[DT0171] - ARTIFICIAL INTELLIGENCE Reinforcement Learning Module a.a.2020/2021

Giovanni Stilo
Department of Information Engineering,
Computer Science and Mathematics
University of L'Aquila
giovanni.stilo@univaq.it

Instructions

- The current assignment must be carried out individually;
- The discussion of the solution must be submitted as PDF;
- Tabular answers must be attached as spreadsheets.
- The main report must refer to each one of the tables.
- The text is intentionally underspecified to check students' preparation;
- Answers and their attachments must be submitted through email (at least 3 days before the exam date) with the following subject:
  [DT0171] - Final Exam - <surname> <name> - 2020/21

The Cooking Chef Problem

Consider the case where the agent is your personal Chef.
In particular, the agent (the smiley on the map) wants to cook the scrambled eggs recipe.
In order to cook the desired recipe, the agent must first collect the needed tools (the egg beater on the map). Then he must reach the stove (the frying pan on the map). Finally, he can cook.

Since you have a lot of hungry, it is fundamentals that the agent cooks the scrambled eggs as fast he can do without let you waiting for more than the necessary.

In order to apply optimal control techniques such as value iteration, you need to model the aforementioned scenario as an MDP. Recall that an MDP is defined as tuple (S, A, P, R, γ), where:

S**:** The (finite) set of all possible states.

A**:** The (finite) set of all possible actions.

P**:** The transition function $P : S \times S \times A \to [0, 1]$, which maps (s', s, a) to $P(s'|s, a)$, i.e., the probability of transitioning to state $s' \in S$ when taking action $a \in A$ in state $s \in S$. Note that $\sum_{s' \in S} P(s'|s, a) = 1$ for all $s \in S, a \in A$.

R**:** The reward function $R : S \times A \times S \to \mathbb{R}$, which maps (s, a, s') to $R(s, a, s')$, i.e., the reward obtained when taking action $a \in A$ in state $s \in S$ and arriving at state $s' \in S$.

γ**:** The discount factor, which controls how important are rewards in the future.

In order to encode this problem as an MDP, you need to define each of the components of the tuple for our particular problem. Note that there may be many different possible encoding.

To answer the questions, consider the instance shown in Figure 1:

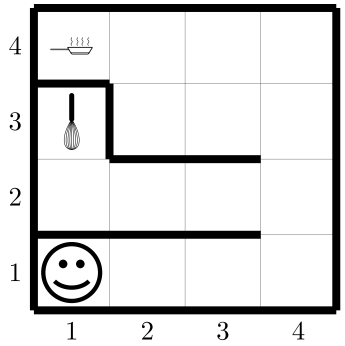

Figure 1: A particular instance of the cooking Chef problem. The goal is for the agent currently located in state $(1, 1)$ to have a policy that always leads to cook the eggs in location $(4, 4)$.

- In the figure, the agent is at $(1, 1)$ (but it can start at any of the grid cells).
- The needed cooking's tool as the egg beater, is in position $(1, 3)$.
- The final goal, displayed as the frying pan, is in position $(1, 4)$.
- The agent is not able to move in diagonal.
- Walls are represented by thick black lines.
- The agent cannot move through walls.
- The episode will ends when the agent successfully cook the scrambled eggs (see the above description).

Part a

Modeling the MDP as an infinite horizon MDP: the agent once he starts to successfully cook never end, and it remains into an absorbing state.

Using the above problem description answer the following questions:

a) Provide a concise description of the states of the MDP. How many states are in this MDP? (i.e. what is $|S|$).

b) Provide a concise description of the actions of the MDP. How many actions are in this MDP? (i.e. what is $|A|$).

c) What is the dimensionality of the transition function P?

d) Report the transition function P for any state s and action a in a tabular format.

e) Describe a reward function $R : S \times A \times S$ and a value of γ that will lead to an optimal policy.

f) Does $\gamma \in (0, 1)$ affect the optimal policy in this case? Explain why.

g) How many possible policies are there? (**All policies**, not just optimal policies.)

h) What is the optimal policy? Draw the grid and label each cell with an arrows in the direction of the optimal action. If multiple arrows, include the probability of each arrow. There may be multiple optimal policies, pick one and show it.

i) Is your policy deterministic or stochastic?

j) Is there any advantage to having a stochastic policy? Explain.

Part b

Now consider that our agent often goes the wrong direction because of how tired it is. Now each action has a 30% chance of going perpendicular to the left of the direction chosen and 20% chance of going perpendicular to the right of the direction chosen. Given this change answer the following questions:

a) Report the transition function P for any state s and action $a \in A$.

b) Does the optimal policy change compared to Part a? Justify your answer.

c) Will the value of the optimal policy change? Explain.