[DT0171] - Artificial Intelligence Reinforcement Learning Module A.A.2022/2023

Gianluca Rea
gianluca.rea@student.univaq.it
278722

Part A

A) Provide a concise description of the states of the MDP. How many states are in this MDP?

S is the set of states in which the agent can be (Figure 1).
The agent's environment can be divided for simplicity into four different environments as shown in the table below.

Environments	Purpose	Having the Whisk
1	Scramble eggs	No
2	Scramble eggs	Yes
3	Pudding eggs	No
4	Pudding eggs	Yes

More in detail the state in S can be divided into:

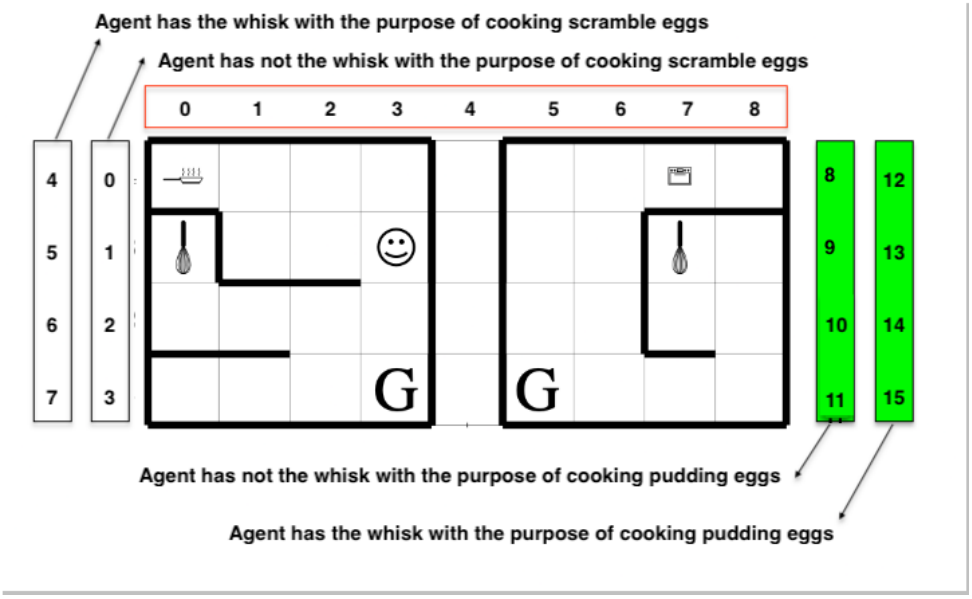

Figure 1 – Agent's Environment

a. States in which the agent does not have the whisk (f = free) and must cook scramble eggs (s = scramble).
$S_{fs} = \{(i,j) | \, i \in [0,3], \, j \in [0,8], \, i,j \in \mathbb{N}|\} \setminus \{S_{0,4}, \, S_{1,4}, \, S_{2,4}, S_{3,4}\}$

b. States in which the agent does have the whisk (w = whisk) and must cook scramble eggs (s = scramble)
$S_{ws} = \{(i,j) | \, i \in [4,7], \, j \in [0,8], \, i,j \in \mathbb{N}|\} \setminus \{S_{4,4}, \, S_{5,4}, \, S_{6,4} \, , S_{7,4}\}$

c. States in which the agent does not have the whisk (f = free) and must cook pudding eggs (p = pudding).
$S_{fp} = \{(i,j) | \, i \in [8,11], \, j \in [0,8], \, i,j \in \mathbb{N}|\} \setminus \{S_{8,4}, \, S_{9,4}, \, S_{10,4} \, , S_{11,4}\}$

d. States in which the agent does have the whisk (w = whisk) and must cook pudding eggs (p = pudding)
$S_{wp} = \{(i,j) | \, i \in [12,15], \, j \in [0,8], \, i,j \in \mathbb{N}|\} \setminus \{S_{12,4}, \, S_{13,4}, \, S_{14,4} \, , S_{15,4}\}$

The final states are $S_{4,0}$, $S_{12,7}$ where there are respectively the pan and the oven.
So we can calculate the dimensions of S as:$|S| = |S_{fs}| + |S_{ws}| + |S_{fp}| + |S_{wp}| = 32 + 32 + 32 + 32 = 128$

B) Provide a concise description of the actions of the MDP. How many actions are in this MDP?

The set of actions that the agent can perform, which are 7 in total and includes: Right, Left, Up, Down, Pick up the whisk, Travel left to right, Travel right to left . If the agent takes the wall with an action, he remains in the same position. So $|A| = 7$

In every state the agent can take the action Right, Left, Up, Down. Only in the state $S_{1,0}$, $S_{1,7}$, $S_{9,0}$, $S_{9,7}$ is possible to pick up the whisk and from this state the agent goes into the state where he has the whisk $S_{5,0}$, $S_{5,7}$, $S_{13,0}$, $S_{13,7}$. Also, the Travel action can be taken only on the 2 sections of the map but with multiple state results as shown in the table below. Notice that the travel can be done from left to right and vice versa.

Left	Right
$S_{3,3}$	$S_{3,5}$
$S_{7,3}$	$S_{7,5}$
$S_{11,3}$	$S_{11,5}$
$S_{15,3}$	$S_{15,5}$

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

The transition function P defined as $P : S \times S \times A \longrightarrow [0,1]$ will have dimension $|P| : 128 \times 128 \times 7$

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

The table can be found in the file "Function_P_PartA.xlsx" . Note that on this chart the two actions representing Travel (left to right) and Travel (right to left) are represented by the same table.

