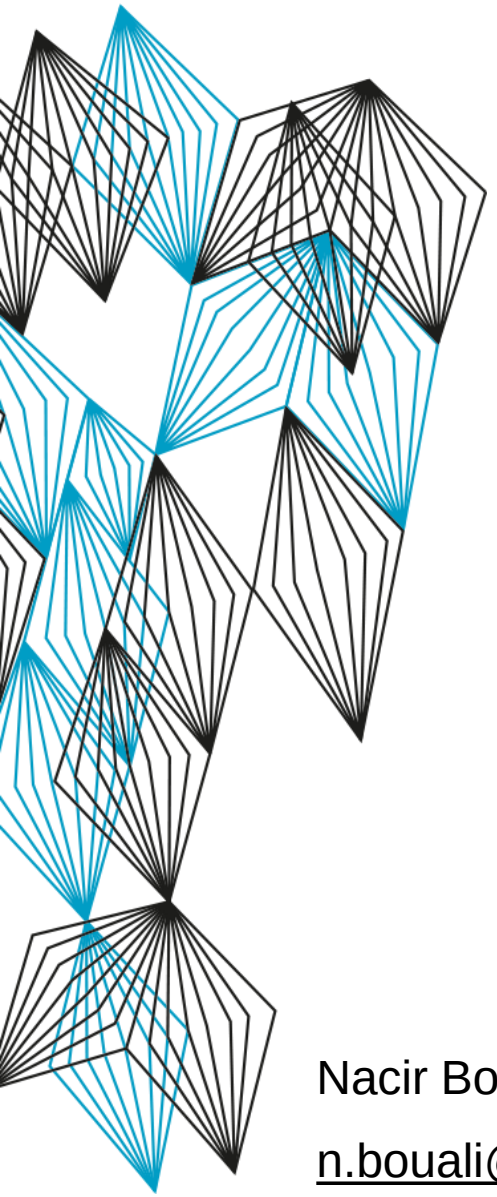

ARTIFICIAL INTELLIGENCE & CYBER SECURITY

REINFORCEMENT LEARNING

ADAPTIVE DYNAMIC PROGRAMMING

Nacir Bouali

n.bouali@utwente.nl

UNIVERSITY OF TWENTE.

REINFORCEMENT LEARNING

- The drawback of direct utility estimation is that it overlooks the recursive relation between the states.

$$U^{\pi}(s) = R(s) + \gamma \sum_{s'} p(s'|s, \pi(s))\, U^{\pi}(s')$$

- If trial 2 doesn't transition through state (3,2) then no information about that state is given.
- We could still deduce that (3,2) has a higher utility given its adjacency to a state with a high utility.

ADAPTIVE DYNAMIC PROGRAMMING

- In ADP, rather than learning the utilities of states, we want to learn the transition model P(s'|s,a).
- Then given P(s'|s,a), we can solve;

$$U^{\pi}(s) = R(s) + \gamma \sum_{s'} p(s'|s, \pi(s))\, U^{\pi}(s')$$

- For a fixed policy, this is a system of |S| equations with |S| unknowns.
- We can then solve it using numpy or any linear algebra library.

ADAPTIVE DYNAMIC PROGRAMMING

- We can estimate P(s'|s,a) by looking at the trials;

$$(1,1) \xrightarrow[Up]{-.04} (1,2) \xrightarrow[Up]{-.04} (1,3) \xrightarrow[Right]{-.04} (1,2) \xrightarrow[Up]{-.04} (1,3) \xrightarrow[Right]{-.04} (2,3) \xrightarrow[Right]{-.04} (3,3) \xrightarrow[Right]{+1} (4,3)$$

$$(1,1) \xrightarrow[Up]{-.04} (1,2) \xrightarrow[Up]{-.04} (1,3) \xrightarrow[Right]{-.04} (2,3) \xrightarrow[Right]{-.04} (3,3) \xrightarrow[Right]{-.04} (3,2) \xrightarrow[Up]{-.04} (3,3) \xrightarrow[Right]{+1} (4,3)$$

$$(1,1) \xrightarrow[Up]{-.04} (1,2) \xrightarrow[Up]{-.04} (1,3) \xrightarrow[Right]{-.04} (2,3) \xrightarrow[Right]{-.04} (3,3) \xrightarrow[Right]{-.04} (3,2) \xrightarrow[Up]{-1} (4,2)$$

- From state (1,3), executing the action « right » takes the agent to state (2,3) 3 out of 4 times.
  - $P(state_{(2,3)}|state_{(1,3)},action=right)=0.75$