EE551000 System Theory
Homework 2: Temporal-Difference Learning

Due: November 6, 2020 23:59

Goal

The goal of this assignment helps you understand TD(0) for prediction in Cliff-walking environment.

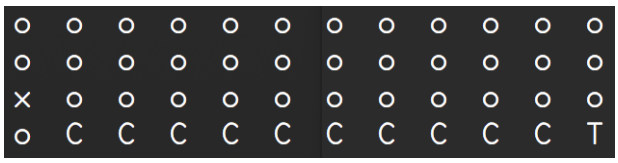
Todo

- Implement two algorithms:
  - ✓ On-policy: SARSA
  - ✓ Off-policy: Q-learning

Details

- File description
  - `env.py`: The cliff-walking environment used in this assignment. You should NOT modify this file.
  - `algo.py`: You'll implement two algorithms in the file. Please follow the instructions to complete your homework.
  - `utils.py`: Helper functions (such as plot) are implemented in this file. We strongly recommend to implement evaluation function or plotting function by your own in order to get familiar with plotting mechanism in Python. We provide an example plotting function as your reference.
  - `main.py`: main file for your implementation.

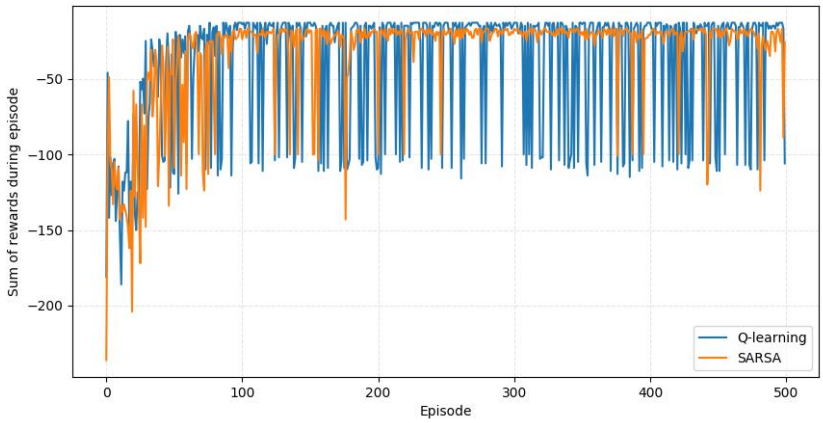
- Cliff-walking environment



  There are total 48 states and four discrete actions (up, down, right, left). "o" stands for each state, "x" for your agent, "C" for cliff and "T" for termination state. Each transition will get reward with -1. The agent gets -100 if it navigates to "C". The goal is to let your agent navigate to termination state with maximum reward.

- You can show how your learned agent acts after each method by running:
  `python main.py --algo [which_algo] -render`
  This allows you to visualize the trajectory to termination state.

- After you've done all the algorithms, you should implement plotting function on your own to analyze different settings. Or you can use the flag `--runAll` to show it out. For example (the figure format only, the result would be different):



- You are allowed to modify all the files except env.py. Please write a README file to explain how to run your code if you implemented extra functions.

Requirements and Installation

- Python version: 3.6
- Please run pip install –r requirements.txt to install necessary libraries.

Report

- **Title, name, student ID**
- **Implementation**
  - ✓ Briefly describe your implementation.
- **Experiments and Analysis**
  - ✓ Plot curves of different methods into a figure. (As example above)
  - ✓ Plot the episode length (time steps taken per episode) v.s. episode. What do you observe?
  - ✓ Render and show the trajectory of each method. What do you observe?
  - ✓ Observe the reward curve of each algorithm. We can observe that the reward curve of SRASA is more stable than Q-learning (less severe drop to -100). Please explain.
  - ✓ Why is Q-learning considered an off-policy control method? How about SARSA?
  - ✓ Vary the TD learning rate α, what happens?

Reminder

- Please upload your code and report.pdf to iLMS before 11/6 (Fri.) 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.
- Please write a README file to explain how to run your code if you implemented extra functions.