ESE 6510 Final Review

Jefferson

Questions We will Look At

1. Sample Question Made by Vaibhav
2. Question 1 of Midterm
3. Question 4 of Midterm

Q1 Sim2Real Transfer For Drone Racing

You are training an agile drone racing policy using behavioral cloning from an algorithmic expert (a model-predictive controller with access to ground-truth state in simulation). The policy takes onboard camera images and IMU data as input and outputs desired collective thrust and body-rate commands. You plan to deploy on a physical quadrotor.

Q1 Sim2Real Transfer For Drone Racing

(a) **(8 pts) Observation gap.** In simulation, the algorithmic expert observes the full privileged state $s^{\text{priv}} = (p, v, R, \omega, \text{gate poses})$, where p, v are position and velocity, R is the rotation matrix, ω is body rate, and gate poses are known exactly. The learned student policy, however, only observes $o = (I_t, \omega_t)$, i.e., an RGB image and IMU gyro reading.

Q1 Sim2Real Transfer For Drone Racing

This Course

(a) **(8 pts) Observation gap.** In simulation, the algorithmic expert observes the full privileged state $s^{\text{priv}} = (p, v, R, \omega, \text{gate poses})$, where p, v are position and velocity, R is the rotation matrix, ω is body rate, and gate poses are known exactly. The learned student policy, however, only observes $o = (I_t, \omega_t)$, i.e., an RGB image and IMU gyro reading.

Q1 Sim2Real Transfer For Drone Racing

This Course

(a) **(8 pts) Observation gap.** In simulation, the algorithmic expert observes the full privileged state $s^{\text{priv}} = (p, v, R, \omega, \text{gate poses})$, where p, v are position and velocity, R is the rotation matrix, ω is body rate, and gate poses are known exactly. The learned student policy, however, only observes $o = (I_t, \omega_t)$, i.e., an RGB image and IMU gyro reading.

What could help scale it up

Q1 Sim2Real Transfer For Drone Racing

(a) **(8 pts) Observation gap.** In simulation, the algorithmic expert observes the full privileged state $s^{\text{priv}} = (p, v, R, \omega, \text{gate poses})$, where p, v are position and velocity, R is the rotation matrix, ω is body rate, and gate poses are known exactly. The learned student policy, however, only observes $o = (I_t, \omega_t)$, i.e., an RGB image and IMU gyro reading.

This is how teacher-student Distillation work

Q1 Sim2Real Transfer For Drone Racing

(a) **(8 pts) Observation gap.** In simulation, the algorithmic expert observes the full privileged state $s^{\text{priv}} = (p, v, R, \omega, \text{gate poses})$, where p, v are position and velocity, R is the rotation matrix, ω is body rate, and gate poses are known exactly. The learned student policy, however, only observes $o = (I_t, \omega_t)$, i.e., an RGB image and IMU gyro reading.

- Define formally what it means for o to be a *sufficient statistic* for action selection, in terms of the conditional distribution $p(a^* \mid s^{\text{priv}})$ vs. $p(a^* \mid o)$.

Q1 Sim2Real Transfer For Drone Racing

(a) **(8 pts) Observation gap.** In simulation, the algorithmic expert observes the full privileged state $s^{\text{priv}} = (p, v, R, \omega, \text{gate poses})$, where p, v are position and velocity, R is the rotation matrix, ω is body rate, and gate poses are known exactly. The learned student policy, however, only observes $o = (I_t, \omega_t)$, i.e., an RGB image and IMU gyro reading.

- Define formally what it means for o to be a *sufficient statistic* for action selection, in terms of the conditional distribution $p(a^* \mid s^{\text{priv}})$ vs. $p(a^* \mid o)$.

  For this, it really means do you think RGB + IMU gives same info as full state implicitly

Q1 Sim2Real Transfer For Drone Racing

(a) **(8 pts) Observation gap.** In simulation, the algorithmic expert observes the full privileged state $s^{\text{priv}} = (p, v, R, \omega, \text{gate poses})$, where p, v are position and velocity, R is the rotation matrix, ω is body rate, and gate poses are known exactly. The learned student policy, however, only observes $o = (I_t, \omega_t)$, i.e., an RGB image and IMU gyro reading.

- Define formally what it means for o to be a *sufficient statistic* for action selection, in terms of the conditional distribution $p(a^* \mid s^{\text{priv}})$ vs. $p(a^* \mid o)$.

$$p(a^* \mid s^{priv}) = p(a^* \mid o)$$

Q1 Sim2Real Transfer For Drone Racing

(a) **(8 pts) Observation gap.** In simulation, the algorithmic expert observes the full privileged state $s^{\text{priv}} = (p, v, R, \omega, \text{gate poses})$, where p, v are position and velocity, R is the rotation matrix, ω is body rate, and gate poses are known exactly. The learned student policy, however, only observes $o = (I_t, \omega_t)$, i.e., an RGB image and IMU gyro reading.

- The velocity v is not directly observable from a single image frame. Propose **one architectural modification** to the student policy that could allow it to implicitly estimate velocity without access to a separate velocity estimator, and justify your proposal in terms of the information contained in the input.

Q1 Sim2Real Transfer For Drone Racing

(a) **(8 pts) Observation gap.** In simulation, the algorithmic expert observes the full privileged state $s^{\text{priv}} = (p, v, R, \omega, \text{gate poses})$, where p, v are position and velocity, R is the rotation matrix, ω is body rate, and gate poses are known exactly. The learned student policy, however, only observes $o = (I_t, \omega_t)$, i.e., an RGB image and IMU gyro reading.

