All the questions are mandatory unless otherwise stated. For submission instructions please refer to the website and the instructions file.

1 Theory

Question 1. [25 points] **Bias-variance trade-off** *(Choose one of the following two tracks)*

A. Monte-Carlo (MC) methods are defined by forming estimates of returns (G_t) by performing rollouts in the environment (to collect trajectories), usually without using concepts from Bellman equations. Temporal Difference (TD) methods are similar to MC methods, except they use a "bootstrapped" target, i.e. the target value depends on the current estimate of the value function.

i Explain how these differences affect the learning process, specifically the bias-variance trade-off. For each method, comment how the estimated value function depends on the amount of data available to the agent. Also, briefly explain the scalability of these approaches with experience. Relate your answer to TD(λ) methods.

ii MC methods are known to have bounds that are independent on the size of the state space:

$$|\widehat{v}(s) - v^{\pi}(s)| \leqslant \frac{R_{\max}}{1-\gamma}\sqrt{\frac{1}{2n}\ln\frac{2}{\delta}}, \tag{1}$$

$\forall s \in \mathcal{S}$, where n is the sample size, rewards are in $[0, R_{max}]$, $\widehat{v}$ is the MC estimate of the value function for policy π, v^{π} is the true value function of this policy and we know the bound holds with probability $1-\delta$. Explain why this is the case, along with the advantages and disadvantages of it. In your answer, focus on how one would learn the optimal policy using MC methods and the sample complexity of such an approach.

iii Explain why/how TD methods can be viewed as learning an implicit model of the world and solving for it at the same time. Describe why this leads to different solutions for MC and TD.

B. In this question we will bound the bias and variance of phased TD updates (Kearns and Singh, 2000). Phased TD algorithm simplifies the complexities introduced

by learning rate α. In phased TD, we are provided with n trajectories generated following a policy π from every state at each phase. In other words, each phased update is carried out using the data from n trajectories. The phased value function is then obtained by averaging the updates on these trajectories. Specifically, we define phased TD(k) and phased TD(λ) respectively as below,

$$\bar{v}^k_{\pi,t+1}(s) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k-1}\gamma^{j-1}r^{(i)}_{j+1} + \gamma^k \bar{v}^k_{\pi,t}(s^{(i)}_{n+1}),$$

$$\bar{v}^\lambda_{\pi,t+1}(s) = \frac{1}{n}\sum_{i=1}^{n}(1-\lambda)\sum_{k=1}^{\infty}\lambda^{k-1}\Big(\sum_{j=1}^{k-1}\gamma^{j-1}r^{(i)}_{j+1} + \gamma^k \bar{v}^\lambda_{\pi,t}(s^{(i)}_{k+1})\Big).$$

i Let $S(t)$ be the set of trajectories generated by π in phase t (n trajectories from each state), $\bar{v}^k_\pi(s,t)$ be the value function estimate of phased TD(k) after phase t, and let $\Delta_t = \max_s(|\bar{v}^k_{\pi,t}(s) - v_\pi(s)|)$. Then for any $\gamma \in [0,1)$, rewards $r \in [-1,+1]$, $0 < \delta < 1$ and $\varepsilon = \sqrt{\frac{2log(2k/\delta)}{n}}$, with probability $1-\delta$, prove that $\Delta_t \leqslant \varepsilon\left(\frac{1-\gamma^k}{1-\gamma}\right) + \gamma^k\Delta_{t-1}$.

ii Let $S(t)$ be the set of trajectories generated by π in phase t (n trajectories from each state), $\bar{v}^\lambda_\pi(s,t)$ be the value function estimate of phased TD(λ) after phase t, and let $\Delta_t = \max_s(|\bar{v}^\lambda_{\pi,t}(s) - v_\pi(s)|)$. Then for any $\gamma \in [0,1)$, $\lambda \in [0,1)$, rewards $r \in [-1,+1]$, $0 < \delta < 1$ and $\varepsilon = \sqrt{\frac{2log(2k/\delta)}{n}}$, with probability $1-\delta$, prove that $\Delta_t \leqslant \varepsilon \min_k\Big(\Big(\frac{1-(\gamma\lambda)^k}{1-\gamma\lambda}\Big) + \frac{(\gamma\lambda)^k}{1-\gamma\lambda}\mathbf{1}\Big) + \Big(\frac{(1-\lambda)\gamma}{1-\gamma\lambda}\Big)\Delta_{t-1}$.

Tip: Use Hoeffding and union bound

Question 2. [25 points] *(Choose one of the following two tracks)*

A. SARSA/Expected SARSA/Q-learning with Function Approximation

i Outline a SARSA algorithm with function approximation that uses Boltzman exploration instead of the standard ε-greedy policy.

ii Explain why SARSA with ε-greedy policy can fail to converge and why Boltzman exploration can help in this respect.(see Gordon, 1996)

iii Explain the connection between SARSA and Q-learning. Show how Q-Learning can be viewed as a special case of Expected Sarsa.

