MDPs, value iteration, policy iteration and q-learning

1 Introduction

In this assignment you will explore Markov Decision Processes, and algorithms to find the optimal policy within them. You will implement value iteration, policy iteration, and tabular Q-learning and apply these algorithms to simple environments including tabular maze navigation (FrozenLake) and controlling a simple crawler robot. For this assignment we will use Jupyter Notebook. For python notebook files the convention is to use the ".ipynb" extension. The problems are taken from Deep RL Bootcamp There are three notebook files

- **Lab 1 - Problem 1.ipynb**
- **Lab 1 - Problem 2.ipynb**
- **Lab 1 - Problem 3.ipynb**

In the FrozenLake environment, we know the environment dynamics, and you will use them to solve your problem. For the crawler robot (Problem 3) we only interact with the environment through sampling. As an extra challenge you may also try to solve Problem 1 and Problem 2 them by only sampling from the environment!

2 Setup

You will need the packages `matplotlib`, `numpy`, `jupyterlab`, and `gym` (here you need version 0.9.2).

2.1 Installation with pip

```
pip3 install matplotlib numpy jupyterlab gym==0.9.2
```

If installing using `--user`, you must add the user-level bin directory to your PATH environment variable in order to launch jupyter notebook, see Section 2.3.

2.2 Installation with anaconda

If you use anaconda, you could install what you need with

```
conda env create -f environment.yml
```

where **environment.yml** is in the same folder as your assignment.

2.3 Launching jupyter notebook

Open a terminal, navigate to the directory where your notebooks are, then run the command

```
jupyter notebook
```

Your browser should now open a tab where you may then continue by opening **Lab 1 - Problem 1.ipynb** (if it does not open a tab automatically you may navigate to the url it prints out).

3 Questions

3.1 Value iteration with unknown dynamics

In this Problem 1 you used the *state*-value function V, rather than *action*-value function in value iteration. If the environment was *unknown*, i.e. we could only learn about the environment through sampling, would you still use the state-value function? Why or why not?

3.1.1 Solution

Hard to find the maximum over actions when sampling, so the action-value function is preferable.