- The velocity v is not directly observable from a single image frame. Propose **one architectural modification** to the student policy that could allow it to implicitly estimate velocity without access to a separate velocity estimator, and justify your proposal in terms of the information contained in the input.

LSTM layer to add memory, or in the observation you can have a history of inputs

Q1 Sim2Real Transfer For Drone Racing

(b) **(12 pts) Domain randomization and its limits.** To improve robustness, you apply domain randomization (DR) during training by sampling simulation parameters $\xi \sim p(\xi)$ (e.g., motor time constants, drag coefficients, visual textures). Let $\mathcal{L}_\xi(\theta)$ denote the BC loss under parameters ξ:

$$\mathcal{L}_\xi(\theta) = -\mathbb{E}_{(s,a^*)\sim\mathcal{D}_\xi}[\log \pi_\theta(a^* \mid o(s,\xi))].$$

The DR objective is

$$\mathcal{L}_{\mathrm{DR}}(\theta) = \mathbb{E}_{\xi\sim p(\xi)}[\mathcal{L}_\xi(\theta)].$$

One note is that you should probably randomize the RGB observation heavily but not included for this question.

Q1 Sim2Real Transfer For Drone Racing

(b) **(12 pts) Domain randomization and its limits.** To improve robustness, you apply domain randomization (DR) during training by sampling simulation parameters $\xi \sim p(\xi)$ (e.g., motor time constants, drag coefficients, visual textures). Let $\mathcal{L}_\xi(\theta)$ denote the BC loss under parameters ξ:

$$\mathcal{L}_\xi(\theta) = -\mathbb{E}_{(s,a^*)\sim\mathcal{D}_\xi}[\log \pi_\theta(a^* \mid o(s,\xi))].$$

The DR objective is

$$\mathcal{L}_{\mathrm{DR}}(\theta) = \mathbb{E}_{\xi\sim p(\xi)}[\mathcal{L}_\xi(\theta)].$$

- Show that minimizing $\mathcal{L}_{\mathrm{DR}}(\theta)$ is equivalent to minimizing the expected KL divergence between the expert distribution and the student distribution *under the mixture* $q(s) = \mathbb{E}_\xi[q_\xi(s)]$, where $q_\xi(s)$ is the marginal state distribution under parameters ξ. (Hint: expand the log and use linearity of expectation.)

Q1 Sim2Real Transfer For Drone Racing

Measuring how well the student assigns high probability to expert's action under certain randomized parameter

$$\mathcal{L}_\xi(\theta) = -\mathbb{E}_{(s,a^*)\sim\mathcal{D}_\xi}[\log \pi_\theta(a^* \mid o(s,\xi))].$$

How well does it imitate expert on average across all parameters

$$\mathcal{L}_{\mathrm{DR}}(\theta) = \mathbb{E}_{\xi\sim p(\xi)}[\mathcal{L}_\xi(\theta)].$$

Q1 Sim2Real Transfer For Drone Racing

Measuring how well the student assigns high probability to expert's action under certain randomized parameter

$$\mathcal{L}_\xi(\theta) = -\mathbb{E}_{(s,a^*)\sim\mathcal{D}_\xi}[\log \pi_\theta(a^* \mid o(s,\xi))].$$

How well does it imitate expert on average across all parameters

$$\mathcal{L}_{\mathrm{DR}}(\theta) = \mathbb{E}_{\xi\sim p(\xi)}[\mathcal{L}_\xi(\theta)].$$

Q1 Sim2Real Transfer For Drone Racing

$$L_\xi(\theta) = -\mathbb{E}_{(s,a^*)\sim D_\xi}\left[\log \pi_\theta(a^* \mid o(s,\xi))\right].$$

Here we rewrite this distribution using factorization of expert data joint distribution

$$p_\xi(s,a^*) = q_\xi(s)\pi_E(a^* \mid s)$$

Q1 Sim2Real Transfer For Drone Racing

$$L_\xi(\theta) = -\mathbb{E}_{(s,a^*)\sim D_\xi}\left[\log \pi_\theta(a^* \mid o(s,\xi))\right].$$

$$L_\xi(\theta) = -\mathbb{E}_{s\sim q_\xi(s)}\left[\sum_a \pi_E(a \mid s) \log \pi_\theta(a \mid o(s,\xi))\right].$$

Q1 Sim2Real Transfer For Drone Racing

$$L_\xi(\theta) = -\mathbb{E}_{s\sim q_\xi(s)}\left[\sum_a \pi_E(a \mid s)\log \pi_\theta(a \mid o(s,\xi))\right].$$

For this question you will need to know this identity

$$-\sum_a p(a)\log q(a) = H(p) + \mathrm{KL}(p\|q),$$

Q1 Sim2Real Transfer For Drone Racing

$$L_\xi(\theta) = -\mathbb{E}_{s\sim q_\xi(s)}\left[\sum_a \pi_E(a \mid s)\log \pi_\theta(a \mid o(s,\xi))\right].$$

Now using

$$p(a) = \pi_E(a \mid s), \qquad q(a) = \pi_\theta(a \mid o(s,\xi)),$$

Q1 Sim2Real Transfer For Drone Racing

$$L_\xi(\theta) = -\mathbb{E}_{s\sim q_\xi(s)}\left[\sum_a \pi_E(a \mid s)\log \pi_\theta(a \mid o(s,\xi))\right].$$

We get

$$L_\xi(\theta) = \mathbb{E}_{s\sim q_\xi(s)}\left[H(\pi_E(\cdot \mid s)) + \mathrm{KL}(\pi_E(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid o(s,\xi)))\right].$$