B. Certainty-equivalence RL

Certainty-equivalence is a model-based RL algorithm which means the algorithm will first estimate a *reward model* $R(s,a)$ and a *transition model* model $P(s,a,\cdot)$ from the available data. In tabular setting, given a dataset of trajectories, $D = \{(s_1, a_1, r_1, s_2, \ldots, s_{H+1})\}$, for a horizon H, we first convert it into chunks of transition tuples, $\{(s,a,r,s')\}$. So, for a trajectory of horizon H, we have H tuples:

$$(s_1, a_1, r_1, s_2), (s_2, a_2, r_2, s_3), \ldots (s_H, a_H, r_H, s_{H+1})$$

Using these chunks the tabular certainty-equivalence model estimates the transition model and the reward model based on maximum likelihood which corresponds to the empirical frequency of the dataset of the observed transitions as shown below:

$$\widehat{P}(s,a) = \frac{1}{|D_{s,a}|} \sum_{(r,s') \in D_{s,a}} \mathbf{e}_s \tag{2}$$

$$\widehat{R}(s,a) = \frac{1}{|D_{s,a}|} \sum_{(r,s') \in D_{s,a}} r,$$

where e_s is a unit vector whose entry for state s is 1 and all other entries are 0, define $D_{s,a}$ as the subset of tuples where the first element of the tuple is s and the second is a, we write $(r,s') \in D_{s,a}$. Let $n = |D_{s,a}|$ and note that the frequency of each state and action pair $n(s,a) > 0$, $\forall s \in \mathcal{S}$ and $\forall a \in \mathcal{A}$. The algorithm then performs policy evaluation or control using these models.

The other popular family of RL algorithms estimates the q-value functions, for example Q-learning and Sarsa, without estimating a model.

i Compare the two approaches by specifying the advantages and disadvantages in terms of performance, memory, and time complexity.

ii We are interested in deriving high-probability guarantees for for the optimal policy in $\widehat{M} = (\mathcal{S}, \mathcal{A}, \widehat{P}, \widehat{R}, \gamma)$ as a function of n. When n is large enough $\widehat{R} \approx R$ and $\widehat{P} \approx P$ with probability at least $1 - \delta$ ($\delta \in (0,1)$), one can prove the following

result using union bound and Hoeffding's inequality

$$\max_{s,a} |\widehat{R}(s,a) - R(s,a)| \leqslant R_{\max}\sqrt{\frac{1}{2n}\ln\frac{4|\mathcal{S}\times\mathcal{A}|}{\delta}} \tag{3}$$
$$\max_{s,a,s'} \left|\widehat{P}\left(s'|s,a\right) - P\left(s'|s,a\right)\right| \leqslant \sqrt{\frac{1}{2n}\ln\frac{4|\mathcal{S}\times\mathcal{A}\times\mathcal{S}|}{\delta}}$$

To prove the suboptimality of $\pi^{\star}_{\widehat{M}}$, you should first try to prove the simulation lemma (which bounds the policy evaluation error in the learned MDP) for all $s \in S$

$$\left\|V^{\pi}_{\widehat{M}} - V^{\pi}_{M}\right\|_{\infty} \leqslant \frac{\varepsilon_R}{1-\gamma} + \frac{\gamma\varepsilon_P R_{\max}}{2(1-\gamma)^2}, \tag{4}$$

where $\max_{s,a} |\widehat{R}(s,a) - R(s,a)| \leqslant \varepsilon_R$ and $\max_{s,a} \|\widehat{P}(s,a) - P(s,a)\|_1 \leqslant \varepsilon_P$.

iii We are now interested in going from policy evaluation to the control setting. We want to bound the error of the optimal policy derived in the approximated MDP $\widehat{M}$: $\pi^{\star}_{\widehat{M}}$. Prove that

$$\forall s \in \mathcal{S}, V^{\star}_{M}(s) - V^{\pi^{\star}_{\widehat{M}}}_{M}(s) \leqslant 2 \sup_{\pi:\mathcal{S}\to\mathcal{A}} \left\|V^{\pi}_{\widehat{M}} - V^{\pi}_{M}\right\|_{\infty} \tag{5}$$

iv Putting (ii) and (iii), along with concentration inequalities, prove the following

$$V^{\star}_{M}(s) - V^{\pi^{\star}_{\widehat{M}}}_{M}(s) = O\left(\frac{|\mathcal{S}|}{\sqrt{n}(1-\gamma)^2}\right), \forall s \in \mathcal{S} \tag{6}$$

2 Coding

Question 1. [25 points] **Continuous Random Walk (Prediction task)**
In this task, a state, s_t, is defined as a point in the interval 0 to 1. This is an episodic task i.e, the episode ends when either state 0 or 1 is exceeded. The episode starts at state 0.5. At each step, the agent moves up or down by a uniformly selected random step in the interval [-0.2, 0.2]. The reward r_t is 0 everywhere except when the agent terminates. When the agent terminates, the reward is equal to the termination state. For example, if agent terminates in a state -0.05, then it receives a reward of -0.05. Set the discount rate γ to 1.

