CS234: Reinforcement Learning – Exam #1

Winter 2021-2022

Name:

8-digit Student ID:

Instructions

This is a 90-minute exam.
You are allowed one two-sided page of handwritten notes for the quiz. You are not allowed to collaborate with anyone else. The only exception is that you can ask the CAs for clarification.
The Stanford Honor Code is printed below. By submitting this exam, you are agreeing to adhere to the standards of the honor code.
The Honor Code states:

1. The Honor Code is an undertaking of the students, individually and collectively
   (a) that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;
   (b) that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.
2. The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.
3. While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

Penalties for violation of the Honor Code can be serious (*e.g.* suspension or even expulsion).

Problem 1 [9 points]

The following questions are of objective type and are either **True or False** questions or **Multiple Choice** questions. **Each MCQ is worth 2 points and each True or False is worth 1 point.** For the True or False questions, state True or False and also give a one line reason for your answer. You get the point only if both your answer and reasoning is correct. **For the Multiple choice questions, more than one option can be correct** and you get 2 points only if you choose all the correct options. No partial marking. No explanation required.

1. You want to train a Q-network on an environment. Which of the following will **guarantee** convergence to the optimal value function when run for an infinite number of steps and you visit each state-action pair an infinite number of times in the limit? Assume step size and learning rate are tuned correctly. Assume we start with some initial value of ϵ close to 1.

   (a) The states and actions are discrete and finite and we use a tabular representation for the Q-function but we don't decay ϵ to 0

   (b) The states and actions are discrete and finite and we use a tabular representation for the Q-function and we decay ϵ to 0

   (c) We train a DQN with an infinitely large neural network and decay ϵ to 0

   (d) We train a DQN with a target network and a replay buffer and and decay ϵ to 0

   (e) We train a double DQN and decay ϵ to 0

   Solution: a,b. Recall that tabular Q-learning is guaranteed to converge to the optimal Q values as long as all states are visited infinitely often. This is true if ϵ is decayed or not decayed. In class we discussed using deep neural networks with Q-learning, as in DQN in assignment 2. If SARSA or MC is used instead with a tabular representation, then ϵ must be decayed to zero. (c) is incorrect because when we use function approximation, we get into deadly triad issues and we cannot guarantee convergence even if the Q-values can be represented perfectly using function approximation (which you can do with an infinitely large network). (d) and (e) are incorrect for similar reasons.

2. Consider an MDP with N states where the agent always gets stuck in a loop of 3 states A, B, and C and doesn't visit the other states. We are running Q-learning on this MDP for a long time and α is tuned appropriately. Initially, $\epsilon = 1$. Choose all the correct answers from below:

   (a) When ϵ is decayed as $1/t$, the Q-value will converge to the optimal values everywhere

   (b) When ϵ is decayed as $1/t$, the Q-value will converge to the optimal values in states A, B and C.

   (c) When ϵ is not decayed, the Q-value will converge to the optimal values in states A, B and C.

   (d) When ϵ is decayed as $1/t$, the policy will be optimal everywhere.

   (e) When ϵ is not decayed, the policy will be optimal in A, B, C.

   Solution: b,c. Q-learning will converge to the optimal Q values for A, B and C because it visits all those states infinitely often and the agent always gets stuck in a loop of those states and does not visit any others. This is true whether or not ϵ is decayed (a) and (d) are incorrect because the Q-value may not converge to the optimal value for all states. (e) is wrong because when ϵ is not decayed, the policy will still take a random action a fixed proportion of the time in states A, B and C.