Q1 Sim2Real Transfer For Drone Racing

$$L_\xi(\theta) = \mathbb{E}_{s\sim q_\xi(s)}\left[H(\pi_E(\cdot \mid s)) + \mathrm{KL}(\pi_E(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid o(s,\xi)))\right].$$

Taking this average

$$L_{\mathrm{DR}}(\theta) = \mathbb{E}_{\xi\sim p(\xi)}[L_\xi(\theta)].$$

Q1 Sim2Real Transfer For Drone Racing

$$L_\xi(\theta) = \mathbb{E}_{s\sim q_\xi(s)}\left[H(\pi_E(\cdot \mid s)) + \mathrm{KL}(\pi_E(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid o(s,\xi)))\right].$$

Taking this average

$$L_{\mathrm{DR}}(\theta) = \mathbb{E}_{\xi\sim p(\xi)}[L_\xi(\theta)].$$

$$L_{\mathrm{DR}}(\theta) = \mathbb{E}_{\xi\sim p(\xi)}\mathbb{E}_{s\sim q_\xi(s)}\left[H(\pi_E(\cdot \mid s)) + \mathrm{KL}(\pi_E(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid o(s,\xi)))\right].$$

Q1 Sim2Real Transfer For Drone Racing

$$L_{\mathrm{DR}}(\theta) = \mathbb{E}_{\xi \sim p(\xi)} \mathbb{E}_{s \sim q_\xi(s)} \left[H(\pi_E(\cdot \mid s)) + \mathrm{KL}(\pi_E(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid o(s, \xi))) \right].$$

With linearity of expectations

$$L_{\mathrm{DR}}(\theta) = \mathbb{E}_{\xi, s \sim q_\xi} \left[H(\pi_E(\cdot \mid s)) \right] + \mathbb{E}_{\xi, s \sim q_\xi} \left[\mathrm{KL}(\pi_E(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid o(s, \xi))) \right].$$

Q1 Sim2Real Transfer For Drone Racing

$$L_{\mathrm{DR}}(\theta) = \mathbb{E}_{\xi, s\sim q_\xi}\left[H(\pi_E(\cdot \mid s))\right] + \mathbb{E}_{\xi, s\sim q_\xi}\left[\mathrm{KL}(\pi_E(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid o(s,\xi)))\right].$$

Not dependent on theta

Q1 Sim2Real Transfer For Drone Racing

$$L_{\mathrm{DR}}(\theta) = \mathbb{E}_{\xi, s\sim q_\xi}\left[H(\pi_E(\cdot \mid s))\right] + \mathbb{E}_{\xi, s\sim q_\xi}\left[\mathrm{KL}(\pi_E(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid o(s,\xi)))\right].$$

Not dependent on theta, therefore

$$\arg\min_\theta L_{\mathrm{DR}}(\theta) = \arg\min_\theta \mathbb{E}_{\xi, s\sim q_\xi}\left[\mathrm{KL}(\pi_E(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid o(s,\xi)))\right].$$

Q1 Sim2Real Transfer For Drone Racing

(b) **(12 pts) Domain randomization and its limits.** To improve robustness, you apply domain randomization (DR) during training by sampling simulation parameters $\xi \sim p(\xi)$ (e.g., motor time constants, drag coefficients, visual textures). Let $\mathcal{L}_\xi(\theta)$ denote the BC loss under parameters ξ:

$$\mathcal{L}_\xi(\theta) = -\mathbb{E}_{(s,a^*)\sim\mathcal{D}_\xi}[\log \pi_\theta(a^* \mid o(s,\xi))].$$

The DR objective is

$$\mathcal{L}_{\mathrm{DR}}(\theta) = \mathbb{E}_{\xi\sim p(\xi)}[\mathcal{L}_\xi(\theta)].$$

- A student claims that if ξ^* (the real-world parameter) is in the support of $p(\xi)$, then $\mathcal{L}_{\mathrm{DR}}(\theta) = 0$ implies zero real-world error. Is this claim correct? Justify your answer by identifying what additional condition is required beyond $\xi^* \in \mathrm{supp}(p(\xi))$.

Q1 Sim2Real Transfer For Drone Racing

(b) **(12 pts) Domain randomization and its limits.** To improve robustness, you apply domain randomization (DR) during training by sampling simulation parameters $\xi \sim p(\xi)$ (e.g., motor time constants, drag coefficients, visual textures). Let $\mathcal{L}_\xi(\theta)$ denote the BC loss under parameters ξ:

$$\mathcal{L}_\xi(\theta) = -\mathbb{E}_{(s,a^*)\sim\mathcal{D}_\xi}[\log \pi_\theta(a^* \mid o(s,\xi))].$$

The DR objective is

$$\mathcal{L}_{\mathrm{DR}}(\theta) = \mathbb{E}_{\xi\sim p(\xi)}[\mathcal{L}_\xi(\theta)].$$

- A student claims that if ξ^* (the real-world parameter) is in the support of $p(\xi)$, then $\mathcal{L}_{\mathrm{DR}}(\theta) = 0$ implies zero real-world error. Is this claim correct? Justify your answer by identifying what additional condition is required beyond $\xi^* \in \mathrm{supp}(p(\xi))$.

Definitely not true, the simulated parameters are not accurate in real world so you are learning the behavior to adapt with these randomization

Q1 Sim2Real Transfer For Drone Racing

(c) **(15 pts) Latency and the Markov property.** Real deployment introduces a sensing-to-actuation latency of τ seconds due to image processing and communication delays (not present in simulation).

