IOC5259 Spring 2022: Reinforcement Learning **(Due: 2022/03/14, Monday, 21:00)**

Homework 1–Part I: Planning for MDPs

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) (10+10=20 points)

(a) Recall that in Lecture 3, 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 $V_*(s) \le \max_a Q_*(s,a)$ 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 value iteration, we can study the *Q-value iteration* by defining the Bellman optimality operator $T^* : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \to \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ 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 γ-contraction operator in terms of ∞-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 (Distributional Perspective of MDPs) (10+10=20 points)

Recall that given a policy π, the distributional Bellman operator $B^\pi : \mathcal{Z} \to \mathcal{Z}$ is defined as

$$[B^\pi Z](s,a) \overset{D}{:=} r(s,a) + \gamma P^\pi Z(s,a), \tag{5}$$

where $\gamma \in (0,1)$. In the following subproblems, we would like to show that the B^π is a contraction operator in the maximal form of the Wasserstein metric (i.e. $\bar{d}_p$ defined in Lecture 5). For ease of exposition, we further consider the following notations: Given any two random variables U, V with CDFs F_U, F_V, we write $d_p(U,V) := d_p(F_U, F_V)$.

(a) To begin with, show that the Wasserstein metric satisfies the following nice properties: Let U and V be two random variables. Let A be another random variable that is independent of U and V. Let Q be a Bernoulli random variable that is independent of U and V and satisfies $P(Q=1) = q$:

- (i) $d_p(aU, aV) = |a| d_p(U, V)$, for any $a \in \mathbb{R}$
- (ii) $d_p(A + U, A + V) \le d_p(U, V)$
- (iii) $d_p(QU, QV) \le q \cdot d_p(U, V)$

(Hint: For (i), you may first show that $d_p(aU, aV) \le |a| d_p(U, V)$; For (ii), for any pair of random variables U', V' with $U' \overset{D}{=} U$, $V' \overset{D}{=} V$, consider some random variable A' that satisfies $A' \overset{D}{=} A$ and is independent of U', V'. Then, try to connect $d_p(A' + U', A' + V')$ and $d_p(U, V)$; For (iii), based on each possible joint distribution of U, V, construct one straightforward joint distribution of QU, QV)

(b) By using the result in (a) and the partition lemma (Lemma 1 in [Belleware et al., ICML 2017]), show that B^π is a γ-contraction operator in $\bar{d}_p$. (Hint: As an intermediate step of the proof, you may need to show that $d_p(B^\pi Z_1(s, a), B^\pi Z_2(s, a)) \le \gamma \sup_{\bar{s},\bar{a}} d_p(Z_1(\bar{s}, \bar{a}), Z_2(\bar{s}, \bar{a}))$, for any state-action pair (s, a))

Problem 3 (Implementing Policy Iteration and Value Iteration) (20 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/. Read through **policy_and_value_iteration .py** and then implement the two functions **policy_iteration** and **value_iteration** (Note: please set $\gamma = 0.9$ and the termination criterion $\varepsilon = 10^{-3}$. Moreover, you could use either Taxi-v2 or Taxi-v3 environment. Note that discrepancy = 0 is a necessary condition 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).