain amount of energy, and the destination cell will fully recharge the robot. Each move results in a penalty of -1, and the robot stops after reaching the destination, i.e., termination state, with a reward +9.

$$r(s,a,s') = \begin{cases} -1 & \text{if } s' \neq \mathrm{T} \\ +9\text{'} & \text{if } s' = \mathrm{T} \end{cases}$$

Note that discounting factor $\gamma = 0.9$ is selected here. Here, we suppose a stochastic policy π, i.e., for each non-terminal state $s_{(i)}, i = 1,2,\dots,15$

$$\pi(a = \mathrm{Up}|s_{(i)}) = 0.25$$
$$\pi(a = \mathrm{Down}|s_{(i)}) = 0.25$$
$$\pi(a = \mathrm{Left}|s_{(i)}) = 0.25$$
$$\pi(a = \mathrm{Right}|s_{(i)}) = 0.25$$

1.2 Questions

(1) Please calculate the transition probability matrix under the stochastic policy π, i.e., fill in the following 16×16 matrix:

$$\mathcal{P} = \begin{bmatrix} a_{1,1} & & & & \\ & \ddots & & & \\ & & a_{i,j} & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}_{16\times 16}$$

where each entry $a_{i,j}$ represents the transition probability from state $s_{(i)}$ to $s_{(j)}$ $(i,j = 1,2,\dots,16)$. Note that $a_{16,16} = 1$ since it is an absorbing state.

(2) Please calculate each entity of state-value function, i.e., $V(s)$, by Monte Carlo estimation.

$$\begin{bmatrix} V(s_{(1)}) & V(s_{(2)}) & V(s_{(3)}) & V(s_{(4)}) \\ V(s_{(5)}) & V(s_{(6)}) & V(s_{(7)}) & V(s_{(8)}) \\ V(s_{(9)}) & V(s_{(10)}) & V(s_{(11)}) & V(s_{(12)}) \\ V(s_{(13)}) & V(s_{(14)}) & V(s_{(15)}) & 0 \end{bmatrix}$$

(3) Please evaluate the policy performance R_{Avg} of the stochastic policy π by Monte Carlo simulation:

$$R_{\mathrm{Avg}} = \sum d_{\mathrm{init}}(s)\left\{\frac{1}{N}\sum_{i=1}^{N} G_i(s)\right\}$$

where $d_{\mathrm{init}}(s)$ is the initial state distribution. Here we suppose it is a uniform distribution over all 15 non-terminal states, i.e., $p(s = s_{(i)}) = 1/15$. Note that as a tradition, each return $G_i(s)$ is calculated by enforcing discounting factor $\gamma = 1$ in policy performance evaluation, which is different from value function estimation.

(4) Please find the optimal tabular policy with one of the following model-free RL algorithms, including Sarsa, Q-learning and Expected Sarsa, and evaluate its policy performance R_{Avg}.