UNIVERSITÀ DEGLI STUDI DELL'AQUILA

Artificial Intelligence

AI Project

Author:
Anusha Annengala

Professor:
Giovanni Stilo

Thursday 8th February, 2024

1 Assignment Questions:

- Report the transition function P for any state s and action a in a tabular format.
- Describe a reward function R : S × A × S and a value of that will lead to an optimal policy
- Now, considering the problem as a model-free scenario, provide a program (written in Python, possibly based on the labs) that can compute the optimal policy for this world by solely considering the pudding eggs scenario. Draw the computed policy in the grid by putting the optimal action in each cell. 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.

2 Introduction:

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 eggs recipe according to your indication (scrambled or pudding). 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 or the oven on the map). Finally, he can cook. Not that there are two special interlinked cells (marked with the G) that allow the agent to go from one side of the map to the other. But to do so, the agent needs to express his will to go on the other side. Cells in (4, 2) and (9, 3) are the special gate ones. They allow the agent to go from one side of the map to another. Those two special cells are interlinked, but the agent needs to express his will to go on the other side. Since you are very hungry, it is fundamental that the agent cooks the eggs according to your taste (scrambled/pudding) as fast as he can without letting you wait for more than necessary.

3 Implementation

3.1 Question 1

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

To report the transition function P for any state s and action a, we need to specify the probability of transitioning to each possible next state given the current state and action. Here's a tabular format for the transition function P(s'—s,a) where s' represents the next state

Current state (s)	Action (a)	Next state (s')

For the cooking chef problem with scrambled eggs, it's difficult to directly compute these probabilities without knowledge of the environment dynamics. However, we can make some assumptions to estimate them. Let's assume that the agent moves deterministically unless it encounters a gate cell or a wall.

Here's one example of how the transition function might look like for a specific state and action:

Current state (s)	Action (a)	Next state (s')
(3,3)	Left	(3,2)
(3,3)	Right	(3,4)
(3,3)	Up	(2,3)
(3,3)	Down	(4,3)

3.2 Question 2

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

To design a reward function R: S X A X S and a value of gamma () that will lead to an optimal policy, we need to consider the goals and constraints of the cooking chef problem with scrambled eggs.

In this scenario:

- The agent's goal is to cook scrambled eggs.
- The agent receives a reward for each step it takes, encouraging it to reach the goal efficiently.
- Additional rewards are given for reaching certain states, such as collecting the egg beater or reaching the frying pan.

Proposed reward function:

- Reward of -1 for each step taken by the agent.
- Reward of +10 for reaching a cell containing the egg beater (to encourage collecting the necessary tools).
- Reward of +100 for reaching a cell containing the frying pan (to encourage reaching the goal of cooking scrambled eggs).

The value of should be chosen to balance immediate rewards with future rewards. Since the agent's main goal is to cook scrambled eggs quickly, a high discount factor would prioritize immediate rewards. Therefore, a value of =0.9 could be a reasonable choice to give some importance to future rewards while still prioritizing immediate rewards.

So, the reward function R can be summarized as follows:

- R(s,a,s') = 1 for all s,a,s' (reward for each step)
- R(s,a,s') = +10 if s' contains the egg beater
- R(s,a,s') = +100 if s' contains the frying pan

This reward function, combined with a discount factor =0.9, would help guide the agent towards an optimal policy where it efficiently collects the necessary tools and reaches the goal of cooking scrambled eggs.

3.3 Question 3

3. Now, considering the problem as a model-free scenario, provide a program (written in Python, possibly based on the labs) that can compute the optimal policy for this world by solely considering the pudding eggs scenario. Draw the computed policy in the grid by putting the optimal action in each cell. 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.

To solve the cooking chef problem with scrambled eggs using a model-free approach, I have used Q-learning algorithm and attached the python notebook that computes the optimal policy.

The program initializes the Q-values for each state-action pair and then uses the Q-learning algorithm to update these values based on rewards obtained during exploration. Finally, it prints the optimal policy where each cell indicates the action to take. The symbols used are '←', '→', '↑', '↓' for left, right, up, down actions respectively, 'G' for gate cells, 'B' for egg beater cells, and 'F' for goal cells.