- The policy π_θ was trained assuming $a_t = \pi_\theta(o_t)$, i.e., the action at time t is a function of the *current* observation. With latency τ, the action a_t is instead computed from $o_{t-\tau}$. Using the Markov property, argue formally why this breaks the assumptions under which BC training was performed and can lead to instability at high speeds.

During deployment the true action is based on delayed actions

$$a_t = \pi_\theta(o_{t-\tau}).$$

Q1 Sim2Real Transfer For Drone Racing

- The policy π_θ was trained assuming $a_t = \pi_\theta(o_t)$, i.e., the action at time t is a function of the *current* observation. With latency τ, the action a_t is instead computed from $o_{t-\tau}$. Using the Markov property, argue formally why this breaks the assumptions under which BC training was performed and can lead to instability at high speeds.

In MDP we assume that future state only dependent on current state

$$p(s_{t+1} \mid s_{0:t}, a_{0:t}) = p(s_{t+1} \mid s_t, a_t),$$

Q1 Sim2Real Transfer For Drone Racing

- The policy π_θ was trained assuming $a_t = \pi_\theta(o_t)$, i.e., the action at time t is a function of the *current* observation. With latency τ, the action a_t is instead computed from $o_{t-\tau}$. Using the Markov property, argue formally why this breaks the assumptions under which BC training was performed and can lead to instability at high speeds.

So even if Ot might be sufficient for St, the delayed state might not give enough information to satisfy MDP condition anymore, but if the delay is short enough or with lstm the policy can still work since state changes are small but it still breaks MDP. At high speed it will probably overshoot or collide or fly like it's drunk

$$p(s_{t+1} \mid s_{0:t}, a_{0:t}) = p(s_{t+1} \mid s_t, a_t),$$

Q2 The question I made

Question 1 (15 points)

Consider an MDP with no terminal states, where the agent interacts with the environment indefinitely. Assume the rewards r_t are independent random variables satisfying

$$\mathbb{E}[r_t] = 0, \qquad \mathrm{Var}(r_t) = \sigma^2.$$

1. Suppose the infinite-horizon return is defined without discounting

$$G_t = \sum_{k=0}^{\infty} r_{t+k}.$$

   Is this return well-defined for such an infinite-horizon process? Compute the variance and determine what it converges to.

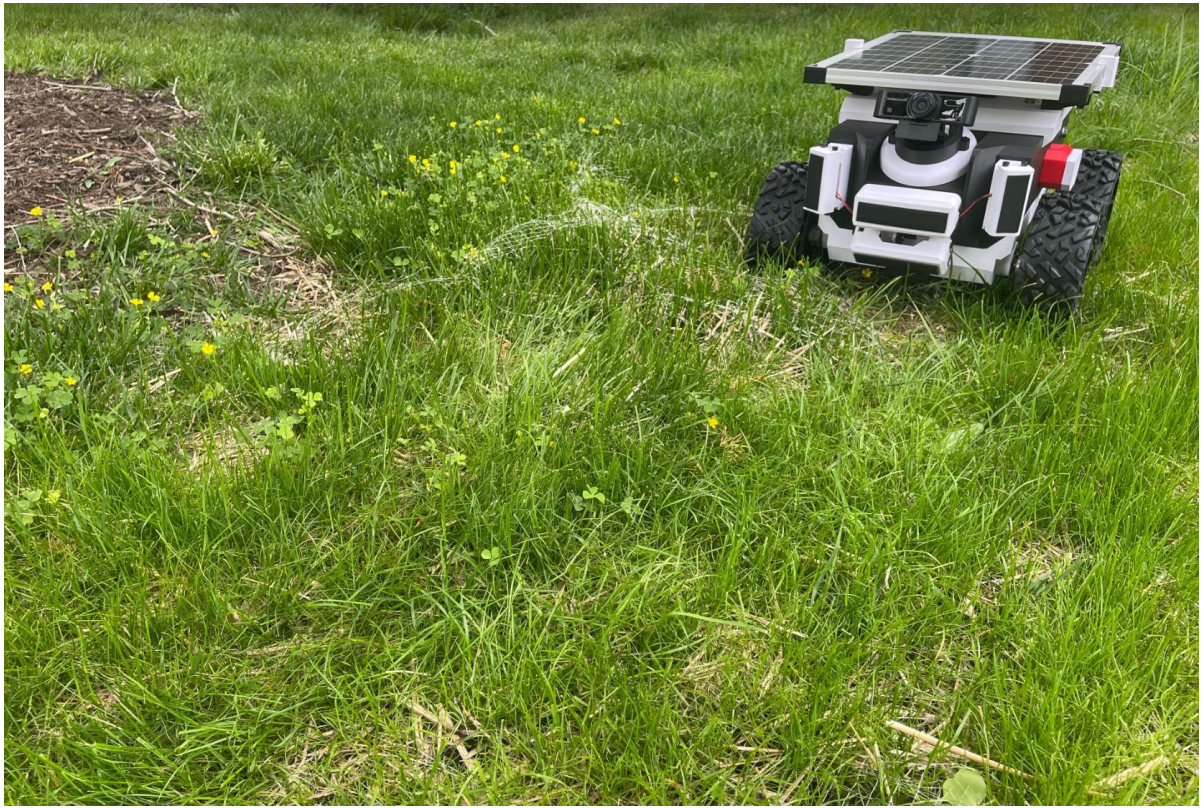
2. Now define the discounted return

$$G_{t,\gamma} = \sum_{k=0}^{\infty} \gamma^k r_{t+k}.$$

   Compute the variance for the new discounted return.

3. Discuss advantages and disadvantages of modeling reinforcement learning problems as infinite-horizon environments versus episodic environments with terminal states. Will one formulation produce a better policy than the other?