3. Consider the following estimates of $Q^{\pi}(s_0, a_0)$ when $\gamma = 1$:
   A: $\sum_{t=0}^{t=\infty} r(s_t, a_t)$
   B: $r(s_0, a_0) + V^{\pi}(s_1)$
   C: $r(s_0, a_0) + r(s_1, a_1) + V^{\pi}(s_2)$
   Here, $a_1, a_2...$ are actions taken according to the policy π and $s_1, s_2..$ are the states visited along the trajectory.
   Choose all the correct answers below:

(a) A is a more biased estimator than C

(b) C is a more biased estimator than B

(c) Variance of estimator A > variance of estimator C

(d) B and C have same variance

(e) B and C are both biased estimators

Solution: c,e. This problem is about policy evaluation and is asking about comparing the bias and variance of a MC estimator versus a temporal difference and a modified temporal difference estimator, that bootstrap a current estimate V^π. As discussed in lecture, Monte Carlo estimators (like A) have higher variance than TD (estimator B) and estimator C (which uses two steps of rewards and then bootstraps). Monte Carlo estimator A is an unbiased estimator. Estimators that bootstrap (estimator B and C) are biased due to boostrapping with an estimator V^π of the value under π.

4. **True or False?** Assume we don't have access to the transition dynamics of the environment. Given only the optimal Q function, we can still recover the optimal policy but given only the optimal V function, we cannot.
Solution: True. To follow the optimal policy, take the best action from the Q-values: $\pi^*(s) = \arg\max_a Q^*(s,a)$. If we don't have the transition dynamics, we cannot recover the policy from the V-function

Consider the following transitions observed from an undiscounted MDP with two states P and Q. The numbers indicate rewards.
Trajectory 1: P, +3, P, +2, Q, -4, P, +4, Q, -3
Trajectory 2: Q, -2, P, +3, Q, -3

5. We obtain V(P) = 1 from the transition data. Which of the following policy evaluation algorithms could have resulted in this answer? Assume the value functions are initialized to 0.

(a) First-visit MC

(b) Every-visit MC

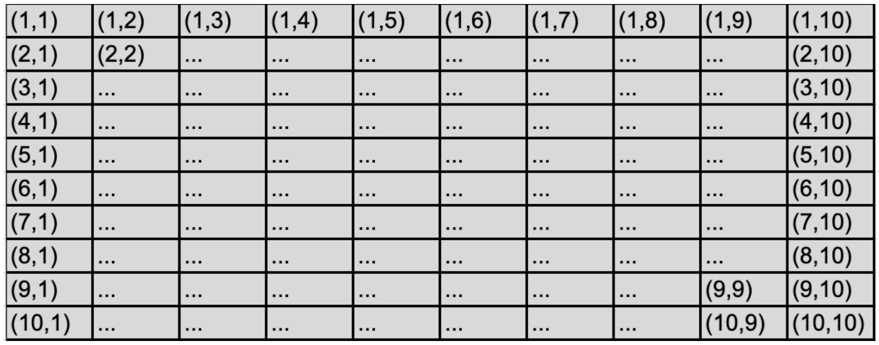
(c) A single TD(0) update with $\alpha = 1/3$ on the transition P, +3, P.

(d) A single TD(1) update with $\alpha = 1/4$ on the transition P, +3, P, +2, Q.

Solution: a,c. TD(1) was not defined in lecture, so we ignored answers to (d) in grading.

Problem 2 [9 points]

Consider a 10×10 grid world as shown in the figure below with coordinates (x, y) ranging from $(1, 1)$ to $(10, 10)$. The action space allows you to move left (0), right (1), up (2) or down (3) and, if you run into a wall, you remain in the same square. The reward function is -1 for taking any action from any state and the episode ends when you reach the goal state $(10, 10)$. Assume the discount factor $\gamma = 1$



(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)	(1,8)	(1,9)	(1,10)
(2,1)	(2,2)	...	...	...	...	...	...	...	(2,10)
(3,1)	...	...	...	...	...	...	...	...	(3,10)
(4,1)	...	...	...	...	...	...	...	...	(4,10)
(5,1)	...	...	...	...	...	...	...	...	(5,10)
(6,1)	...	...	...	...	...	...	...	...	(6,10)
(7,1)	...	...	...	...	...	...	...	...	(7,10)
(8,1)	...	...	...	...	...	...	...	...	(8,10)
(9,1)	...	...	...	...	...	...	...	(9,9)	(9,10)
(10,1)	...	...	...	...	...	...	...	(10,9)	(10,10)

Figure 1: Grid World

(a) **[2 pts]** By inspection, write an expression for the optimal Q-value for each state action pair as a function of x, y and the action a. Ignore the cases where actions run you into walls.

