# CS234: Reinforcement Learning – Exam #2

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).
**Note:** Your exam will be scanned and uploaded to Gradescope. Keep all answers **away** from the page margins to avoid being cut off during the scanning process.

# Problem 1 [18 points] – True / False Multiple Choice Questions

You need to **select all that are true** and provide a **one-line justification** for your answers. You get half the point for the correct selections, and half the point for the right justifications.

1. - **2 pts.** Is the REINFORCE algorithm guaranteed to converge to the optimal policy if the policy class can represent the optimal policy ?

   (a) True (Yes)

   (b) False (No)

   Solution: False. There is no global convergence guarantee for REINFORCE because policy learning/update is through stochastic gradient descent.

2. - **2 pts.** Consider the upper confidence bound bandit (UCB) algorithm. There are 2 arms. Let $\mu_1$ be the unknown mean reward of arm 1, and $\mu_2$ be the unknown mean reward of arm 2. Select all that are true.

   (a) UCB1 may never select one of the arms

   (b) UCB1 balances exploration with exploitation

   (c) If $\mu_1 = \mu_2$, UCB1 will always select the arm that has been pulled the least number of times

   Solution: (b) is true. (c) is incorrect because UCB selects the arm with the highest upper confidence bound, which uses the empirical mean rewards. (a) is incorrect because UCB1 will sample each arm once at the beginning.

3. - **2 pts.** Behavior cloning on a dataset $\mathcal{D}$

   (a) Uses reinforcement learning to learn a policy to match the behavior policy used to gather the dataset $\mathcal{D}$

   (b) Can perform well if the dataset includes all states in the domain and an expert generated the dataset $\mathcal{D}$

   (c) Involves learning a reward model from the domain and uses that to help compute a policy

   Solution: (b) since, with full state coverage and expert action labels, we avoid the covariate shift problem. (a) is incorrect because behavior cloning uses supervised learning to learn a mapping from states to actions: it does not use the reward signal, whereas reinforcement learning would use the reward signal. (c) is inverse RL.

4. - **2 pts** Consider defining a safe RL algorithm as one that only returns a new decision policy $\pi'$ if with high probability, $V^{\pi'} \neq V^{\pi_b}$, where $V^{\pi_b}$ is the value of the current behavior policy. A safe RL algorithm:

   (a) Is not relevant in domains where some states are unrecoverable

   (b) Is particularly important in Atari games like Pong

   (c) Will guarantee when the new decision policy $\pi'$ is executed, the sum of its rewards on each trajectory will be equal or greater than if one had executed the old policy $\pi_b$

   (d) May result in the old policy $\pi_b$ still being used even if there exists a policy $\pi'$ with a higher value

   Solution: (d). (c) is wrong because safe RL algorithms guarantee an improvement with high probability. (a) is wrong because "unrecoverable state" could mean driving a car into a ditch – safe RL is quite important for those domains. (b) is wrong because in a digital game, you can just reset the game state after a horrible mistake – you don't need to be "safe".

5. - **2 pts** Let $S$ and $A$ denote the total number of states and actions, respectively. Consider a RL algorithm that at each state $s$, selects an action $a$ that satisfies $\max_{a'} Q^\star(s,a') - Q^\star(s,a) \leq \varepsilon$ with probability at least $1-\delta$ for some $\delta \in (0,1)$ on all but $\mathcal{O}\left(\frac{SA}{\varepsilon^3(1-\gamma)^6}\right)$ steps. Is this algorithm a PAC RL algorithm?

   (a) True (Yes)

   (b) False (No)

   Solution: True. This is a polynomial term.

6. - **2 pts** Assume there is no confounding; the behavior policy $\pi_b(a|s) > 0$ for all states and actions; and there is a non-zero probability of reaching any state s under $\pi_b$. The evaluation policy is $\pi_e$. Select the following statements that describes importance sampling methods that are true:

   (a) The estimate of $V^{\pi_e}$ computed by importance sampling is unbiased

   (b) The estimate of $V^{\pi_e}$ computed by weighted importance sampling is biased

   (c) The estimate of $V^{\pi_e}$ computed by weighted importance sampling is not consistent in finite horizon, bounded reward MDPs.

   Solution: (a), (b). IS is unbiased, WIS is biased, WIS is consistent – from lecture slides.

