**535515 Spring 2023: Reinforcement Learning** **(Due: 2023/03/08, Wednesday, 21:00)**

# Homework 0: Fundamentals – MDPs, Policy Iteration, and Value Iteration

**Submission Guidelines**: Your deliverables shall consist of 2 separate files – (i) A PDF file: Please compile all your write-ups into one .pdf file (photos/scanned copies are acceptable; please make sure that the electronic files are of good quality and reader-friendly); (ii) A zip file: Please compress all your source code into one .zip file. Please submit your deliverables via E3.

**Problem 1 (Q-Value Iteration)** (20+20=40 points)

**(a)** Recall that in Lecture 4, we define $V_*(s) := \max_\pi V^\pi(s)$ and $Q_*(s,a) := \max_\pi Q^\pi(s,a)$. Suppose $\gamma \in (0,1)$. Prove the following Bellman optimality equations:

$$V_*(s) = \max_a Q_*(s,a) \tag{1}$$

$$Q_*(s,a) = R_s^a + \gamma \sum_{s'} P_{ss'}^a V_*(s'). \tag{2}$$

Please carefully justify every step of your proof. (Hint: For (1), you may first prove that $V_*(s) \leq \max_a Q_*(s,a)$ and then show $V_*(s) < \max_a Q_*(s,a)$ cannot happen by contradiction. On the other hand, (2) can be shown by using the similar argument or by leveraging the fact that $Q^\pi(s,a) = R_s^a + \gamma \sum_{s'} P_{ss'}^a V^\pi(s')$)

**(b)** Based on (a), we thereby have the recursive Bellman optimality equation for the optimal action-value function $Q_*$ as:

$$Q_*(s,a) = R_s^a + \gamma \sum_{s'} P_{ss'}^a \big(\max_{a'} Q_*(s',a')\big) \tag{3}$$

Similar to the standard Value Iteration, we can also study the *Q-Value Iteration* by defining the Bellman optimality operator $T^* : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \rightarrow \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ for the action-value function: for every state-action pair $(s,a)$

$$[T^*(Q)](s,a) := R_s^a + \gamma \sum_{s'} P_{ss'}^a \max_{a'} Q(s',a') \tag{4}$$

Show that the operator $T^*$ is a $\gamma$-contraction operator in terms of $\infty$-norm. Please carefully justify every step of your proof. (Hint: For any two action-value functions $Q, Q'$, we have $\|T^*(Q) - T^*(Q')\|_\infty = \max_{(s,a)} \big|[T^*(Q)](s,a) - [T^*(Q')](s,a)\big|$)

**Problem 2 (Soft Policy Iteration for Regularized MDPs)** (20 points)

In this problem, let us verify the policy update of Soft Policy Iteration discussed in Page 41 of Lecture 4: In the $k$-th iteration, given the entropy-regularized Q function $Q_\Omega^{\pi_k}$ with $\Omega(\pi(\cdot|s)) := \sum_{a\in\mathcal{A}} \pi(a|s)\log\pi(a|s)$, under Soft Policy Iteration, the new policy for the $k+1$-iteration can be obtained by solving the following optimization problem for each state $s \in \mathcal{S}$:

$$\pi_{k+1}(\cdot|s) = \arg\max_\pi \big\{\langle \pi(\cdot|s), Q_\Omega^{\pi_k}(s,\cdot)\rangle - \Omega(\pi(\cdot|s))\big\}. \tag{5}$$

Note that we can further write the above optimization problem in a more explicit manner:

$$\max_{\pi(\cdot|s)} \sum_{a\in\mathcal{A}} \big(\pi(a|s) Q_\Omega^{\pi_k}(s,a) - \pi(a|s)\log\pi(a|s)\big), \quad \text{subject to} \sum_{a\in\mathcal{A}} \pi(a|s) - 1 = 0, \tag{6}$$

where the constraint is meant to ensure that $\pi$ is a valid policy. Please show that the optimal solution to the above optimization problem is

$$\pi_{k+1}(\cdot|s) = \frac{\exp(Q_{\Omega}^{\pi_k}(s,\cdot))}{\sum_{a\in\mathcal{A}} \exp(Q_{\Omega}^{\pi_k}(s,a))}. \tag{7}$$

(Hint: To show (7), we leverage the Lagrange multiplier technique that we learn in the Calculus class. Specifically, let $\mu \in \mathbb{R}$ be the *Lagrange multiplier* associated with the constraint in 6. Then, we can construct the *Lagrangian* as

$$L(\pi) := \sum_{a\in\mathcal{A}} \big(\pi(a|s)Q_{\Omega}^{\pi_k}(s,a) - \pi(a|s)\log\pi(a|s)\big) - \mu\big(\sum_{a\in\mathcal{A}} \pi(a|s) - 1\big). \tag{8}$$

Then, the optimal solution satisfies $\frac{\partial L(\pi)}{\partial \pi(a|s)} = 0$, for every $a \in \mathcal{A}$.)

**Problem 3 (Implementing Policy Iteration and Value Iteration)** (40 points)

In this problem, we will implement policy iteration and value iteration for a classic MDP environment called "Taxi" (Dietterich, 2000). This environment has been included in the OpenAI Gym: `https://gym.openai.com/envs/Taxi-v3/`. To accomplish this task, you may take the following steps:

- Get familiar with the Taxi environment by reading the Gym documentation at `https://www.gymlibrary.dev/environments/toy_text/taxi/`. The state space consists of 500 possible states as there are 25 taxi positions, 5 possible locations of the passenger (including the case when the passenger is in the taxi), and 4 destination locations. Moreover, the agent has 6 possible actions (namely, 0: move south; 1: move north; 2: move east; 3: move west; 4: pickup passenger; 5: drop off passenger). The rewards are: (i) -1 per step unless other reward is triggered; (ii) +20 for delivering passenger; (iii) -10 for executing "pickup" and "drop-off" actions illegally.

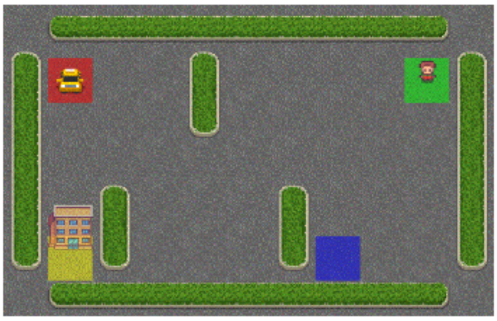

Figure 1: An illustration of the Taxi environment.

- Read through **policy_and_value_iteration .py** and then implement the two functions **policy_iteration** and **value_iteration** based on the pseudo code of PI and VI provided in the lecture slides.

- Note: Please set $\gamma = 0.9$ and the termination criterion $\varepsilon = 10^{-3}$. Moreover, you could use either Taxi-v2 or Taxi-v3 environment (Taxi-v3 is recommended). Note that discrepancy = 0 is a necessary condition (but not sufficient) of correct implementation, and with the default $\varepsilon = 10^{-3}$, you shall be able to observe zero discrepancy between the policies obtained by PI and VI.