Q2 The question I made



Q2 The question I made

What happens when you have no termination condition, there is no resets and it just keeps surviving.

Q2 Infinite Horizon

Consider an MDP with no terminal states, where the agent interacts with the environment indefinitely. Assume the rewards r_t are independent random variables satisfying

$$\mathbb{E}[r_t] = 0, \qquad \mathrm{Var}(r_t) = \sigma^2.$$

1. Suppose the infinite-horizon return is defined without discounting

$$G_t = \sum_{k=0}^{\infty} r_{t+k}.$$

Is this return well-defined for such an infinite-horizon process? Compute the variance and determine what it converges to.

Q2 Infinite Horizon

Consider an MDP with no terminal states, where the agent interacts with the environment indefinitely. Assume the rewards r_t are independent random variables satisfying

$$\mathbb{E}[r_t] = 0, \qquad \mathrm{Var}(r_t) = \sigma^2.$$

1. Suppose the infinite-horizon return is defined without discounting

$$G_t = \sum_{k=0}^{\infty} r_{t+k}.$$

Is this return well-defined for such an infinite-horizon process? Compute the variance and determine what it converges to.

You will need to know how to compute the variance

Q2 Infinite Horizon

$$\mathbb{E}[r_t] = 0, \qquad \mathrm{Var}(r_t) = \sigma^2.$$

1. Suppose the infinite-horizon return is defined without discounting

$$G_t = \sum_{k=0}^{\infty} r_{t+k}.$$

$$\mathrm{Var}(G_t) = \mathrm{Var}\left(\sum_{k=0}^{\infty} r_{t+k}\right)$$

Q2 Infinite Horizon

$$\mathbb{E}[r_t] = 0, \qquad \mathrm{Var}(r_t) = \sigma^2.$$

1. Suppose the infinite-horizon return is defined without discounting

$$G_t = \sum_{k=0}^{\infty} r_{t+k}.$$

$$\mathrm{Var}(G_t) = \mathrm{Var}\left(\sum_{k=0}^{\infty} r_{t+k}\right) = \sum_{k=0}^{\infty} \mathrm{Var}(r_{t+k})$$

Assume reward are independent

Q2 Infinite Horizon

$$\mathbb{E}[r_t] = 0, \qquad \mathrm{Var}(r_t) = \sigma^2.$$

1. Suppose the infinite-horizon return is defined without discounting

$$G_t = \sum_{k=0}^{\infty} r_{t+k}.$$

$$\mathrm{Var}(G_t) = \mathrm{Var}\left(\sum_{k=0}^{\infty} r_{t+k}\right) = \sum_{k=0}^{\infty} \mathrm{Var}(r_{t+k}) = \sum_{k=0}^{\infty} \sigma^2.$$

Q2 Infinite Horizon

$$\mathbb{E}[r_t] = 0, \qquad \mathrm{Var}(r_t) = \sigma^2.$$

1. Suppose the infinite-horizon return is defined without discounting

$$G_t = \sum_{k=0}^{\infty} r_{t+k}.$$

$$\mathrm{Var}(G_t) = \mathrm{Var}\left(\sum_{k=0}^{\infty} r_{t+k}\right) = \sum_{k=0}^{\infty} \mathrm{Var}(r_{t+k}) = \sum_{k=0}^{\infty} \sigma^2.$$

$$\mathrm{Var}(G_t) = \sigma^2 \sum_{k=0}^{\infty} 1 = \infty.$$

Q2 Infinite Horizon

Hence this is not well defined

$$\mathrm{Var}(G_t) = \sigma^2 \sum_{k=0}^{\infty} 1 = \infty.$$

Q2 Infinite Horizon

2. Now define the discounted return

$$G_{t,\gamma} = \sum_{k=0}^{\infty} \gamma^k r_{t+k}.$$

Compute the variance for the new discounted return.

Q2 Infinite Horizon

2. Now define the discounted return

$$G_{t,\gamma} = \sum_{k=0}^{\infty} \gamma^k r_{t+k}.$$

Compute the variance for the new discounted return.

$$\mathrm{Var}(G_{t,\gamma}) = \sum_{k=0}^{\infty} \gamma^{2k} \mathrm{Var}(r_{t+k}) = \sum_{k=0}^{\infty} \gamma^{2k} \sigma^2 = \sigma^2 \sum_{k=0}^{\infty} \gamma^{2k}.$$

To move variance inside you need to multiply by another gamma^k

Q2 Infinite Horizon

Since it is geometric series you can also write

$$\mathrm{Var}(G_{t,\gamma}) = \frac{\sigma^2}{1-\gamma^2}, \qquad |\gamma| < 1.$$

Q2 Infinite Horizon

2. Now define the discounted return

$$G_{t,\gamma} = \sum_{k=0}^{\infty} \gamma^k r_{t+k}.$$

Compute the variance for the new discounted return.

It converges

Q2 Infinite Horizon

2. Now define the discounted return

$$G_{t,\gamma} = \sum_{k=0}^{\infty} \gamma^k r_{t+k}.$$

Compute the variance for the new discounted return.

It converges

Q2 Infinite Horizon

3. Discuss advantages and disadvantages of modeling reinforcement learning problems as infinite-horizon environments versus episodic environments with terminal states. Will one formulation produce a better policy than the other?

Infinite horizon better aligns with real task (low bias, high variance)

With artificial resets you get high bias but low variance

Neither is better

Q2 Infinite Horizon

3. Discuss advantages and disadvantages of modeling reinforcement learning problems as infinite-horizon environments versus episodic environments with terminal states. Will one formulation produce a better policy than the other?

