Reinforcement Learning Lab

Lesson 4: Temporal Difference Methods

Davide Corsi and Alberto Castellini

University of Verona
email: davide.corsi@univr.it

Academic Year 2022-23

Environment Setup

The first step for the setup of the laboratory environment is to update the repository and load the miniconda environment.

- Update the repository of the lab:

```
cd RL-Lab
git stash
git pull
git stash pop
```

- Activate the *miniconda* environment:

```
conda activate rl-lab
```

Safe Procedure

Always back up the previous lessons' solutions before executing the repository update.

Today Assignment

In today's lesson, we implement the Q-Learning and SARSA algorithms in Python. In particular, the file to complete is:

```
RL-Lab/lessons/lesson_4_code.py
```

Inside the file, two functions are partially implemented. The objective of this lesson is to complete them.

- **def QLearning()**
- **def SARSA()**

Expected results can be found in:

```
RL-Lab/results/lesson_4_results.txt
```

Q-Learning

```
Require: environment $[A, S]$, problem, episodes, $\alpha$, $\gamma$, expl_func, expl_param
Ensure: policy, rewards, lengths
 1: $\forall a \in A, \forall s \in S$ initialize $Q(s, a)$ arbitrarily
 2: rewards, lengths $\leftarrow [0, ..., 0]$                         ▷ Null vectors of length episodes
 3: for $i \leftarrow 0$ to episodes do
 4:     Initialize $s$
 5:     repeat
 6:         $a \leftarrow$ EXPL_FUNC($Q, s$, expl_param)
 7:         $s', r \leftarrow$ take action $a$ from state $s$               ▷ Act and observe
 8:         $Q(s, a) \leftarrow Q(s, a) + \alpha(R + \gamma \max_{a' \in A_s} Q(s', a') - Q(s, a))$   ▷ TD
 9:         $s \leftarrow s'$
10:     until $s$ is terminal
11:     Update rewards, lengths
12: $\pi \leftarrow [0, ..., 0]$                                        ▷ Null vector of length $|S|$
13: for each $s$ in $S$ do                                             ▷ Extract policy
14:     $\pi_s \leftarrow \operatorname{argmax}_{a \in A_s} Q(s, a)$
15: return $\pi$, rewards, lengths
```

SARSA

```
Require: $environment\ [A, S], problem, episodes, \alpha, \gamma, expl\_func, expl\_param$
Ensure: $policy, rewards, lengths$
 1: $\forall a \in A, \forall s \in S$ initialize $Q(s, a)$ arbitrarily
 2: $rewards, lengths \leftarrow [0, ..., 0]$          ▷ Null vectors of length $episodes$
 3: for $i \leftarrow 0$ to $episodes$ do
 4:     Initialize $s$
 5:     $a \leftarrow \text{EXPL\_FUNC}(Q, s, expl\_param)$
 6:     repeat
 7:         $s', r \leftarrow$ take action $a$ from state $s$          ▷ Act and observe
 8:         $a' \leftarrow \text{EXPL\_FUNC}(Q, s', expl\_param)$
 9:         $Q(s, a) \leftarrow Q(s, a) + \alpha(R + \gamma Q(s', a') - Q(s, a))$          ▷ TD
10:         $s \leftarrow s'$
11:         $a \leftarrow a'$
12:     until $s$ is terminal
13:     Update $rewards, lengths$
14: $\pi \leftarrow [0, ..., 0]$          ▷ Null vector of length $|S|$
15: for each $s$ in $S$ do          ▷ Extract policy
16:     $\pi_s \leftarrow \underset{a \in A_s}{\text{argmax}}\ Q(s, a)$
```