7. - **2 pts** Define $n$ as number of times an arm is pulled, and $i$ marks the index of the arm. According to UCB, we can construct a confidence interval $[\hat{\theta}_a^n, \hat{\theta}_b^n]$ after observing $n$ rewards by pulling an arm: $(r_1^i, r_2^i, ..., r_n^i)$. We assume each reward is binary $\{0,1\}$ and distributed as a Bernoulli random variable with $\theta_i$, for the arm $i$. Will $\theta_i \in [\hat{\theta}_a^n, \hat{\theta}_b^n]$ always be true $\forall n > 1, \forall i$?

   (a) True

   (b) False

   Solution: False. The upper bound estimated will be guaranteed with a probability (by Hoeffding bound). A counterexample is: if for the first 10 steps, you observe all 0s from Arm 1, while the true $\theta$ for Arm 1 is 0.99. It's very unlikely to happen, but it could still happen. In this case, your UCB upper bound will not contain the true parameter.

8. - **4 pts** There are two policies $\pi_e$ and $\pi_b$. You have data from executing $\pi_b$ on two trajectories. There are 2 unique actions $a_1$ and $a_2$. **All trajectories in this domain are only 1 action long (meaning the agent takes one action and then the trajectory always terminates).** $\pi(a|s)$ is the probability of policy $\pi$ taking action $a$ given state $s$. Rewards are bounded $r \in [0,1]$. The first trajectory is:

   - $(s_1, a_1, r = 0)$, where $\pi_b(a_1|s_1) = 0.5, \pi_b(a_2|s_1) = 0.5, \pi_e(a_1|s_1) = 0.1, \pi_e(a_2|s_1) = 0.9)$

   The second trajectory is:

   - $(s_2, a_1, r = 1)$, where $\pi_b(a_1|s_2) = 0.05, \pi_b(a_2|s_2) = 0.95, \pi_e(a_1|s_2) = 0.9, \pi_e(a_2|s_2) = 0.1)$

   Use importance sampling to estimate $V^{\pi_e}$. What is the value of $V^{\pi_e}$?

   (a) undefined

   (b) 0.5

   (c) 1

   (d) 5

   (e) 9

   (f) 18

**Short answer:** Will $\pi_e$ achieve this value when we executed it on the underlying MDP described in the problem statement. Why or why not? (Find your answer in the problem statement to answer this question – "general" answers based on the property of the estimator will not give you points)

Is there another IS estimator discussed in class that might yield a better estimate of $V^{\pi_e}$, and if so, why?

Solution: (e). Calculation: $V^{\pi_e} = \frac{1}{2}(\frac{0.1}{0.5} * 0 + \frac{0.9}{0.05} * 1) = 9$ (Note that for an IS estimator you have to divide by 2). $\pi_e$ will not achieve this reward of 9 because in this MDP, all trajectories terminate after 1 action, and reward is bounded between [0, 1]. The maximal achievable reward is 1. 9 is not possible. Some other IS estimators that make different tradeoffs of bias and variance can yield better mean squared error, such as weighted importance sampling estimator that has lower variance. Note that in this MDP, since all trajectories have the length of 1, PDIS is the same as IS; therefore it wouldn't help even if we collect more data.

## Problem 2 [10 points]

We will be analyzing the behavior of various multi-armed bandit algorithms. Let us consider a 3-armed bandit where $\mathcal{A} = \{1, 2, 3\}$. **Answer and provide a short justification for each question below**. [Hint: Recall that pure greedy, $\epsilon$-greedy, and UCB all begin by sampling each arm once, which corresponds to the first 3 entries in the table.]

| $t$: Timestep | $a_t$: Action | $r_t$: Observed reward after taking action $a_t$ |
|---|---|---|
| 1 | 2 | 0 |
| 2 | 1 | 1 |
| 3 | 3 | 1 |
| 4 | 1 | 1 |
| 5 | 1 | 1 |

1. - **2pts.** Could pure greedy have generated timesteps 4 and 5 above?

   Solution: True. The first three steps are from sampling each action. At $t = 4$, action 1 and action 2 are tied for the highest $\hat{Q}(a) = 1$. Either action 1 or 2 could be selected under the greedy method. Likewise, at $t = 5$, action 1 and action 2 both still have the same $\hat{Q}(a) = 1$ and either action could be selected

2. - **2pts.** Could $\epsilon$-greedy with $\epsilon > 0$ have generated timesteps 4 and 5 above?

   Solution: True. With a non-zero $\epsilon$, any sequence of 5 pulls is possible.