Solution: $Q((x, y), a) = x + y - 22 + 2(a \mod 2)$
A piece-wise solution across different actions that captures the same Q-function is also acceptable

(b) **[2 pts]** If our state-action pair is represented by the features $[1, x, y, a]$, does there exist a linear function approximator that can learn the optimal Q-function? Here, a is the action and can be 0,1,2 or 3 as described in the question. If so, write down the value of the optimal weight vector $[w_0, w_1, w_2, w_3]$ that achieves this. Ignore the cases where actions run you into walls.

Solution: No.

(c) **[2 pts]** If our state-action pair is represented by the features $[1, x, y, a \mod 2]$, does there exist a linear function approximator that can learn the optimal Q-function? Here, a is the action and can be 0,1,2 or 3 as described above. If so, write down the value of the optimal weight vector $[w_0, w_1, w_2, w_3]$ that achieves this. Ignore the cases where actions run you into walls.
Solution: Yes, $w = [-22, 1, 1, 2]$

(d) **[3 pts]** At least one of the representations from (b) or (c) can represent the optimal Q-function (ignoring actions that run you into walls). Will using linear function approximation with that representation converge to the optimal Q-values when run under GLIE assumptions? Will using a DQN converge to the optimal Q-values when run under GLIE assumptions? Explain why or why not.

Solution: No for both. Q-learning under function approximation is not guaranteed to converge because of the deadly triad issue of off-policy learning, bootstrapping and function approximation. Note this is true even when the function approximator can represent the optimal Q values.

Problem 3 [10 points]

In general for a finite-horizon MDP, the optimal policy is non-stationary (it depends on the number of timesteps left). The Bellman equation is defined similarly to what you have seen before. For all timesteps $h \in [H]$,

$$V_h^\pi(s) = Q_h^\pi(s, \pi_h(s)) \qquad Q_h^\pi(s,a) = \mathcal{R}(s,a) + \mathbb{E}_{s' \sim \mathcal{T}(\cdot|s,a)}\left[V_{h+1}^\pi(s')\right],$$

where $Q_H^\pi(s,a) = \mathcal{R}(s,a)$ and we denote the index set $[H] \triangleq \{1, 2, \ldots, H\}$.

(a) **[5 pts]** For any $h \in [H]$ and any two non-stationary policies $\pi = (\pi_h, \pi_{h+1}, \ldots, \pi_H)$ and $\overline{\pi} = (\overline{\pi}_h, \overline{\pi}_{h+1}, \ldots, \overline{\pi}_H)$, prove the following identity by induction on h:

$$V_h^\pi(s) - V_h^{\overline{\pi}}(s) = \sum_{h'=h}^{H} \mathbb{E}_{s_{h'} \sim d_{h'}^\pi(\cdot|s_h = s)}\left[Q_{h'}^{\overline{\pi}}(s_{h'}, \pi_{h'}(s_{h'})) - Q_{h'}^{\overline{\pi}}(s_{h'}, \overline{\pi}_{h'}(s_{h'}))\right],$$

where $d_{h'}^\pi$ denotes the probability distribution over states visited by policy π at timestep h' after starting at timestep h in state s.

