[DT0171] - ARTIFICIAL INTELLIGENCE

Reinforcement Learning

Module a.a.2022/2023

Prof. Giovanni Stilo

Department of Information Engineering,

Computer Science and Mathematics

University of L'Aquila

Student: Martina Nolletti

Department of Information Engineering,

Computer Science and Mathematics

University of L'Aquila

The Cooking Chef Problem

The Markov Decision Process is Markov Reward Process + actions (Markov Reward Process is a Markov Chain + rewards).

The MDP is defined as a tuple (S, A, P, R, γ), where:

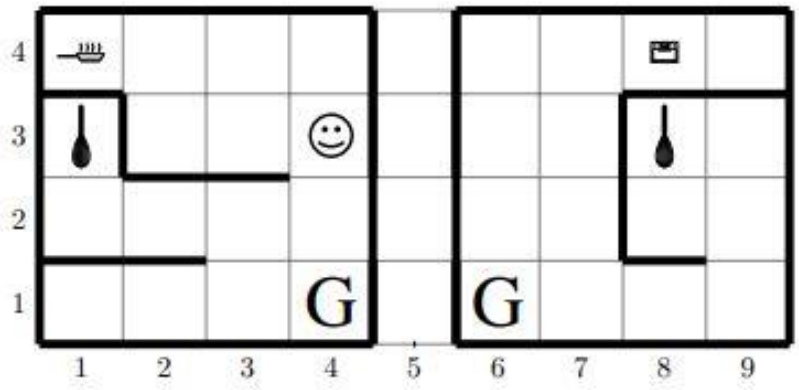
S: The (finite) set of all possible states.

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

P: The transition function P: $S \times S \times A \rightarrow [0, 1]$, which maps (s', s, a) to P(s' | s, a), i.e. the probability of transitioning to state $s' \in S$ when performing an action $a \in A$ in the 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 $R: S \times A \times S \rightarrow R$, which maps (s, a, s') to R(s, a, s'), i.e. the reward obtained when performing the action $a \in A$ in the state $s \in S$ and we arrive at the state $s' \in S$.

γ: The discount factor controls the importance of rewards in the future.



Part a.

Modeling the MDP as an infinite horizon MDP: the agent, once he starts to cook successfully, never ends, and it remains in 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|).

Each state S is associated with possible actions (A and B) that the rational agent can choose. Each action leads to a different future state (S' and S"). There can be two types of models:

- deterministic model in which every action leads 100% to a subsequent predetermined state. Each action A only has a probability of leading to the next desired state S'. There is also the risk that the action leads to a different state (Sx).
- probabilistic model more suitable for dealing with decisions under uncertainty. In addition to the probability of reaching the goal (S'), there is also the risk of falling into a worse state (Sx).

In this MDP there are 36 states. The agent can assume any of the 36 possible positions on the map.

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

The space of actions $A = U_s \in_s A(s)$ are all those actions that can be performed in function of the state. The central problem of an MDP is to identify the best action to perform in each state to obtain the maximum

possible value of a cumulative reward function. The function for each state s ∈ S identifies the action a ∈ A to be applied is called stationary "policy".

In this MDP there are four actions the agent can take, as:

- Right: The agent moves to the cell to the right of his current position if it is not blocked by a wall.
- Left: The agent moves to the cell to the left of his current position if it is not blocked by a wall.
- Above: The agent moves to the cell above the cell they are currently in if it is not blocked by a wall.
- Below: The agent moves to the cell below the cell they are currently in if it is not blocked by a wall.

The agent can move to any of the adjacent cells that are not blocked by walls and automatically acquires the cooking tool and start cooking as soon as he enters the cell with the tool and stove. By automatically acquiring the cooking tool and starting to cook as soon as you enter the cell with the stove, you don't need any further actions such as picking up the tool or starting to cook.

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

The transition probability P defines the one-step dynamics of the environment, i.e. the probability that, given a state S and an action A at time T, the next possible state is reached. Its dimensionality is given by the number of elements in the domain and the range of the transition function.

So being that the number of states is 36 and the number of shares is 4 -> S*S*A = States*States*Actions = 36*36*4 = 5.184.

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

The transition function P is represented through a table, where each entry corresponds to the transition probability from one state to another for a given action.

For example, a possible table would look like this:

Initial state (s)	**Next state (s')**	**Action**	**Transition Function P**
s(3,3)	s'(2,3)	Left	1
Initial state (s)	**Next state (s')**	**Action**	**Transition Function P**
s(3,3)	s'(4,3)	Right	1

All tables are in the file Excel PartA_PointD_AI_NollettiMartina.xlsx in the spreadsheets "Action Left", "Action Right", "Action Above" and "Action Below".

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

The reward function defines the reward (positive or negative) received by the agent: it is the probability of obtaining a reward for having passed from the state to have performed the action.

