AI Lab - Session 3

Markov Decision Process

Riccardo Sartea

University of Verona
Department of Computer Science

May 24^{th} 2019

UNIVERSITÀ
di VERONA
Dipartimento
di INFORMATICA

Start Your Working Environment

Start the previously installed (Session 1) conda environment *ai-lab*

Listing 1: Upgrade and spin up

```
cd ai-lab
git pull
conda remove --name myenv --all
conda env create -f ai-lab-environment.yml
conda activate ai-lab
jupyter notebook
```

NumPy

What is it

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- *a powerful N-dimensional array object*
- *sophisticated (broadcasting) functions*
- *tools for integrating C/C++ and Fortran code*
- *useful linear algebra, Fourier transform, and random number capabilities*

What is it for

Fast array manipulation and mathematical operations. Think of it as a MATLAB like environment for Python: try to speed up the computations writing code in a vectorial fashion.

Where to find it

http://www.numpy.org

Tutorial

To open the tutorial navigate with your browser to:
session3/session3_tutorial.ipynb

Assignments

- Your assignments for this session are at: *session3/session3_mdp.ipynb*. You will be required to implement value iteration and policy iteration algorithms
- In the following you can find pseudocodes for such algorithms

Value Iteration

```
Input: $environment\ [T, R, A, S]$, $\gamma, \delta, maxiters,$
Output: $policy$ - state/action mapping
$V \leftarrow [0, ..., 0]$                                    ▷ Null vector of length $|S|$
$iter \leftarrow 0$
repeat                                                        ▷ Compute Bellman Equation
    $V' \leftarrow V$
    $iter \leftarrow iter + 1$
    for each $s$ in $S$ do
        $V_s \leftarrow \max_{a \in A_s} \sum_{s' \in S} T(s, a, s')(R(s, a, s') + \gamma V_{s'})$
until $max(|V - V'|) < \delta$ or $iter = maxiters$
$\pi \leftarrow [0, ..., 0]$                                  ▷ Null vector of length $|S|$
for each $s$ in $S$ do                                        ▷ Extract policy
    $\pi_s \leftarrow \operatorname{argmax}_{a \in A_s} \sum_{s' \in S} T(s, a, s')(R(s, a, s') + \gamma V_{s'})$
return $\pi$
```

Policy Iteration

```
Input: $environment\ [T, R, A, S]$, $\gamma, \delta, vmaxiters, pmaxiters$,
Output: $policy$ - state/action mapping
$V \leftarrow [0, ..., 0]$                                        ▷ Null vector of length $|S|$
$\pi \leftarrow [0, ..., 0]$                                      ▷ Initial policy length $|S|$
$piter \leftarrow 0$
repeat
    $\pi' \leftarrow \pi$
    $piter \leftarrow piter + 1$
    $viter \leftarrow 0$
    repeat                                                        ▷ Evaluate Policy
        $V' \leftarrow V$
        $viter \leftarrow viter + 1$
        for each $s$ in $S$ do
            $V_s \leftarrow \sum_{s' \in S} T(s, \pi_s, s')(R(s, \pi_s, s') + \gamma V_{s'})$
    until $max(|V - V'|) < \delta$ or $viter = vmaxiters$
    for each $s$ in $S$ do                                        ▷ Improve policy
        $\pi_s \leftarrow \operatorname{argmax}_{a \in A_s} \sum_{s' \in S} T(s, a, s')(R(s, a, s') + \gamma V_{s'})$
until $\pi = \pi'$ or $piter = pmaxiters$
return $\pi$
```