Deadline: 11.Mar.2022 at 23:59
Submission type: report PDF format
Where: Moodle - Exercise 2

Exercise 2

Reinforcement Learning State Representation

The objective of this exercise is to investigate and understand the effect of choosing different Markov Decision Process (MDP) state representations for reinforcement learning (RL).

Description

During the lectures, we've emphasized the need to correctly specify the MDP parameters for a given RL problem. That is, correctly defining the state space, the action space and the reward function such that the problem definition does not violate the Markov assumption. In this exercise, we'll use the baseline cart-pole problem to analyze the effects of choosing a non-Markovian state space for the RL problem to attempt to solve.

You're provided with a basic python script to implement tabular Q-learning for the cart-pole problem, which uses the OpenAI gym libraries (relies on Python 3.5 or higher, not yet fully supported on Python 3.9). The implementation provides two options for the problem state space. The first uses:

$$[x, x', \theta, \theta']$$

Where x is the cart position along the track, x' is the cart velocity, θ is the pole rotation and θ' is the pole's rotational velocity. The second option uses a reduced version:

$$[x,\theta]$$

For both versions, the state space is discretized along each dimension. There are two actions, either push the cart either to the left or to the right.

To run the script, first install the dependencies[1]:

```
cd path/to/Exercise2
pip install -r requirements.txt
```

The script also provides some basic plotting functions to aid your analysis.

Your tasks:

1. **(3.0) Analyze the baseline behavior**
   Train a policy with Q-learning using the baseline configuration by running:

   ```
   python cart_ctrl.py
   ```

[1] As a general rule we recommend doing this in a virtual environment, however, the install for this exercise is fairly light and only includes the following libraries: gym, matplotlib and argparse

Deadline: 11.Mar.2022 at 23:59
Submission type: report PDF format
Where: Moodle - Exercise 2

Exercise 2

Reinforcement Learning State Representation

Note that this will save the learned policy as policy0.pickle and will also save a plot of the mean system performance over episodes in training.pdf. If you would like to specify your own policy output file, use:

```
python cart_ctrl.py -train file_name.pickle
```

After training, you can observe the learned policy performance in the cart-pole domain by executing:

```
python cart_ctrl.py -test file_name.pickle
```

Inspect the training curve as well as the learned policy performance and explain the trends that you observe (min 100 words, max 250 words). Include relevant plots.

Note: The training is stochastic so each time you run it you will get different results. Sometimes you may even observe learning failure.

2. **(3.0) Compare the behavior when learning with the reduced state**
   Now run the script to train the Q-learning policy using the reduced state space definition.

```
python cart_ctrl.py -non_markov -train file_name.pickle
```

Compare the resulting training curve as well as the learned policy performance against what you obtained from question 1. Explain any differences you observe in terms of the choice of MDP state (min 100 words, max 250 words). Include relevant plots.

IMPORTANT NOTE: This is not meant to be a coding exercise and so you have also been provided with sample plots and learned policies for both state representations. You may use these for your analyses and comparisons. However, you are strongly encouraged to try it out for yourself to get familiar with the general setup of RL problems especially those using OpenAI gym environments.

Extra points (Optional):

(0.5 points) Compare to another state representation
Edit the python script to try a different state definition (e.g. [x',θ']) and repeat the analysis and comparison from question 2.

(0.2 points) Directly analyze the learned Q function
Use and edit the plot_Qslice function to visualize and compare the learned Q values at each of the state-actions for the Markovian and non-Markovian state definitions. The function is set up to plot x-θ slices of the Q function for any given x', θ' and action.