DDA 4230: Reinforcement learning **CUHKSZ**

Assignment 2

TA: Sheng Xu

Due Date: Oct. 28th, 11:59 pm
Total points available: 100 pts.

Note: *Please note that external references are allowed only if you give appropriate reference. There is no required format of reference. Please elaborate on your answers as well (do not just give a number, etc).*

Problem 1: Empirical MDP (20 points)

Suppose that we are given a generative model, which takes an input (s, a) and output the next state $s' \sim P(\cdot|s, a)$ and reward $r(s, a)$. We can have an empirical estimate of the transition kernel $\hat{P}$ through $\hat{P}(s'|s, a) = \frac{\text{count}(s,a,s')}{N}$, where N is the number of times we query the generative model with (s, a). Therefore, we query the generative model for $|\mathcal{S}||\mathcal{A}|N$ times in total.

Using this empirical estimate of transition kernel, we can obtain $\hat{Q}^\pi = r + \gamma \hat{P} V^\pi$. Recall that the optimal Q function is defined as $Q^* = \sup_\pi Q^\pi$. Define $P^\pi = P(s, \pi(s))$, show that for all π we have that:

$$Q^\pi - \widehat{Q}^\pi = \gamma \left(I - \gamma \widehat{P}^\pi\right)^{-1} (P - \widehat{P}) V^\pi$$

Problem 2: Practice of bandit problem (40 points)

Consider a two-armed bandit problem, where each arm's distribution is Bernoulli. Consider the following problem variants in Table 1, with respective Bernoulli distribution parameters specified for each arm:

Table 1: Table of Problem 2.

Problem	Arm 1	Arm 2
P1	0.9	0.6
P2	0.9	0.7
P3	0.6	0.4
P4	0.5	0.5

Write a Python program to simulate each of the above bandit problems. Specifically, for each problem, you need to do:

1. Choose the horizon n as 10000.
2. For each algorithm, repeat the experiment 100 times.
3. Store the number of times an algorithm plays the optimal arm, for each round $t = 1, \cdots, n$.
4. Store the regret in each round $t = 1, \cdots, n$.
5. Plot the percentage of optimal arm played and regret against the rounds $t = 1, \cdots, n$,
6. For each plot, add standard error bars.

Do the above for the following bandit algorithms:

1. UCB algorithm

Consider the UCB algorithm, which plays each arm once initially and then, in each round t, plays the arm I_t as follows:

$$I_t = \arg\max_{k=1,2} \hat{\mu}_k(t-1) + \sqrt{\frac{2\log t}{T_k(t-1)}}$$

2. Variant UCB algorithm

Consider a variant of the UCB algorithm, say UCB', where the horizon n is used in the confidence width as follows:

$$I_t = \arg\max_{k=1,2} \hat{\mu}_k(t-1) + \sqrt{\frac{2\log n}{T_k(t-1)}}$$

3. Explore-then-Commit (ETC) algorithm

Consider the explore-then-commit (ETC) algorithm with exploration parameter k chosen optimally so that the gap-dependent regret is minimum (this choice for k would require information about underlying gap).

4. Interpretation and summarization

Interpret the numerical results and submit your conclusions. In particular, discuss the following:

1. Discuss the difference between the two UCB algorithms, from both intuitions and experimental results.
2. Discuss the results obtained for ETC with optimal k and correlate the results to the theoretical findings.

Note that you need to submit: 1) Source code, preferably one in .ipynb that is readable with some comments; 2) Plots/tabulated results in a document; 3) Discussion of the results - either hand-written or typed-up.

Problem 3: Value and Policy Iteration Implementation (40 points)

In this problem, you are going to implement value and policy iteration and perform experiments with OpenAI Gym environment. Skeleton codes (in python) are provided through Google Colab (you can use it for free with the computation demands of this assignment), and here is the link to the skeleton code. You will need to **create a copy of this colab notebook in your own Google Drive**.

1. Implementation

Please implement both value and policy iteration. **Submit your code a share link from Google Drive. No mark will be rewarded if your code cannot be ran.**

2. Experiment

Please try to play around with the Frozen Lake environment (check the code link for details) with different configurations (gamma, improvement/evaluation iterations, random seed) and with value and policy iteration. Please try your best to get a policy with a score as high as possible. Repeat each experiment with 5 different random seeds, then report the mean and standard deviation results of the highest score you've achieved across the 5 repeats, and write a few sentences to compare the two iteration methods.