University of the Witwatersrand, Johannesburg

School of Computer Science & Applied Mathematics

Artificial Intelligence (COMS4033A/7044A)
Reinforcement Learning

1 Introduction

In this lab, we'll compared the two temporal-difference learning algorithms, SARSA and Q-learning, on an environment. Fortunately, many libraries exist that provide a standard API between an agent and its environment—the most popular one is called `gymnasium`. The library is located here: `https://gymnasium.farama.org/`—please read the basic usage guide before continuing. Many of the most popular RL libraries are written in Python, so that will be our language of choice.

To install this Python library, run the following command:

```
pip install gymnasium
```

The environment we'll be tackling is called `CliffWalking`, where an agent must navigate from a start location to a goal while attempting to avoid falling of a cliff. A description of the environment can be found here: `https://gymnasium.farama.org/environments/toy_text/cliff_walking/` and you can create the environment as follows:

```python
import gymnasium as gym
env = gym.make('CliffWalking-v0')
```

Note that all `gymnasium` environments have a standard interface, as described in the basic usage documentation. You should make use of this interface inside your learning algorithms, so that when the agent selects an action, it then receives the next observation, reward, etc.

2 Q-Learning

Your first task is to implement the Q-learning algorithm and test it on the `CliffWalking` environment. Each episode should end when either the environment indicates it (through the `done` flag) or when 100 steps have elapsed. At the end of each episode, record the sum of rewards received for that episode.

When implementing the algorithm use an ϵ-greedy policy with $\epsilon = 0.1$, $\gamma = 0.99$ and $alpha = 0.1$.

3 SARSA

Once you have implemented Q-learning, implement SARSA use the same settings and parameters.

4 Submissions

For both algorithms, do the following. Run each algorithm for 1000 episodes. You should then have a list of size 1000, one for each episode return. Then, run the entire procedure 10 times in total and average the results over these 10 runs. This will leave you with a list of length 1000, where each entry at index i represents the average return received for episode i.

Plot these curves for both algorithms's results on the same axes, making sure to label each line with the relevant algorithm. Write a short discussion (3-4 sentences) analysing the results, focusing particularly on why you think the curves look different?

On Moodle, submit the following:

1. Your code
2. The plot
3. A text file with a brief discussion