# Homework 5. Reinforcement learning

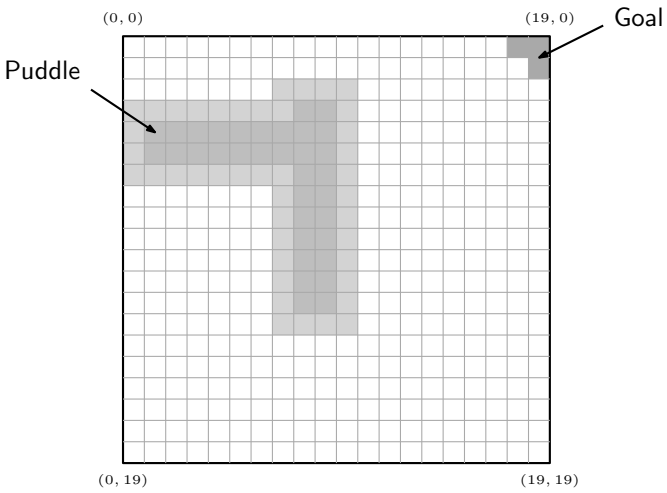


Figure 1: Puddle world.

Consider an all terrain vehicle navigating the grid world depicted in Fig. 1. The three shaded cells in the upper right corner correspond to the goal state, while the L-shaped shaded cells in the middle of the grid correspond to a puddle in which the vehicle may get stuck and damaged.

The vehicle has available the standard four actions—up, down, left and right. Each action

- Succeeds and moves the vehicle to the adjacent cell in the corresponding direction with a probability of 0.92;
- Fails and moves the vehicle to any of the other 3 adjacent cells with a probability of 0.02;
- Fails and the vehicle remains in the same cell with a probability of 0.02.

See Fig. 2 for an illustration of some movement situations.

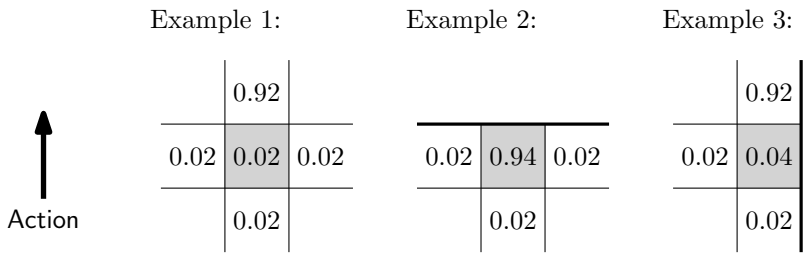


Figure 2: Movement of the vehicle in the puddle world. The examples depict the transition probabilities to the different cells when the vehicle is originally in the shaded cell and takes action "Up".

The vehicle incurs maximal cost (1) for standing in the darker part of the puddle; in the lighter part of the puddle, it incurs a cost of 0.5. Standing in the goal cells costs 0, while all other cost 0.05. Consider a discount $\gamma = 0.95$.

Suppose that the agent is allowed to move around the environment for $T$ time steps, experimenting the different actions and visiting the different states. A excerpt of its trajectory is

$$\begin{gathered}
\dots \\
x_{t-1} = (17, 3), \\
a_{t-1} = L, \\
c_{t-1} = 0.05, \\
x_t = (16, 3), \\
a_t = D, \\
c_t = 0.05, \\
x_{t+1} = (16, 4), \\
a_{t+1} = L, \\
c_{t+1} = 0.05, \\
\dots
\end{gathered}$$

At time step $t$, the $Q$-values estimated by the for state $(16, 3)$ are:

$$\mathbf{Q}^{(t)}_{(16,3),:} = \begin{bmatrix} 0.25 & 0.32 & 0.32 & 0.25 \end{bmatrix}$$

and for state $(16, 4)$,

$$\mathbf{Q}^{(t)}_{(16,4),:} = \begin{bmatrix} 0.29 & 0.36 & 0.36 & 0.29 \end{bmatrix}.$$

## Exercise 1.

(a) Indicate the $Q$-values after a $Q$-learning update with step-size $\alpha = 0.1$, resulting from the transition at time step $t$.

(b) Indicate the $Q$-values after a SARSA update with step-size $\alpha = 0.1$, resulting from the transition at time step $t$.

(c) Explain the difference between on-policy and off-policy learning using Questions 1a and 1b to illustrate your explanation.

**Solution 1:**

(a) The $Q$-learning update is given by

$$Q^{(t+1)}(x_t, a_t) = Q^{(t)}(x_t, a_t) + \alpha(c_t + \gamma \min_{a' \in \mathcal{A}} Q^{(t)}(x_{t+1}, a') - Q^{(t)}(x_t, a_t))$$

where, from the provided transitions,

$$x_t = (16, 3), \qquad a_t = D, \qquad c_t = 0.05, \qquad x_{t+1} = (16, 4).$$

This results in the update

$$Q^{(t+1)}((16, 3), D) = 0.32 + 0.1 \times (0.05 + 0.95 \times 0.29 - 0.32) = 0.32.$$

(b) The SARSA update is given by

$$Q^{(t+1)}(x_t, a_t) = Q^{(t)}(x_t, a_t) + \alpha(c_t + \gamma Q^{(t)}(x_{t+1}, a_{t+1}) - Q^{(t)}(x_t, a_t))$$

where, from the provided transitions,

$$x_t = (16, 3), \qquad a_t = D, \qquad c_t = 0.05, \qquad x_{t+1} = (16, 4) \qquad a_{t+1} = L.$$

This results in the update

$$Q^{(t+1)}((16, 3), D) = 0.32 + 0.1 \times (0.05 + 0.95 \times 0.36 - 0.32) = 0.33.$$

(c) An on-policy learning algorithm learns the values associated with the policy used to sample the MDP. For example, SARSA is an on-policy algorithm, as it learns the $Q$-values for the learning policy. This is apparent in its dependence on the action $a_{t+1}$, as seen in Question 1b.

An off-policy learning algorithm, on the other hand, learns the value associated with some target policy—not necessarily the one used to sample the MDP. For example, $Q$-learning is an off-policy algorithm, as it learns the $Q$-values for the optimal policy independently of the learning policy. This is apparent in its computation of the action $a'$ that minimizes $Q(x_{t+1}, a')$, as seen in Question 1a.