**[DT0171] – Final Exam**

**Reinforcement Learning Part**

**a.a.2023/2024   
Alessandro Pio n.294417**

1. **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:
2. **Provide a concise description of the states of the MDP. How many states are in this MDP?(i.e. what is |S|).**

In a Markov Decision Process (MDP), each state (S) is associated with a set of possible actions (A and B) that a rational agent can choose. Executing an action leads to a transition to a different future state (S' and S"). There are two distinct models for these transitions:

1. Deterministic Model: In this model, each action has a predetermined outcome, leading with 100% certainty to a specific subsequent state. However, there is an exception: each action (A) only has a probability of successfully reaching the desired state (S'), and there is also the intrinsic risk that the action may lead to an unintended state (Sx).
2. Probabilistic Model: Designed to handle decisions under conditions of uncertainty, it introduces an additional level of complexity. In addition to the probability of successfully reaching the goal state (S'), there is also consideration of the risk associated with the potential transition to a worse state (Sx).

The set of states *S* in this Markov Decision Process (MDP) represents a chef's possible positions within a grid-like world. So we can have 32 states without the beater and 32 states with the beater so a total of 64 states.

Therefore the cardinality ∣*S*∣ = 64

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

The actions, denoted as A, encompass all feasible actions based on the current state in this Markov Decision Process (MDP). The main challenge in an MDP lies in determining the optimal action to take in each state to maximize the cumulative reward function. The function prescribing the action to take in a specific state is defined as a "policy," and in this context, it is stationary. Within this MDP, there are seven distinct actions available to the agent (assuming the agent makes moves avoiding obstacles like walls):

* R: The agent moves to the cell to the right of its current position.
* L: The agent moves to the cell to the left of its current position.
* U: The agent moves to the cell above its current position.
* D: The agent moves to the cell below its current position.
* TL: The agent teleports to the left grid.
* TR: The agent teleports to the right grid.
* T: The agent picks up the whisk.

Therefore, based on this, we can say that |A| = 7.

1. **What is the dimensionality of the transition function P?**

The transition probability, denoted as P, describes the one-step dynamics of the environment. It specifies the probability that, given a particular state (S) and an action (A) taken at time T, the system transitions to a specific subsequent state.

As previously mentioned, the number of states in the problem is ∣S∣ = 64, and the number of actions is ∣A∣ = 7. Consequently, the dimensionality of the transition function P will be 64x64x7, indicating that for each state s and each action a, there is a probability of transition to each of the 64 possible states.

Therefore, P is a 64x64x7 matrix, where the rows correspond to the starting states, the columns correspond to the resulting states, and the third dimension corresponds to the action.

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

The tables are located within the file 'AlessandroPio\_PartA.xlsx'. The states are represented as followsImmagine che contiene Carattere, schermata, testo, linea

Descrizione generata automaticamente

Within the provided tables, there is no distinction made for the set of states with or without the egg beater, as the actions remain constant in the two different situations. The only instance where the table related to the "take" action is not valid is when the agent already possesses the egg beater.

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

We describe a reward function 𝑅 : 𝑆 × 𝐴 × 𝑆 ⟶ R where we aim to encourage the agent to pick up the egg beater before heading to the cooking area (oven or pan) and to proceed with cooking in the fewest steps possible.

In the specific case of our chef, we want the agent to prioritize picking up the egg beater before heading to the cooking area. To incentivize this behavior, we assign negative values to all states except those where the egg beater is located, a positive value for those states, and a higher positive value for the final state (depending on the recipe we intend to prepare) when the chef reaches the cooking area with the tool.

Finally, the gamma value allows us to balance the trade-off between short-term and long-term rewards by giving more or less weight to rewards obtained in the future. An optimal value to promote fast convergence of the problem could be 0.8.

1. **Does γ ∈ (0, 1) affect the optimal policy in this case? Explain why.**

The discount factor γ, a variable within the range [0, 1], significantly impacts the agent's strategy, carefully balancing the exploration of potential long-term benefits and the pursuit of immediate rewards. A higher γ value, approaching 1, gives more weight to long-term rewards, encouraging the agent to explore and identify optimal paths. This heightened consideration for future gains facilitates the rapid identification of an optimal policy.

In contrast, a lower γ value, approaching 0, emphasizes the importance of immediate rewards. While this approach prompts the agent to prioritize quick gains in the short term, it does not guarantee the discovery of the optimal policy. In our context, the gamma value exerts a significant influence on the optimal policy.

