

# TD1

## Artificial Intelligence - Introduction to Reinforcement Learning ENSISA 2A

Ali El Hadi ISMAIL FAWAZ

November 3, 2025



Une école d'ingénieurs de l'Université de Haute-Alsace

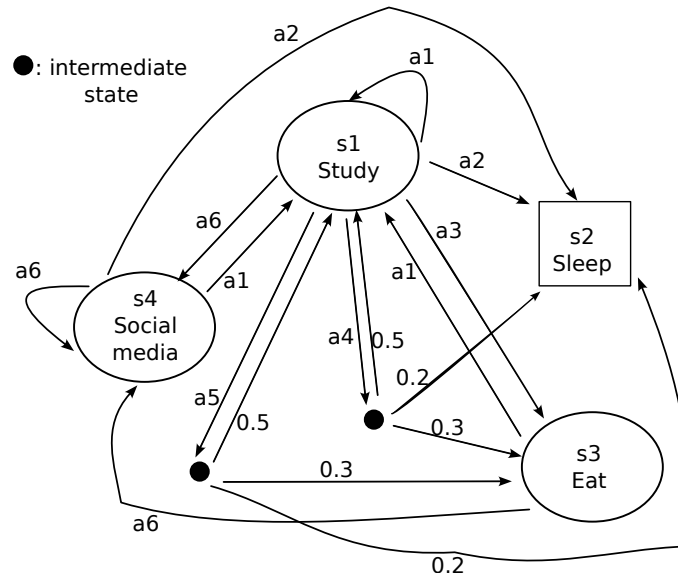


## Problem I: Covid Lockdown MDP

Consider the state-transition diagram of your Covid lockdown routine that includes 4 states:

- State  $s_1$ : Study
- State  $s_2$ : Sleep (terminal state)
- State  $s_3$ : Eat
- State  $s_4$ : Social Media

Here is a figure representing the MDP:



For this same MDP, the following actions are possible to take with their corresponding reward:

- Action  $a_1$ : Study, reward = +1
- Action  $a_2$ : Sleep, reward = +2
- Action  $a_3$ : Eat, reward = +1
- Action  $a_4$ : Sport, reward = +1
- Action  $a_5$ : Chess, reward = +1
- Action  $a_6$ : Sleep, reward = -1

Example:

If you are at state  $s_1$ , you can do the following:

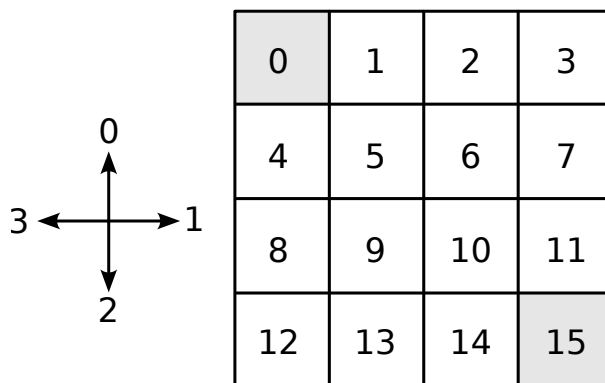
- Take action  $a_2$  and get to state  $s_2$  with a transition probability  $P_{s_1 s_2}^{a_2} = 1$  and get a reward  $R_{s_1}^{a_2} = +2$
- Take action  $a_5$ , get reward  $R_{s_1}^{a_5} = +1$  and go to:
  - state  $s_1$  again with transition probability  $P_{s_1 s_1}^{a_5} = 0.5$
  - state  $s_3$  with transition probability  $P_{s_1 s_3}^{a_5} = 0.3$
  - state  $s_2$  with transition probability  $P_{s_1 s_2}^{a_5} = 0.2$
- ...

## Questions

1. For each action, define the state transition probability matrices
2. Recall the definition of the state-value function  $V^\pi(s)$  following a policy  $\pi$  for a discounted MDP
3. Recall the Bellman expectation equation for the state-value function
4. Assuming a uniform random policy  $\pi(a|s)$  that gives an equal probability for all actions **that could be taken** from state  $s$ . For instance, in the case of state  $s_1$ , given that all possible actions can be taken, then  $\pi(a_i|s_1) = 1/6 \ i \in \{1, 2, 3, 4, 5, 6\}$ . Given the initial state-value function  $V_1^\pi(s) = 2\forall s$ , apply one epoch of the policy evaluation to compute the new state-value function  $V_2^\pi(s)\forall s$ . Assume the discount factor  $\gamma = 1$ .
5. Apply one epoch of the policy improvement algorithm and in a greedy manner, compute the new policy  $\pi_2$ .
6. Recall the definition of the optimal state-value function  $V^*(s)$  of an MDP.
7. Recall the Bellman optimality equation for state-value function.
8. Starting with initial value function  $V_1(s) = 2\forall s$ , apply one epoch using value iteration to compute  $V_2(s)\forall s$ , use discount factor  $\gamma = 1$ .
9. Is your new value function optimal ? Why ?

## Problem II: Grid World

For this exercise, you will implement the two model-based dynamic programming algorithms: value iteration and policy iteration, to solve the Grid World problem. Consider having an  $n \times n$  grid, example in Figure below when  $n = 4$ .



The position of an agent can be from squares 1 to 14 and the grey squares are the terminal state. The goal of the agent is to find the most optimal way to get to one of the two terminal states.

The actions that can be taken by the agent are:

- UP== 0
- RIGHT== 1
- DOWN== 2
- LEFT== 3

If one action throws the agent outside the grid, then its considered to leave the agent in his current position.

A reward of  $-1$  is given to each step the agent takes until reaching a terminal state.

For this problem, consider the following parameters:

- Use  $\gamma$  discount factors of 1, 0.9 and 0.8
- The stopping criterion is defined as  $\max_s (|V_{new}(s) - V_{old}(s)|) \leq \theta$
- Use  $\theta = 10^{-4}$

## Questions

1. Implement the value iteration algorithm to solve such problem. Return the optimal policy and corresponding value function.
2. Implement the policy evaluation and policy improvement algorithms. Return the optimal policy and corresponding value function.