Infinite horizon better aligns with real task (low bias, high variance)

With artificial resets you get high bias but low variance

Neither is better

Q3

Consider a one-state bandit MDP with three actions $\{a_1, a_2, a_3\}$ and deterministic rewards $r(a_1) = 4$, $r(a_2) = 1$, $r(a_3) = 0$. The policy is parameterized via softmax:

$$\pi_\theta(a_i) = \frac{e^{\theta_i}}{\sum_{j=1}^{3} e^{\theta_j}}, \qquad \theta = (\theta_1, \theta_2, \theta_3).$$

We run REINFORCE to update our policy. The single-sample gradient after taking action a_k with reward $r(a_k)$ is:

$$\nabla_\theta J = r(a_k) \cdot \nabla_\theta \log \pi_\theta(a_k).$$

(a) Compute $\frac{\partial}{\partial \theta_i} \log \pi_\theta(a_k)$ for the softmax policy above. Express your answer in terms of the policy probabilities $\pi_\theta(a_i)$ and an indicator $\mathbf{1}[i = k]$.

(b) Using your result from part (a), write out the expected policy gradient $\mathbb{E}_{a \sim \pi_\theta}[\nabla_\theta J]$. Be sure to show all work.

(c) A student claims: "Since all rewards are non-negative, the REINFORCE update always increases θ_k for the sampled action. Therefore the algorithm can never learn to avoid a suboptimal action." Is this a true statement? Using your expression from part (b), compute $\mathbb{E}[\nabla_\theta J_2]$ (the expected update to the parameter for action a_2) to support your answer.

Q3

Consider a one-state bandit MDP with three actions $\{a_1, a_2, a_3\}$ and deterministic rewards $r(a_1) = 4$, $r(a_2) = 1$, $r(a_3) = 0$. The policy is parameterized via softmax:

$$\pi_\theta(a_i) = \frac{e^{\theta_i}}{\sum_{j=1}^{3} e^{\theta_j}}, \qquad \theta = (\theta_1, \theta_2, \theta_3).$$

We run REINFORCE to update our policy. The single-sample gradient after taking action a_k with reward $r(a_k)$ is:

$$\nabla_\theta J = r(a_k) \cdot \nabla_\theta \log \pi_\theta(a_k).$$

(a) Compute $\frac{\partial}{\partial \theta_i} \log \pi_\theta(a_k)$ for the softmax policy above. Express your answer in terms of the policy probabilities $\pi_\theta(a_i)$ and an indicator $\mathbf{1}[i = k]$.

Q3

$$\pi_\theta(a_k) = \frac{e^{\theta_k}}{\sum_{j=1}^{3} e^{\theta_j}}.$$

Q3

$$\pi_\theta(a_k) = \frac{e^{\theta_k}}{\sum_{j=1}^{3} e^{\theta_j}}.$$

We first take the log

$$\log \pi_\theta(a_k) = \theta_k - \log\left(\sum_{j=1}^{3} e^{\theta_j}\right).$$

Q3

$$\pi_\theta(a_k) = \frac{e^{\theta_k}}{\sum_{j=1}^{3} e^{\theta_j}}.$$

We first take the log

$$\log \pi_\theta(a_k) = \theta_k - \log\left(\sum_{j=1}^{3} e^{\theta_j}\right).$$

Differentiating theta gives the following since $\theta = (\theta_1, \theta_2, \theta_3)$.

$$\frac{\partial}{\partial \theta_i} \theta_k = \mathbf{1}[i = k].$$

Q3

$$\pi_\theta(a_k) = \frac{e^{\theta_k}}{\sum_{j=1}^{3} e^{\theta_j}}.$$

We first take the log

$$\log \pi_\theta(a_k) = \theta_k - \log\left(\sum_{j=1}^{3} e^{\theta_j}\right).$$

Differentiating theta gives the following since

$$\frac{\partial}{\partial \theta_i}\theta_k = \mathbf{1}[i = k].$$

Log differentiation

$$\frac{\partial}{\partial \theta_i}\log\left(\sum_{j=1}^{3} e^{\theta_j}\right) = \frac{1}{\sum_{j=1}^{3} e^{\theta_j}} \cdot \frac{\partial}{\partial \theta_i}\sum_{j=1}^{3} e^{\theta_j} = \frac{e^{\theta_i}}{\sum_{j=1}^{3} e^{\theta_j}} = \pi_\theta(a_i).$$

Q3

$$\log \pi_\theta(a_k) = \theta_k - \log\left(\sum_{j=1}^{3} e^{\theta_j}\right).$$

$$\frac{\partial}{\partial \theta_i}\theta_k = \mathbf{1}[i=k].$$

$$\frac{\partial}{\partial \theta_i}\log\left(\sum_{j=1}^{3} e^{\theta_j}\right) = \frac{1}{\sum_{j=1}^{3} e^{\theta_j}} \cdot \frac{\partial}{\partial \theta_i}\sum_{j=1}^{3} e^{\theta_j} = \frac{e^{\theta_i}}{\sum_{j=1}^{3} e^{\theta_j}} = \pi_\theta(a_i).$$

$$\boxed{\frac{\partial}{\partial \theta_i}\log \pi_\theta(a_k) = \mathbf{1}[i=k] - \pi_\theta(a_i).}$$

Q3

Consider a one-state bandit MDP with three actions $\{a_1, a_2, a_3\}$ and deterministic rewards $r(a_1) = 4$, $r(a_2) = 1$, $r(a_3) = 0$. The policy is parameterized via softmax:

$$\pi_\theta(a_i) = \frac{e^{\theta_i}}{\sum_{j=1}^{3} e^{\theta_j}}, \qquad \theta = (\theta_1, \theta_2, \theta_3).$$

We run REINFORCE to update our policy. The single-sample gradient after taking action a_k with reward $r(a_k)$ is:

$$\nabla_\theta J = r(a_k) \cdot \nabla_\theta \log \pi_\theta(a_k).$$

(a) Compute $\frac{\partial}{\partial \theta_i} \log \pi_\theta(a_k)$ for the softmax policy above. Express your answer in terms of the policy probabilities $\pi_\theta(a_i)$ and an indicator $\mathbf{1}[i = k]$.