A value lower than 0.8 might lead the agent, in some situations, to choose longer routes to reach the whisk and then go to the oven or pan, depending on what needs to be cooked. On the other hand, a gamma value between 0.8 and 0.9 will ensure that the agent consistently chooses the shortest path to reach the whisk and then the oven or pan.

It's important to note that the gamma value also influences the convergence time of the MDP. Specifically, the higher its value, the longer the convergence time will be.

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

A policy (denoted by the symbol π) acts as a guide for the agent, mapping states to actions and specifying the optimal action to take in each state.

Mathematically, it is expressed as a function π: S -> A, where S represents the set of states and A represents the set of actions. In the specific context of this scenario, where there are 64 states and 7 possible actions, the total number of potential policies can be calculated as |A||S| = 7^64.

1. **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.**   
     
   The program is called ‘CookingChefProblem\_AlessandroPio’

Moreover, the code has been run different times and, at the end, the policies obtained were always the same, shown inside the notebook

1. **Is the computed policy deterministic or stochastic?**

The agent decides which action to take following an epsilon-greedy strategy: with a probability of epsilon, it will explore a random action, while with a probability of (1 - epsilon), it will exploit the acquired knowledge to choose an action.

The Q-values associated with the selected action are then updated using the Q-learning formula, considering both the immediate reward and the estimated future rewards.

Therefore, this algorithm introduces an element of randomness (stochastic) into the adaptive learning strategy, allowing the agent to refine its strategy for the optimal course of action in various situations.

1. **Is there any advantage to having a stochastic policy? Explain.**

Resorting to a stochastic policy provides several advantages in Reinforcement Learning, especially in contexts requiring efficient exploration of the state space and adaptability to dynamic changes in the environment.

This strategy allows the agent to avoid local optima, maintain robust decision-making capabilities in uncertain conditions, and successfully handle stochastic scenarios. In a deterministic environment, the use of a probabilistic policy may seem unnecessary, but it proves particularly effective in stochastic environments and with partially observable states.

The flexibility of a stochastic policy becomes evident when the agent can select actions based on learned probability distributions, ensuring better uncertainty management and the ability to adapt to changing conditions. Moreover, this approach addresses the limitations of deterministic policies, which may be impractical in scenarios where part of the state is hidden from the agent. Therefore, a stochastic policy emerges as a more suitable and versatile choice for tackling the challenges of complex machine learning contexts.

1. **Part B**Now consider that your agent might go in the wrong direction because of his tiredness. Then, each action has a 50% chance of going in the chosen direction and 50% chance of going perpendicular to the right of the direction chosen. Accordingly, with these new settings, answer the following questions:
2. **Report the transition function P for any state s and action a ∈ A.**The transition function, denoted as P, is characterized by the following probabilities:
3. If the agent moves in the anticipated direction, the probability is 0.5.
4. If the agent moves in a direction perpendicular to the right of the predicted direction, the probability is also 0.5.
5. If the agent is in a state where success and failure are equally likely, the probability is 1

These probabilities are outlined in tables located within the Excel file named ‘AlessandroPio\_PartB.xlsx’ In particular, this file contains tables with relative informations called "Left," "Right," "Up," e "Down."

1. **Does the optimal policy change compared to Part a? Justify your answer.**The introduction of a probability of failure adds uncertainty to the outcome of agent actions, potentially altering the optimal policy.

This uncertainty implies that the agent might make different decisions and take different actions compared to scenarios with precise information. Consequently, the optimal policy undergoes changes, requiring the agent to consider alternative paths to mitigate the risk of failure.

This may involve strategies such as intentionally getting stuck in certain states to ensure a correct direction or accepting a suboptimal policy to avoid potential failures.

1. **Will the value of the optimal policy change? Explain how.**Unlike the previous situation characterized by the agent's certainty of information, the current scenario introduces a probability of error, bringing uncertainty that can influence action outcomes and subsequent rewards. This may result in a reduction in the value associated with the optimal policy compared to the previous certainty-driven situation. Additionally, this change in the value of the optimal policy is driven by the agent's need to allocate more time to specific states to verify if it is moving in the correct direction. Consequently, the agent's optimal policy may undergo changes when facing uncertainties about action outcomes. Expected rewards could potentially decrease due to the likelihood of spending more time in certain states or failing to reach the desired target state.