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

We describe a reward function $R: S \times A \times S \longrightarrow \mathbb{R}$ with the goal of arriving at the oven/pan after we collected a whisk. The goal of the agent can be translated into a maximization reward function with the minimum path.
The best solution is to assign to

$$R(s,a,s') = \begin{cases} 10 \ if \ ((4,1),L,(4,0)) \ or \ ((12,6),R,(12,7)) \ or \ ((12,8),L,(12,7)) \\ -1 \ o/w \end{cases}$$

Several tests within the "policy_eval.ipynb" notebook were carried out. For each gamma value, the mdp converges to an optimal policy for that value. What changes, besides obviously the value functions, are the convergence times. In any case, inside the notebook, it is verified that a good policy is obtained for gamma values between 0.8 and 0.9.

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

The discount factor $\gamma \epsilon (0,1)$ determines how much the agent considers future and present rewards. A number close to zero means that the agent only considers immediate rewards and therefore can be defined as "short-sighted", therefore he will only learn from actions that make an immediate profit. A value close to one, on the other hand, will cause him to value her actions on the total of all future earnings.

In our case, the gamma value significantly influences the optimal policy a value lower than 0.8 will cause the agent in some cases to take longer routes to reach the whisk and then go to the oven/pan depending on what to cook. A gamma value between 0.8 and 0.9, on the other hand, will cause the agent to always choose the shortest path to reach the whisk and then the oven/pan.
Gamma also influences the convergence time of the mdp, in fact, the higher its value, the higher the convergence time will be.

G) How many possible policy are there?

The number of possible policies is given by the number of possible combinations of state actions, ie: $|A|^{|S|}$ and since $|A| = 7$ and $|S| = 128$, we will have 7^{128} possible policies.

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 as the "real-world" environment.

The notebook realized is the "mf.ipynb". In this notebook, the first-visit Montecarlo method was implemented. The method can learn directly from *experience* or *episodes* rather than relying on prior knowledge of the environment dynamics.

Initialize:

$\pi \leftarrow$ policy to be evaluated

$V \leftarrow$ an arbitrary state-value function

$Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Repeat forever:

(a) Generate an episode using π

(b) For each state s appearing in the episode:

$R \leftarrow$ return following the first occurrence of s

Append R to $Returns(s)$

$V(s) \leftarrow \text{average}(Returns(s))$

First-visit Montecarlo method

The value function obtained with $\gamma = 0.9$ and 1000 iterations is shown in the image belove (Figure 2).

	0	1	2	3	4	5	6	7	8
0	-9.999910	-9.999749	-9.999221	-9.999477	0.0	-9.999850	-9.999868	-9.999975	-9.999992
1	-9.966693	-9.999550	-9.998592	-9.994750	0.0	-9.999765	-9.999916	-9.978059	-9.991256
2	-9.975593	-9.984765	-9.987968	-9.987378	0.0	-9.999583	-9.999740	-9.990582	-9.990443
3	-9.999556	-9.999096	-9.997138	-9.993686	0.0	-9.996363	-9.997433	-9.997156	-9.996417

	0	1	2	3	4	5	6	7	8
4	0.000000	-4.567307	-7.279727	-8.409243	0.0	-7.262483	-5.292164	0.000000	-1.000000
5	-9.930941	-6.798296	-7.939670	-8.704064	0.0	-8.069324	-7.563649	-9.948296	-9.919345
6	-9.902228	-9.788748	-9.657040	-9.361983	0.0	-8.776021	-8.709291	-9.947174	-9.902599
7	-9.902419	-9.834092	-9.688173	-9.518185	0.0	-9.272147	-9.296197	-9.618801	-9.791249

Figure 2 – Value Function

The next image shows the policy of the agent.

Figure 3 – Agent Policy

A – This is the environment in which agent does not have the whisk.
B – This is the environment in which agent does have the whisk.

I) Is the computed policy deterministic or stochastic?

For each action, the agent always makes the same transition, therefore the policy is deterministic.

J) Is the any advantage to having a stochastic policy? Explain

Having a deterministic environment makes using a probabilistic policy unnecessary. There are two cases where the probabilistic policy outclasses the determinist:

1. Stochastic Environment
   A deterministic policy will always opt for the same action, as it learns a unique deterministic state-to-action mapping. A stochastic policy instead will select an action according to a learned probability distribution.

2. Partially observable states
   There are cases where a part of the state is hidden from the agent, a stochastic policy is preferable as it can naturally act around uncertainty and infer the hidden states.

A deterministic policy can be considered as a subset of the stochastic policy when the states are fully observable e mostly deterministic

Part B

Introduction

Now consider that our agent often goes the wrong direction because of how tired it is. Now each action has a 60% chance of going perpendicular to the left of the direction chosen and 40% 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 tabular format

The table can be found in the file "Function_P_PartB.xlsx" . Note that on this chart the two actions representing Travel (left to right) and Travel (right to left) are represented by the same table. Also the change in value caused by the tiredness of the agent impact only the Up, Down, Left, and Right actions. The take the whisk and travels actions do not mutate from Part A.

B) Does the optimal policy change compared to part A? Justify the answer

In this case, there will be a massive impact on the optimal policy caused by the replacement of discrete actions with stochastic ones. Another key fact is that the action taken about moving will always be wrong.

C) Will the value of the optimal policy change? Explain how

Looking at the value function below

$$V(s) \leftarrow \sum_{a} \pi(a|\mathrm{s}) \sum_{s',r} p(s',r|\mathrm{s},\mathrm{a})[r + \gamma V(s')]$$

we can see that the $p(s',r|\mathrm{s},\mathrm{a})$ which describe the probability to end up in state s' and receive reward r starting in state s and selecting the action a a will change. In this first part we had that the probability to end in state s' starting from s was always 1. In this part the agent always makes a mistake having the 60% chance to go perpendicular to the left and 40% change to go perpendicular to the right of the actual direction chosen.
This makes the value of the optimal policy change.