MSc in Computer Science and Engineering

Learning and Decision Making 2017-2018

Homework 2. Markov decision problems

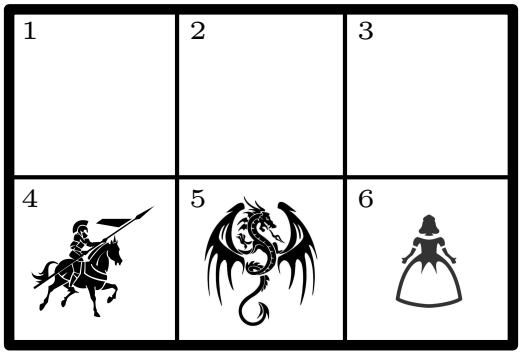

Figure 1: Grid environment where a knight must save a princess and avoid the dragon. The princess and the dragon do not move.

Consider a knight seeking to save a captured princess in a 2×3 grid world (see Fig. **??**). The princess is held in position 6 of the grid, while a dragon guards the passage in position 5 of the grid.

At each step, the knight may move in any of the four directions—up, down, left and right. The movement succeeds with a 0.6 probability and fails with a 0.4 probability. When the movement fails, the knight may stay in the same cell or move to one of the immediately adjacent cells (if there is one) with equal probability.[1] See Fig. **??** for some examples.

The goal of the knight is to save (reach) the princess and avoid the dragon. In this homework, you will model the decision of the knight as a Markov decision problem (MDP).

Exercise 1.

(a) Identify the state space, $\mathcal{X}$, and the action space, $\mathcal{A}$, for the MDP.

[1] When that implies going beyond the limits of the grid, the corresponding probability adds to the probability of staying in the same cell.

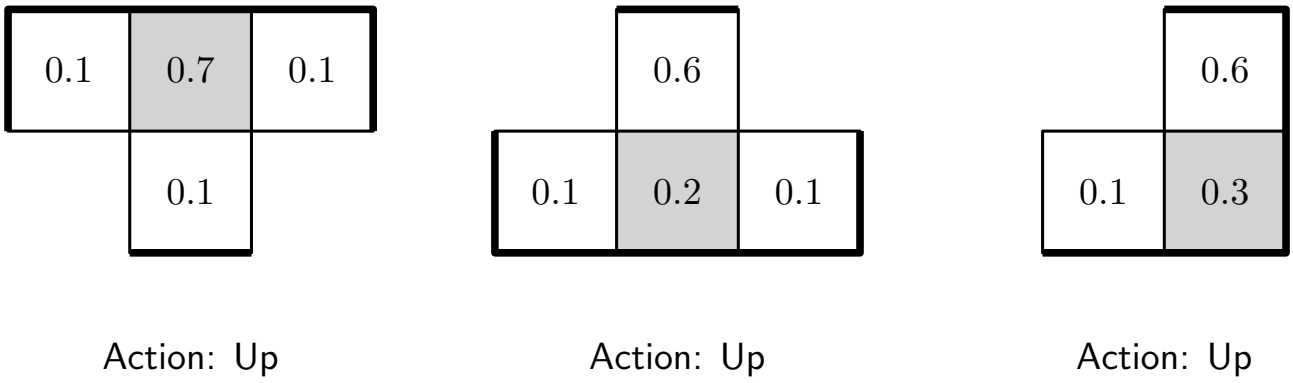

Figure 2: Examples of how transition probabilities change depending on the knight's position in the grid.

(b) Write down the transition probabilities and a (possible) cost function for the MDP. Make sure that the cost function is as simple as possible and verifies $c(x, a) \in [0, 1]$ for all states $x \in \mathcal{X}$ and actions $a \in \mathcal{A}$. Note, in particular, that the cost should depend only on the knight *standing* in the same cell as the princess (success) or the dragon (penalty).

(c) Compute the cost-to-go function associated with the policy in which the knight always goes up, using a discount $\gamma = 0.9$. You can use any software of your liking for the harder computations, but should indicate all other computations.