Solution: This result is a variant of the performance-difference lemma [Kakade and Langford, 2002] adapted to the finite-horizon setting. The proof of the performane-difference lemma for the infinite-horizon, discounted MDP setting was given as Problem 1 of Assignment 2.
We prove the claim by induction on h, as specified. As a base case for the induction, take $h = H$ and simply observe that

$$\begin{aligned}
V_h^\pi(s) - V_h^{\overline{\pi}}(s) &= Q_H^\pi(s, \pi_H(s)) - Q_H^{\overline{\pi}}(s, \overline{\pi}(s)) \\
&= \mathcal{R}(s, \pi_H(s)) - Q_H^{\overline{\pi}}(s, \overline{\pi}(s)) \\
&= Q_H^{\overline{\pi}}(s, \pi_H(s)) - Q_H^{\overline{\pi}}(s, \overline{\pi}(s)) \\
&= \mathbb{E}_{s_H \sim d_H^\pi(\cdot|s_H = s)}\left[Q_H^{\overline{\pi}}(s_H, \pi_H(s_H)) - Q_H^{\overline{\pi}}(s_H, \overline{\pi}(s_H))\right].
\end{aligned}$$

Now assume as an inductive hypothesis that the claim holds at timestep $h+1$. Then,

$$\begin{aligned}
V_h^\pi(s) - V_h^{\overline{\pi}}(s) &= Q_h^\pi(s, \pi_h(s)) - Q_h^{\overline{\pi}}(s, \overline{\pi}_h(s)) \\
&= Q_h^\pi(s, \pi_h(s)) - Q_h^{\overline{\pi}}(s, \pi_h(s)) + Q_h^{\overline{\pi}}(s, \pi_h(s)) - Q_h^{\overline{\pi}}(s, \overline{\pi}_h(s)).
\end{aligned}$$

Examining the first difference term in isolation, we have

$$Q_h^\pi(s, \pi_h(s)) - Q_h^{\overline{\pi}}(s, \pi_h(s)) = \mathbb{E}_{s' \sim \mathcal{T}(\cdot|s, \pi_h(s))}\left[V_{h+1}^\pi(s') - V_{h+1}^{\overline{\pi}}(s')\right].$$

Taking an expectation, we have

$$\begin{aligned}
\mathbb{E}_{s_h \sim d^\pi(\cdot|s_h = s)}\left[\mathbb{E}_{s' \sim \mathcal{T}(\cdot|s_h, \pi_h(s_h))}\left[V_{h+1}^\pi(s') - V_{h+1}^{\overline{\pi}}(s')\right]\right] &= \mathbb{E}_{s_{h+1} \sim d_{h+1}^\pi(\cdot|s_h = s)}\left[V_{h+1}^\pi(s_{h+1}) - V_{h+1}^{\overline{\pi}}(s_{h+1})\right] \\
&= \sum_{h'=h+1}^{H} \mathbb{E}_{s_{h'} \sim d_{h'}^\pi(\cdot|s_h = s)}\left[Q_{h'}^{\overline{\pi}}(s_{h'}, \pi_{h'}(s_{h'})) - Q_{h'}^{\overline{\pi}}(s_{h'}, \overline{\pi}_{h'}(s_{h'}))\right]
\end{aligned}$$

where the first line follows from the tower property of expectation and the second line follows from the inductive hypothesis. Meanwhile, for the second difference term, taking the same expectation yields

$$\mathbb{E}_{s_h \sim d^\pi(\cdot|s_h = s)}\left[Q_h^{\overline{\pi}}(s, \pi_h(s)) - Q_h^{\overline{\pi}}(s, \overline{\pi}_h(s))\right] = \mathbb{E}_{s_h \sim d^\pi(\cdot|s_h = s)}\left[Q_h^{\overline{\pi}}(s_h, \pi_h(s_h)) - Q_h^{\overline{\pi}}(s_h, \overline{\pi}_h(s_h))\right].$$

So, applying linearity of expectation and combining terms yields

$$\begin{aligned}
V_h^\pi(s) - V_h^{\overline{\pi}}(s) &= \mathbb{E}_{s_h \sim d^\pi(\cdot|s_h = s)}\left[V_h^\pi(s_h) - V_h^{\overline{\pi}}(s_h)\right] \\
&= \sum_{h'=h}^{H} \mathbb{E}_{s_{h'} \sim d_{h'}^\pi(\cdot|s_h = s)}\left[Q_{h'}^{\overline{\pi}}(s_{h'}, \pi_{h'}(s_{h'})) - Q_{h'}^{\overline{\pi}}(s_{h'}, \overline{\pi}_{h'}(s_{h'}))\right],
\end{aligned}$$

as desired.