(b) Using your result from part (a), write out the expected policy gradient $\mathbb{E}_{a \sim \pi_\theta}[\nabla_\theta J]$. Be sure to show all work.

Q3

$$\nabla_\theta J = r(a_k) \cdot \nabla_\theta \log \pi_\theta(a_k).$$

(a) Compute $\frac{\partial}{\partial \theta_i} \log \pi_\theta(a_k)$ for the softmax policy above. Express your answer in terms of the policy probabilities $\pi_\theta(a_i)$ and an indicator $\mathbf{1}[i = k]$.

(b) Using your result from part (a), write out the expected policy gradient $\mathbb{E}_{a \sim \pi_\theta}[\nabla_\theta J]$. Be sure to show all work.

$$\mathbb{E}[\nabla_\theta J_i] = \sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k)\, \frac{\partial}{\partial \theta_i} \log \pi_\theta(a_k).$$

Since we are taking expectation over action out of pi

Q3

$$\nabla_\theta J = r(a_k) \cdot \nabla_\theta \log \pi_\theta(a_k).$$

(a) Compute $\frac{\partial}{\partial \theta_i} \log \pi_\theta(a_k)$ for the softmax policy above. Express your answer in terms of the policy probabilities $\pi_\theta(a_i)$ and an indicator $\mathbf{1}[i = k]$.

(b) Using your result from part (a), write out the expected policy gradient $\mathbb{E}_{a \sim \pi_\theta}[\nabla_\theta J]$. Be sure to show all work.

$$\mathbb{E}[\nabla_\theta J_i] = \sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k)\, \frac{\partial}{\partial \theta_i} \log \pi_\theta(a_k).$$

Now Plug in result from a)

$$\mathbb{E}[\nabla_\theta J_i] = \sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k)\, \big(\mathbf{1}[i = k] - \pi_\theta(a_i)\big).$$

Q3

$$\mathbb{E}[\nabla_\theta J_i] = \sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k) \left(\mathbf{1}[i=k] - \pi_\theta(a_i)\right).$$

Distribute the terms

$$\mathbb{E}[\nabla_\theta J_i] = \sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k)\, \mathbf{1}[i=k] \; - \; \sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k)\, \pi_\theta(a_i).$$

Q3

$$\mathbb{E}[\nabla_\theta J_i] = \sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k)\, \big(\mathbf{1}[i=k] - \pi_\theta(a_i)\big).$$

Distribute the terms

$$\mathbb{E}[\nabla_\theta J_i] = \sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k)\, \mathbf{1}[i=k] \; - \; \sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k)\, \pi_\theta(a_i).$$

Noticed that when i=k it is 1 and everything else it is 0 so we can just simplify

$$\sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k)\, \mathbf{1}[i=k] = \pi_\theta(a_i)\, r(a_i).$$

Q3

$$\mathbb{E}[\nabla_\theta J_i] = \sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k)\, \mathbf{1}[i=k] \; - \; \sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k)\, \pi_\theta(a_i).$$

In this second term the pi theta does not depend on k, so we factor

$$\sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k)\, \pi_\theta(a_i) = \pi_\theta(a_i) \sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k).$$

Q3

$$\mathbb{E}[\nabla_\theta J_i] = \sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k)\, \mathbf{1}[i=k] \; - \; \sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k)\, \pi_\theta(a_i).$$

In this second term the pi theta does not depend on k, so we factor

$$\sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k)\, \pi_\theta(a_i) = \pi_\theta(a_i) \sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k).$$

The term below is also the expected reward under the current policy we can write it as $\bar{r}$

$$\sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k)$$

Q3

$$\mathbb{E}[\nabla_\theta J_i] = \sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k)\, \mathbf{1}[i=k] \; - \; \sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k)\, \pi_\theta(a_i).$$

In this second term the pi theta does not depend on k, so we factor

$$\sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k)\, \pi_\theta(a_i) = \pi_\theta(a_i) \sum_{k=1}^{3} \pi_\theta(a_k)\, r(a_k).$$

$$\begin{aligned} \mathbb{E}[\nabla_\theta J_i] &= \pi_\theta(a_i)\, r(a_i) - \pi_\theta(a_i)\, \bar{r} \\ &= \boxed{\pi_\theta(a_i)\Big(r(a_i) - \bar{r}\Big).} \end{aligned}$$

Q3

Consider a one-state bandit MDP with three actions $\{a_1, a_2, a_3\}$ and deterministic rewards $r(a_1) = 4$, $r(a_2) = 1$, $r(a_3) = 0$. The policy is parameterized via softmax:

$$\pi_\theta(a_i) = \frac{e^{\theta_i}}{\sum_{j=1}^{3} e^{\theta_j}}, \qquad \theta = (\theta_1, \theta_2, \theta_3).$$

(c) A student claims: "Since all rewards are non-negative, the REINFORCE update always increases θ_k for the sampled action. Therefore the algorithm can never learn to avoid a suboptimal action." Is this a true statement? Using your expression from part (b), compute $\mathbb{E}[\nabla_\theta J_2]$ (the expected update to the parameter for action a_2) to support your answer.