3. - **2pts.** Assume the reward is known to be bounded between 0 and 1 for all arms. Could the Upper Confidence Bound (UCB1) bandit algorithm have generated timesteps 4 and 5 above?

   Solution: False. Recall UCB1 calculates an upper bound on the reward of each arm after $t$ total pulls, where $N_t(a_i)$ is the number of times arm $a_i$ has been pulled at that point. Note, $t = \sum_a N_t(a)$. UCB1 computes a UCB on arm $a_i$ as:

   $$UCB_t(a_i) = \hat{Q}_t(a_i) + \sqrt{\frac{2\ln(t)}{N_t(a_i)}} \tag{1}$$

   Therefore after three arm pulls, $UCB_t = [1 + \sqrt{\frac{2\ln(t)}{1}}, 0 + \sqrt{\frac{2\ln(t)}{1}}, 1 + \sqrt{\frac{2\ln(t)}{1}}]$. Arm 1 and arm 2 have identical UCB so either would be fine to pull. Assume, as in the table, arm 1 is pulled. Now the new UCB at this point will be $UCB_t = [1 + \sqrt{\frac{2\ln(t)}{2}}, 0 + \sqrt{\frac{2\ln(t)}{1}}, 1 + \sqrt{\frac{2\ln(t)}{1}}]$ Note that in the UCB1, the numerator inside the square root term is identical across all arms. But now the denominator inside the square root for arm 1 $N_t(a_1) = 2$, whereas the denominator inside the square root for arm 3 is $N_t(a_3) = 1$. Therefore UCB1 would pull arm 3 at this point since its upper bound is higher. Therefore UCB1 could not have generated the observed sequence. Note that it is not necessary to recall the exact expression for the numerator inside the square root to get this question correct: the important thing is to notice that arm 1 and arm 3 will have identical means but different counts, and so UCB1 will compute a higher UCB for arm 3.

4. - **2pts.** Assume the reward for each arm is sampled from independent Bernoulli distributions. We initialize our priors for the probability of each arm receiving reward as $Beta(1, 1)$. With these priors, could Thompson sampling (TS) have generated timesteps 4 and 5 above?

   Solution: True. After all three arms have been pulled, the posterior for each arm is Beta(2,1), Beta(1,2), Beta(2,1), respectively. There are many ways to sample from these distributions such that arm 1 would be the best arm to pull on t=4. After arm 1 is pulled on t=5, the new posterior over theta for each arm is (Beta(3, 1), Beta(1, 2), Beta(2, 1)). Again, it is possible to same theta parameters from each posterior so that arm 1 would be the best arm to sample next. For example, we could have sampled $\theta_1 = 0.8, \theta_2 = 0.2, \theta_3 = 0.7$.

5. - **2pts.** Assume the reward for each arm is sampled from independent Bernoulli distributions. We initialize our priors for arm 1 and 2 as $Beta(1, 1)$, and the prior for the mean of arm 3 as $Beta(5, 1)$. With these priors, could Thompson sampling (TS) have generated timesteps 4 and 5 above?

   Solution: True. After all three arms have been pulled, the posterior for each arm is Beta(2,1), Beta(1,2), Beta(6,1), respectively. There are many ways to sample from these distributions such that arm 1 would be the best arm to pull on t=4. After arm 1 is pulled on t=5, the new posterior over theta for each arm is (Beta(3, 1), Beta(1, 2), Beta(6, 1)). Again, it is possible to same theta parameters from each posterior so that arm 1 would be the best arm to sample next. For example, we could have sampled $\theta_1 = 0.9, \theta_2 = 0.2, \theta_3 = 0.8$.

## Problem 3 [15 points]

Consider an infinite-horizon, discounted MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma \rangle$ where $\gamma \in [0, 1)$ and the state-action space is finite ($|\mathcal{S} \times \mathcal{A}| < \infty$). We will consider a parameterized policy $\pi_\theta(a \mid s)$ that denotes the probability of taking action $a \in \mathcal{A}$ from state $s \in \mathcal{S}$ with parameter vector $\theta \in \mathbb{R}^d$ for arbitrary $d > 1$. We let $d^{\pi_\theta}(s)$ is the stationary visitation probability of policy $\pi_\theta$ to state $s \in \mathcal{S}$. At any point in this question, you may denote the Fisher information matrix as

$$\mathcal{I}(\theta) = \mathbb{E}_{s \sim d^{\pi_\theta}} \left[ \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ \nabla_\theta \log \left( \pi_\theta(a \mid s) \right) \nabla_\theta \log \left( \pi_\theta(a \mid s) \right)^\top \right] \right].$$

A key challenge in designing efficient policy-gradient methods is combating the high variance of the gradient estimates, and in this problem you will explore which baselines may best minimize this variance. For each part of this problem, any term(s) that can be simplified to a numerical constant **must be simplified** to receive full credit.