Algorithm 1: Solution: Policy Search by Dynamic Programming (PSDP) [Bagnell et al., 2003]

```
Data: Finite-horizon MDP $\mathcal{M}$, Policy class $\Pi \subseteq \{\mathcal{S} \to \mathcal{A}\}$, Sequence of state distributions
      $\mu = (\mu_1, \mu_2, \dots, \mu_H)$.
for $h = H, H-1, H-2 \dots, 1$ do
    $\pi = (\pi_{h+1}, \dots, \pi_H)$
    if $h = H$ then
        $Q_h^\pi = \mathcal{R}(s,a), \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}$
    end
    else
        $Q_h^\pi(s,a) = \mathcal{R}(s,a) + \mathbb{E}_{s' \sim \mathcal{T}(\cdot|s,a)}\left[V_{h+1}^\pi(s')\right], \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}$
    end
    $\pi_h = \arg\max_{\pi' \in \Pi} \mathbb{E}_{s \sim \mu_h(\cdot)}\left[Q_h^\pi(s, \pi'(s))\right]$
end
Return non-stationary policy $\pi = (\pi_1, \pi_2, \dots, \pi_H)$
```

(b) **[5 pts]** Assume that the state space of $\mathcal{M}$ is finite ($|\mathcal{S}| < \infty$) and rewards are bounded in the unit interval $[0,1]$. For any two sequences of exactly H state distributions $\mu = (\mu_1, \mu_2, \dots, \mu_H)$ and $\mu' = (\mu'_1, \mu'_2, \dots, \mu'_H)$, define

$$D(\mu \mid\mid \mu') = \frac{1}{H}\sum_{h=1}^{H}\sum_{s\in\mathcal{S}}\Big|\mu_h(s) - \mu'_h(s)\Big|.$$

Given input $\mu = (\mu_1, \mu_2, \dots, \mu_H)$ (a sequence of H state distributions), Algorithm 1 outputs non-stationary policy $\pi = (\pi_1, \pi_2, \dots, \pi_H)$. Prove that for any non-stationary policy $\pi_{\text{ref}} = (\pi_{\text{ref},1}, \pi_{\text{ref},2}, \dots, \pi_{\text{ref},H}) \in \Pi^H$ and any state $s \in \mathcal{S}$,

$$\Big|V_1^{\pi_{\text{ref}}}(s) - V_1^\pi(s)\Big| \leq H^2 \cdot D(d^{\pi_{\text{ref}}} \mid\mid \mu),$$

where $d^{\pi_{\text{ref}}} = (d_1^{\pi_{\text{ref}}}, d_2^{\pi_{\text{ref}}}, \dots, d_H^{\pi_{\text{ref}}})$ (these are the state distributions reached by the non-stationary policy π_{ref}).

Hint: Remember that rewards are bounded in $[0,1]$ and, for any function $f \in \{\mathcal{S} \to \mathbb{R}\}$ such that $||f||_\infty \leq C < \infty$ and any two state distributions $\mu, \mu' \in \Delta(\mathcal{S})$,

$$\Big|\mathbb{E}_{s\sim\mu(\cdot)}\left[f(s)\right] - \mathbb{E}_{s\sim\mu'(\cdot)}\left[f(s)\right]\Big| \leq C \cdot \sum_{s\in\mathcal{S}}\Big|\mu(s) - \mu'(s)\Big|.$$