- Implement sparse coarse coding (refer to section 9.5.4 Sutton and Barto, 2018) to construct binary features from the real valued state. You are required to follow the following protocol: Divide the state space [0,1] into 10 equal sized intervals. Also add an additional interval in order to allow for an offset of the whole tiling. Repeat this 10 times to obtain 10 different tilings each offset by a different randomly selected fraction of the interval. In addition, add a feature corresponding to each interval that takes the value 1 when the state was within that tile, and 0 otherwise.

- Using the features constructed, implement TD(λ) with *accumulating traces* (see Chapter 12 Sutton and Barto, 2018) for various values of λ and learning rate α. Use a linear function approximator that is initialized to predict 0. You must run the experiment on 50 different seeds for each combination of λ and α values and average the results.

- Present your results in a plot where the x-axis is learning rate and the y-axis is Mean Squared Value Error (MSVE) after the agent is trained for 25 episodes. Since the state space is continuous, sample 21 different points in the interval [0,1], evenly spaced at 0.05. Also, note that the correct predictions are equal to the position. Note that for each λ value you will get a curve that is averaged over 50 different seeds. Plot the averaged result, and also show the confidence interval (using the standard deviation from the all the runs).

- What do you observe? Write a brief report on your observations.

Question 2. [25 points] **Control task** *(Choose one of the following two tracks)*

A. Use the mountain car environment from Open AI gym (Brockman et al., 2016). Set the discount factor γ to 1.

- Construct features from the 2D continuous state space using grid tilings. Construct the features in exactly the same way as section 10.1 of Sutton's book (Sutton and Barto, 2018). To summarize the method, you are required to use 8 tilings, with each tile 1/8th of the bounded distance (refer to the section 10.1 for details) in each dimension and asymmetric offsets. As a result, you obtain features $\varphi(s, a)$.

- Using the features constructed, implement SARSA(λ) with *replacing traces* (see Chapter 12 Sutton and Barto, 2018) for various values of λ and learning rate α. Use a linear function approximator that is initialized to predict 0. Run the

experiment on 20 different seeds for each combination of λ and α values. For exploration use an ε-greedy policy with $\varepsilon = 0.1$.

- Present your results in a plot where the x-axis is learning rate and the y-axis is number of steps per episode to reach the goal once the algorithm is trained on 1000 episodes (this will be maximum steps i.e. 2000 steps if the goal is not reached). Note that for each λ value you will get a curve that is averaged over 20 different seeds. Also show the confidence interval (using the standard deviation obtained from the runs).

- What do you observe? Write a brief report on your observations.

B. Use the cartpole environment from Open AI gym (Brockman et al., 2016). Set the discount factor γ to 0.9.

- Implement 4-step SARSA, 4-step Expected SARSA, and 4-step q-learning algorithms (see chapter 10 Sutton and Barto, 2018) with a two layered Neural Network (64 hidden units in each layer) to learn the task. Use experience replay to stabilize the learning process. For exploration use an ε-greedy policy with $\varepsilon = 0.1$.

- Run all 3 algorithms with the following sizes of experience replay buffers: 50, 100, 250, 500.

- Present your results in two different plots. (1) In the first plot, the x-axis should show the learning rate and the y-axis the average return on the last 10 episodes of training (you must train until convergence). In this plot, you will have 4 different lines: one line for each variant of experience replay. (2) For the second plot, choose the best combination of experience replay and learning rate. For this combination, present a plot with episodes in x-axis and the return obtained in the corresponding episode as y-axis. Run all the experiments on 20 different seeds. Also show the confidence interval (using the standard deviation obtained from the runs).

- What do you observe? Write a brief report on your observations.

References

Brockman, Greg et al. (2016). *OpenAI Gym*. eprint: arXiv:1606.01540.

Gordon, Geoffrey J. (1996). *Chattering in SARSA(lambda) - A CMU Learning Lab Internal Report*. Tech. rep.

Kearns, Michael J and Satinder P Singh (2000). "Bias-Variance Error Bounds for Temporal Difference Updates." In: *COLT*. Citeseer, pp. 142–147.

Sutton, Richard S and Andrew G Barto (2018). *Reinforcement learning: An introduction*. MIT press.