EE466000 Introduction to Reinforcement Learning
Homework 1: Multi-Armed Bandit

Due: April 4, 2021 23:59

Goal

The goal of this assignment helps you get familiar with basic action-value based methods in multi-armed bandit problems.

Todo

- Implement the algorithm:
  - ✓ ε-Greedy
- Get familiar with basic Python syntax.

Details

- Problem description
  - Implement a 6-armed bandit problem with $q_*(1) = 0.3, q_*(2) = -5, q_*(3) = 5, q_*(4) = -1.1, q_*(5) = 1, q_*(6) = 0$.
  - When a learning method applied to the problem, the actual reward, R_t, was given by a normal distribution with mean $q_*(A_t)$ and variance 1.

- File description
  - `hw1.ipynb`: Since it's the first homework, we will provide more instructions in this file, please follow the instructions to complete your homework.

  - The `bandit` environment is used in this assignment. You will implement normal distribution to randomly generate the reward function of each bandit. You should modify the `step` function in this class.

  - In the class `learning`, you will implement ε-Greedy action selection and update the action values. Please modify the `chooseAction` and `updateValue` function in this class to complete your homework.

  - We strongly recommend to implement evaluation function or plotting function by your own to get familiar with plotting mechanism in Python. We provide an basic plotting as your reference.

- After you've done all the algorithms, you should implement plotting function on your own to analyze different settings.

- Please write a **README** file to explain how to run your code if you implemented extra functions.

Requirements and Installation

- Python version: 3.7
- Please run `pip install [library_name]` to install necessary libraries.

Report

- **Title, name, student ID**

- **Implementation**
  - ✓ In ε-Greedy, how do you select action if the action values are equal?
  - ✓ Briefly describe your implementation.

- **Experiments and Analysis**
  - ✓ Get average rewards curves of different settings over 1000 steps and average the result of 30 learning process into a figure.
    - Vary ε value with 0, 0.2, 0.8. What happens? Why? Please plot the curves into a figure.
  - ✓ Is there any way to always get the best result when $\varepsilon = 0$? How?

Reminder

- Please upload your code main.py and report.pdf to iLMS before 4/4 (Sun) 23:59. No late submission allowed.
- DO NOT zip your code into a single file.
- Please do not copy&paste the code from your classmates.