Solution: This result is proven in more generality (allowing for an approximately-optimal output of PSDP) as Theorem 1 of [Bagnell et al., 2003].
Ignoring the absolute value on the left-hand side and applying the previous part (1) (the performance-difference lemma), we have

$$V_1^{\pi_{\text{ref}}}(s) - V_1^\pi(s) = \sum_{h=1}^{H}\mathbb{E}_{s_h \sim d_h^\pi(\cdot|s_1=s)}\left[Q_h^\pi(s_h, \pi_{\text{ref},h}(s_h)) - Q_h^\pi(s_h, \pi_h(s_h))\right]$$

Since all rewards are bounded in the unit interval $[0,1]$, by assumption, it follows that value is bounded in $[0,H]$ and so the contents of the expectation above are bounded in $[-H,H]$. Therefore, applying the hint, we have

$$\begin{aligned}
V^{\pi_{\text{ref}}}(s) - V^{\pi}(s) &= \sum_{h=1}^{H} \mathbb{E}_{s_h \sim d_h^{\pi_{\text{ref}}}(\cdot|s_1=s)} \left[Q_h^{\pi}(s_h, \pi_{\text{ref},h}(s_h)) - Q_h^{\pi}(s_h, \pi_h(s_h)) \right] \\
&\leq \sum_{h=1}^{H} \left(\mathbb{E}_{s_h \sim \mu_h(\cdot)} \left[Q_h^{\pi}(s_h, \pi_{\text{ref},h}(s_h)) - Q_h^{\pi}(s_h, \pi_h(s_h)) \right] + H \cdot \sum_{s \in \mathcal{S}} \left| d_h^{\pi_{\text{ref}}} - \mu_h \right| \right) \\
&\leq \sum_{h=1}^{H} \left(\underbrace{\max_{\pi' \in \Pi} \mathbb{E}_{s_h \sim \mu_h(\cdot)} \left[Q_h^{\pi}(s_h, \pi'(s_h)) - Q_h^{\pi}(s_h, \pi_h(s_h)) \right]}_{=0} + H \cdot \sum_{s \in \mathcal{S}} \left| d_h^{\pi_{\text{ref}}} - \mu_h \right| \right) \\
&= H \cdot \sum_{h=1}^{H} \sum_{s \in \mathcal{S}} \left| d_h^{\pi_{\text{ref}}} - \mu_h \right| \\
&= H^2 \cdot \frac{1}{H} \sum_{h=1}^{H} \sum_{s \in \mathcal{S}} \left| d_h^{\pi_{\text{ref}}} - \mu_h \right| = H^2 \cdot D(d^{\pi_{\text{ref}}} \mid\mid \mu),
\end{aligned}$$

where the fourth line follows since PSDP (Algorithm 1) selects $\pi_h = \arg\max_{\pi' \in \Pi} \mathbb{E}_{s \sim \mu_h(\cdot)} \left[Q_h^{\pi}(s, \pi'(s)) \right]$, by design. Taking absolute values on both sides (the right-hand side is already non-negative) yields the desired result.

This result says that, for any non-stationary reference policy $\pi_{\text{ref}} \in \Pi^H$, the policy output by PSDP will obtain similar value if the state visitation distributions of π_{ref}, $d^{\pi_{\text{ref}}} = (d_1^{\pi_{\text{ref}}}, d_2^{\pi_{\text{ref}}}, \ldots, d_H^{\pi_{\text{ref}}})$, are close to $\mu = (\mu_1, \mu_2, \ldots, \mu_H)$ where, formally, $D(\mu \mid\mid \mu')$ is roughly the total variation distance between each of the H state distributions averaged across the entire horizon H. So, if the optimal policy $\pi^\star \in \Pi^H$ and we can supply a sequence of state distributions μ that is similar to the visitation of the optimal policy, then the performance of PSDP will be near-optimal.

References

J. Andrew Bagnell, Sham Kakade, Andrew Y. Ng, and Jeff Schneider. Policy search by dynamic programming. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, pages 831–838, 2003.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 267–274, 2002.