Solution 1:

The MDP model describes the knight's decision process. Since the dragon and the princess do not move, the state needs only to contain information about the position of the knight.

(a) We can represent the state as

$$\mathcal{X} = \{1, 2, 3, 4, 5, 6\},$$

where the numbers correspond to the cells in the map. The action space is $\mathcal{A} = \{U, D, L, R\}$, corresponding to the movements in the four directions (up, down, left and right).

(b) The transition probabilities come:

$$\boldsymbol{P}_U = \begin{bmatrix} 0.8 & 0.1 & 0.0 & 0.1 & 0.0 & 0.0 \\ 0.1 & 0.7 & 0.1 & 0.0 & 0.1 & 0.0 \\ 0.0 & 0.1 & 0.8 & 0.0 & 0.0 & 0.1 \\ 0.6 & 0.0 & 0.0 & 0.3 & 0.1 & 0.0 \\ 0.0 & 0.6 & 0.0 & 0.1 & 0.2 & 0.1 \\ 0.0 & 0.0 & 0.6 & 0.0 & 0.1 & 0.3 \end{bmatrix}, \quad \boldsymbol{P}_D = \begin{bmatrix} 0.3 & 0.1 & 0.0 & 0.6 & 0.0 & 0.0 \\ 0.1 & 0.2 & 0.1 & 0.0 & 0.6 & 0.0 \\ 0.0 & 0.1 & 0.3 & 0.0 & 0.0 & 0.6 \\ 0.1 & 0.0 & 0.0 & 0.8 & 0.1 & 0.0 \\ 0.0 & 0.1 & 0.0 & 0.1 & 0.7 & 0.1 \\ 0.0 & 0.0 & 0.1 & 0.0 & 0.1 & 0.8 \end{bmatrix},$$

$$\boldsymbol{P}_L = \begin{bmatrix} 0.8 & 0.1 & 0.0 & 0.1 & 0.0 & 0.0 \\ 0.6 & 0.2 & 0.1 & 0.0 & 0.1 & 0.0 \\ 0.0 & 0.6 & 0.3 & 0.0 & 0.0 & 0.1 \\ 0.1 & 0.0 & 0.0 & 0.8 & 0.1 & 0.0 \\ 0.0 & 0.1 & 0.0 & 0.6 & 0.2 & 0.1 \\ 0.0 & 0.0 & 0.1 & 0.0 & 0.6 & 0.3 \end{bmatrix}, \quad \boldsymbol{P}_R = \begin{bmatrix} 0.3 & 0.6 & 0.0 & 0.1 & 0.0 & 0.0 \\ 0.1 & 0.2 & 0.6 & 0.0 & 0.1 & 0.0 \\ 0.0 & 0.1 & 0.8 & 0.0 & 0.0 & 0.1 \\ 0.1 & 0.0 & 0.0 & 0.3 & 0.6 & 0.0 \\ 0.0 & 0.1 & 0.0 & 0.1 & 0.2 & 0.6 \\ 0.0 & 0.0 & 0.1 & 0.0 & 0.1 & 0.8 \end{bmatrix}.$$

As for the cost function, it depends only on the position of the knight, so it should be constant across actions. The dragon cell should yield maximum cost (1), and the cell with the princess should yield minimum cost (0). The other states should yield an intermediate value. Therefore, a possible cost function is:

$$\mathbf{C} = \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 \\ 1.0 & 1.0 & 1.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix},$$

which penalizes the knight with a cost of 0.2 in the intermediate states.

(c) To compute the cost to go function associated with that policy, we solve the linear system $J^\pi = \mathbf{c}_\pi + \gamma \boldsymbol{P}_\pi J^\pi$, where $\boldsymbol{P}_\pi = \boldsymbol{P}_U$ and $\mathbf{c}_\pi = \mathbf{C}_{:,U}$. The solution is given by:

$$J^\pi = (\mathbf{I} - \gamma \boldsymbol{P}_\pi)^{-1} \mathbf{c}_\pi = \begin{bmatrix} 2.23 & 2.39 & 2.12 & 2.33 & 3.27 & 1.97 \end{bmatrix}^\top .$$