Solution: The results of this question are due to Bhatnagar et al. [2009].

1. - **6 pts.** Let $\phi(s, a) = \nabla_\theta \log \left( \pi_\theta(a \mid s) \right) \in \mathbb{R}^d$ be the feature vector associated with any state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ and define a linear value-function approximator $\widehat{Q}^{\pi_\theta}_\omega(s, a) = \omega^\top \phi(s, a)$ with parameter vector $\omega \in \mathbb{R}^d$. For some fixed, arbitrary baseline function $b : \mathcal{S} \to \mathbb{R}$, find the minimizer $\omega^\star$ of the objective

$$\mathcal{L}(\omega) = \mathbb{E}_{s \sim d^{\pi_\theta}} \left[ \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ \left( Q^{\pi_\theta}(s, a) - \widehat{Q}^{\pi_\theta}_\omega(s, a) - b(s) \right)^2 \right] \right].$$

Solution:

$$\begin{aligned}
\nabla_\omega \mathcal{L}(\omega) &= \nabla_\omega \mathbb{E}_{s \sim d^{\pi_\theta}} \left[ \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ \left( Q^{\pi_\theta}(s, a) - \widehat{Q}^{\pi_\theta}_\omega(s, a) - b(s) \right)^2 \right] \right] \\
&= -2 \mathbb{E}_{s \sim d^{\pi_\theta}} \left[ \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ \left( Q^{\pi_\theta}(s, a) - \widehat{Q}^{\pi_\theta}_\omega(s, a) - b(s) \right) \phi(s, a) \right] \right] \\
&= -2 \mathbb{E}_{s \sim d^{\pi_\theta}} \left[ \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ \left( Q^{\pi_\theta}(s, a) - \omega^\top \phi(s, a) - b(s) \right) \phi(s, a) \right] \right].
\end{aligned}$$

Setting the gradient equal to zero and dividing through by -2 yields

$$\mathbb{E}_{s \sim d^{\pi_\theta}} \left[ \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ \left( Q^{\pi_\theta}(s, a) - \omega^\top \phi(s, a) - b(s) \right) \phi(s, a) \right] \right] = 0.$$

Re-arranging terms yields

$$\mathbb{E}_{s \sim d^{\pi_\theta}} \left[ \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ \omega^\top \phi(s, a) \phi(s, a) \right] \right] + \mathbb{E}_{s \sim d^{\pi_\theta}} \left[ \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ b(s) \phi(s, a) \right] \right] = \mathbb{E}_{s \sim d^{\pi_\theta}} \left[ \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ Q^{\pi_\theta}(s, a) \phi(s, a) \right] \right].$$

First, observe that

$$\begin{aligned}
\mathbb{E}_{s \sim d^{\pi_\theta}} \left[ \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ \omega^\top \phi(s, a) \phi(s, a) \right] \right] &= \mathbb{E}_{s \sim d^{\pi_\theta}} \left[ \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ \phi(s, a) \phi(s, a)^\top \omega \right] \right] \\
&= \mathbb{E}_{s \sim d^{\pi_\theta}} \left[ \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ \phi(s, a) \phi(s, a)^\top \right] \right] \omega \\
&= \mathcal{I}(\theta) \omega.
\end{aligned}$$

Next, observe that

$$\begin{aligned}
\mathbb{E}_{s \sim d^{\pi_\theta}} \left[ \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ b(s) \phi(s, a) \right] \right] &= \mathbb{E}_{s \sim d^{\pi_\theta}} \left[ b(s) \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ \phi(s, a) \right] \right] \\
&= \mathbb{E}_{s \sim d^{\pi_\theta}} \left[ b(s) \sum_{a \in \mathcal{A}} \pi_\theta(a \mid s) \nabla_\theta \log \left( \pi_\theta(a \mid s) \right) \right] \\
&= \mathbb{E}_{s \sim d^{\pi_\theta}} \left[ b(s) \sum_{a \in \mathcal{A}} \nabla_\theta \pi_\theta(a \mid s) \right] \\
&= \mathbb{E}_{s \sim d^{\pi_\theta}} \left[ b(s) \nabla_\theta \sum_{a \in \mathcal{A}} \pi_\theta(a \mid s) \right] \\
&= \mathbb{E}_{s \sim d^{\pi_\theta}} \left[ b(s) \nabla_\theta 1 \right] = 0.
\end{aligned}$$

