Planning Lab - Lesson 3

Markov Decision Process (MDP)

Alessandro Farinelli
Thanks to Davide Corsi, Luca Marzari and Celeste Veronese for helping with slides and code

University of Verona
Department of Computer Science

May 24, 2024

UNIVERSITÀ
di VERONA
Dipartimento
di INFORMATICA

Start Your Working Environment

Update your repository to download the new lesson

Important: do a backup copy of your working directory to make sure you avoid any issue

```
> cd AI_Lab
> git commit -a -m "a message describing the commit"
> git pull
> conda activate ai-lab
> jupyter notebook
```

To open the assignment navigate with your browser to:
MDP/MDP_3_problem.ipynb

Assignments

In this lab session we will focus on MDP, your assignments as specified in: *MDP/MDP_3_problem.ipynb* are the following:

- You *must* implement the *value iteration* algorithm (*required*)
- You *can* implement the *policy iteration* algorithm (*optional*)

The notebook includes working code to test the algorithms in different environments In the following you can find the pseudocode for such algorithms

Value Iteration (REQUIRED)

function VALUE-ITERATION(mdp, ϵ) **returns** a utility function
  inputs: mdp, an MDP with states S, actions $A(s)$, transition model $P(s' \mid s, a)$,
    rewards $R(s)$, discount γ
    ϵ, the maximum error allowed in the utility of any state
  local variables: U, U', vectors of utilities for states in S, initially zero
    δ, the maximum change in the utility of any state in an iteration

  repeat
    $U \leftarrow U'$; $\delta \leftarrow 0$
    for each state s **in** S **do**
      $U'[s] \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\, U[s']$
      if $|U'[s] - U[s]| > \delta$ **then** $\delta \leftarrow |U'[s] - U[s]|$
  until $\delta < \epsilon(1-\gamma)/\gamma$
  return U

Policy Iteration (OPTIONAL)

function POLICY-ITERATION(mdp) **returns** a policy
  inputs: mdp, an MDP with states S, actions $A(s)$, transition model $P(s' \mid s, a)$
  local variables: U, a vector of utilities for states in S, initially zero
    π, a policy vector indexed by state, initially random

  repeat
    $U \leftarrow$ POLICY-EVALUATION(π, U, mdp)
    $unchanged? \leftarrow$ true
    for each state s **in** S **do**
      if $\max_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\ U[s'] > \sum_{s'} P(s' \mid s, \pi[s])\ U[s']$ **then do**
        $\pi[s] \leftarrow \operatorname{argmax}_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\ U[s']$
        $unchanged? \leftarrow$ false
  until $unchanged?$
  return π

To implement the *Policy-Evaluation* step, use the following formula:

$$U_i(s) = R(s) + \gamma \sum_{s'} P(s' \mid s, \pi_i(s)) U_i(s') \ .$$