It can be defined as follows:

• When the agent reaches the target state (where there is either a frying pan or an oven), the reward is +10.

• When the agent picks up the cooking tool, the reward is +1.

• When the agent hits the wall, the reward is -150.

• In all other cases, the reward is -0.5.

Giving a higher reward to the agent for getting the cooking tool might slow him down as he would focus on getting the cooking tool rather than reaching the cells to cook. Setting the reward to 5 will encourage the agent to get the tools as soon as possible, and then focus on cooking.

The variable $\gamma \in [0, 1]$ is the discount factor. The intuition behind using a discount is that there is no certainty about future rewards. While it is important to consider future rewards to increase performance, it is equally important to limit the contribution of future rewards to performance (one cannot be 100% sure about the future). A high gamma value (close to 1) will give more importance to long-term rewards, while a low gamma value (close to 0) will give more importance to immediate rewards.

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

As mentioned in the previous question, a gamma value closer to 1 would make the agent explore more, trying to find the optimal path and thus finding the optimal policy faster as it gives more weight to future rewards. While a gamma value closer to 0 would lead the agent to focus on immediate rewards as soon as possible, but might not find the optimal policy. Therefore, the gamma value can influence the optimal policy.

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

A policy, π, is a mapping from states to actions and defines the action the agent should take in each state. It is defined as a function π: S -> A where S are the states and A are the actions. Having in this case 36 states and 4 actions, the possible policies are: $|A|^{|S|} = 4^{36}$.

h) Now, considering the problem as a model-free scenario, provide a program (written in python) able to compute the optimal policy for this world considering the pudding eggs scenario solely. Draw the computed policy in the grid by putting in each cell the optimal action. If multiple actions are possible, include the probability of each arrow. There may be multiple optimal policies, pick one to show it. Note that the model is not available for computation but must be encoded to be used in the "real world" environment.

The program is in the folder ChefProgram_Homework3_NollettiMartina.

i) Is the computed policy deterministic or stochastic?

Using the epsilon-greedy algorithm the policy is stochastic. The Epsilon-Greedy algorithm uses the exploration-exploitation trade-off of instructing the agent to explore (i.e. choose a random option with epsilon probability) and exploit (i.e. choose the option that so far seems to be the best) the rest of the time. This way, as time passes and the agent chooses different options, he will have an idea of which choices return him the highest reward. However, he occasionally chooses a random action just to ensure nothing is missing. Using this learning algorithm, the agent can converge toward the optimal strategy for whatever situation she is trying to learn.

In the presented program, the agent chooses the stock with the highest value with probability (1 - epsilon) and chooses a random stock with probability epsilon.

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

A stochastic policy will select action according to a learned probability distribution. The advantages of having a stochastic policy are as follows:

- Exploration: One of the main benefits of a stochastic policy is to allow the agent to explore the state space more efficiently, which can improve the agent's performance over time.
- Adaptive: using a stochastic policy, an agent can react and adapt to the different changes that may arise, to find the best solution.

- Avoid local optimums: A deterministic policy that always takes the best action for a state can lead the agent to a local optimum. By allowing the agent to take suboptimal actions with some probability, a stochastic policy can help the agent find the global optimum.
- Robustness: A stochastic policy can make the agent more robust, as it allows the agent to adapt to changes in the environment.

Part b.

Now consider that your agent, because of his tiredness, might go in the wrong direction. Then each action has a 60% chance of going in the chosen direction and a 40% chance of going perpendicular to the right of the direction chosen. Accordingly, with these new settings, answer the following questions:

a) Report the transition function P for any state s and action a ∈ A.

The transition function P can be defined as:

- 0.6, if the agent moves in the expected direction.
- 0.4, if the agent moves in a direction perpendicular to the right of the predicted direction.
- 1, if the agent is in a state with equal probability of success and failure.

All tables are in the file Excel PartB_PointA_MartinanNolletti.xlsx in the spreadsheets “Action Left”, “Action Right”, “Action Above” and “Action Below”.

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

The optimal policy may change as the agent now has a probability of failure and therefore the outcome of an action is uncertain.

This means that the agent may make different decisions and then take different actions than when you have accurate information. Thus, the optimal policy will differ in that the agent will also need to account for other paths to avoid failure, such as getting stuck in some states to ensure the right direction or getting stuck in a suboptimal policy.

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

Compared to the previous scenario, in which the agent had certain and perfect information, he now has a probability of failure as the uncertainty in the outcome of the actions can influence the rewards. So, the value of the optimal policy may be lower than the previous one.

Furthermore, it may also change because the agent may have to spend more time in some states to see if they are moving in the right direction. Thus, the value of the optimal policy may change if an agent faces uncertainty about the outcome of her actions, and expected rewards may be lower due to the possibility of spending more time on some stats or not reaching the target state.