Putting everything together, we have

$$\mathcal{I}(\theta)\omega = \mathbb{E}_{s\sim d^{\pi_\theta}}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[Q^{\pi_\theta}(s,a)\phi(s,a)\right]\right] = \mathbb{E}_{s\sim d^{\pi_\theta}}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[Q^{\pi_\theta}(s,a)\nabla_\theta \log\left(\pi_\theta(a\mid s)\right)\right]\right].$$

Left multiplying by the inverse of the Fisher information on both sides yields

$$\omega^\star = \mathcal{I}(\theta)^{-1}\mathbb{E}_{s\sim d^{\pi_\theta}}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[Q^{\pi_\theta}(s,a)\nabla_\theta \log\left(\pi_\theta(a\mid s)\right)\right]\right].$$

2. - **3 pts.** **True or False?** For all states $s \in \mathcal{S}$ and for the minimizer $\omega^\star$ from part (1),

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\widehat{Q}^{\pi_\theta}_{\omega^\star}(s,a)\right] = \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\omega^{\star\top}\phi(s,a)\right] > 0.$$

Include a short explanation for your answer.

Solution: False. Following a calculation identical to the one in the previous part, we have

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\widehat{Q}^{\pi_\theta}_{\omega^\star}(s,a)\right] = \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\omega^{\star\top}\phi(s,a)\right] = \omega^{\star\top}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\phi(s,a)\right] = 0.$$

3. - **6 pts.** Let $\omega^\star$ be the (fixed) minimizer from part (1). Assuming that $d^{\pi_\theta}(s) > 0$ for all $s \in \mathcal{S}$, find the optimal baseline function $b^\star$ that minimizes the objective

$$\mathcal{L}(\omega^\star, b) = \mathbb{E}_{s\sim d^{\pi_\theta}}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\left(Q^{\pi_\theta}(s,a) - \widehat{Q}^{\pi_\theta}_{\omega^\star}(s,a) - b(s)\right)^2\right]\right].$$

Solution: Fix an arbitrary state $\overline{s} \in \mathcal{S}$. Taking the partial derivative, we have

$$\begin{aligned}\frac{\partial}{\partial b(\overline{s})}\mathcal{L}(\omega^\star, b) &= \frac{\partial}{\partial b(\overline{s})}\mathbb{E}_{s\sim d^{\pi_\theta}}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\left(Q^{\pi_\theta}(s,a) - \widehat{Q}^{\pi_\theta}_{\omega^\star}(s,a) - b(s)\right)^2\right]\right] \\ &= -2d^{\pi_\theta}(\overline{s})\cdot\mathbb{E}_{a\sim\pi_\theta(\cdot|\overline{s})}\left[Q^{\pi_\theta}(\overline{s},a) - \widehat{Q}^{\pi_\theta}_{\omega^\star}(\overline{s},a) - b(\overline{s})\right].\end{aligned}$$

Setting equal to zero and dividing through by $-2d^{\pi_\theta}(\overline{s})$ yields

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|\overline{s})}\left[Q^{\pi_\theta}(\overline{s},a) - \widehat{Q}^{\pi_\theta}_{\omega^\star}(\overline{s},a) - b(\overline{s})\right] = \mathbb{E}_{a\sim\pi_\theta(\cdot|\overline{s})}\left[Q^{\pi_\theta}(\overline{s},a) - \widehat{Q}^{\pi_\theta}_{\omega^\star}(\overline{s},a)\right] - b(\overline{s}) = 0.$$

From the previous part, we know that

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\widehat{Q}^{\pi_\theta}_{\omega^\star}(s,a)\right] = 0,$$

so we have

$$b^\star(\overline{s}) = \mathbb{E}_{a\sim\pi_\theta(\cdot|\overline{s})}\left[Q^{\pi_\theta}(\overline{s},a)\right] = V^{\pi_\theta}(\overline{s}).$$

# References

Shalabh Bhatnagar, Richard S Sutton, Mohammad Ghavamzadeh, and Mark Lee. Natural actor-critic algorithms. *Automatica (Journal of IFAC)*, 45(11):2471–2482, 2009.