Start by setting i = 2 for equation from b)

$$\mathbb{E}[\nabla_\theta J_2] = \pi_\theta(a_2)\big(r(a_2) - \bar{r}\big).$$

Q3

Consider a one-state bandit MDP with three actions $\{a_1, a_2, a_3\}$ and deterministic rewards $r(a_1) = 4$, $r(a_2) = 1$, $r(a_3) = 0$. The policy is parameterized via softmax:

$$\pi_\theta(a_i) = \frac{e^{\theta_i}}{\sum_{j=1}^{3} e^{\theta_j}}, \qquad \theta = (\theta_1, \theta_2, \theta_3).$$

(c) A student claims: "Since all rewards are non-negative, the REINFORCE update always increases θ_k for the sampled action. Therefore the algorithm can never learn to avoid a suboptimal action." Is this a true statement? Using your expression from part (b), compute $\mathbb{E}[\nabla_\theta J_2]$ (the expected update to the parameter for action a_2) to support your answer.

Start by setting i = 2 for equation from b)

$$\mathbb{E}[\nabla_\theta J_2] = \pi_\theta(a_2)\big(r(a_2) - \bar{r}\big).$$

This is also 1 from the question

$$\mathbb{E}[\nabla_\theta J_2] = \pi_\theta(a_2)\big(1 - \bar{r}\big).$$

Q3

Consider a one-state bandit MDP with three actions $\{a_1, a_2, a_3\}$ and deterministic rewards $r(a_1) = 4$, $r(a_2) = 1$, $r(a_3) = 0$. The policy is parameterized via softmax:

$$\pi_\theta(a_i) = \frac{e^{\theta_i}}{\sum_{j=1}^{3} e^{\theta_j}}, \qquad \theta = (\theta_1, \theta_2, \theta_3).$$

(c) A student claims: "Since all rewards are non-negative, the REINFORCE update always increases θ_k for the sampled action. Therefore the algorithm can never learn to avoid a suboptimal action." Is this a true statement? Using your expression from part (b), compute $\mathbb{E}[\nabla_\theta J_2]$ (the expected update to the parameter for action a_2) to support your answer.

To make this negative r bar just have to be greater than 1

$$\mathbb{E}[\nabla_\theta J_2] = \pi_\theta(a_2)\big(1 - \bar{r}\big).$$

Q3

Consider a one-state bandit MDP with three actions $\{a_1, a_2, a_3\}$ and deterministic rewards $r(a_1) = 4$, $r(a_2) = 1$, $r(a_3) = 0$. The policy is parameterized via softmax:

$$\pi_\theta(a_i) = \frac{e^{\theta_i}}{\sum_{j=1}^{3} e^{\theta_j}}, \qquad \theta = (\theta_1, \theta_2, \theta_3).$$

(c) A student claims: "Since all rewards are non-negative, the REINFORCE update always increases θ_k for the sampled action. Therefore the algorithm can never learn to avoid a suboptimal action." Is this a true statement? Using your expression from part (b), compute $\mathbb{E}[\nabla_\theta J_2]$ (the expected update to the parameter for action a_2) to support your answer.

To make this negative r bar just have to be greater than 1

$$\mathbb{E}[\nabla_\theta J_2] = \pi_\theta(a_2)\big(1 - \bar{r}\big).$$

$$\bar{r} = 4\,\pi_\theta(a_1) + 1 \cdot \pi_\theta(a_2) + 0 \cdot \pi_\theta(a_3) = 4\,\pi_\theta(a_1) + \pi_\theta(a_2).$$

Q3

Consider a one-state bandit MDP with three actions $\{a_1, a_2, a_3\}$ and deterministic rewards $r(a_1) = 4$, $r(a_2) = 1$, $r(a_3) = 0$. The policy is parameterized via softmax:

$$\pi_\theta(a_i) = \frac{e^{\theta_i}}{\sum_{j=1}^{3} e^{\theta_j}}, \qquad \theta = (\theta_1, \theta_2, \theta_3).$$

(c) A student claims: "Since all rewards are non-negative, the REINFORCE update always increases θ_k for the sampled action. Therefore the algorithm can never learn to avoid a suboptimal action." Is this a true statement? Using your expression from part (b), compute $\mathbb{E}[\nabla_\theta J_2]$ (the expected update to the parameter for action a_2) to support your answer.

Clearly this can easily be greater than 1, assume you have a uniform policy where each pi = ⅓ then it is already 5/3

$$\bar{r} = 4\,\pi_\theta(a_1) + 1 \cdot \pi_\theta(a_2) + 0 \cdot \pi_\theta(a_3) = 4\,\pi_\theta(a_1) + \pi_\theta(a_2).$$

Q3

Consider a one-state bandit MDP with three actions $\{a_1, a_2, a_3\}$ and deterministic rewards $r(a_1) = 4$, $r(a_2) = 1$, $r(a_3) = 0$. The policy is parameterized via softmax:

$$\pi_\theta(a_i) = \frac{e^{\theta_i}}{\sum_{j=1}^{3} e^{\theta_j}}, \qquad \theta = (\theta_1, \theta_2, \theta_3).$$

(c) A student claims: "Since all rewards are non-negative, the REINFORCE update always increases θ_k for the sampled action. Therefore the algorithm can never learn to avoid a suboptimal action." Is this a true statement? Using your expression from part (b), compute $\mathbb{E}[\nabla_\theta J_2]$ (the expected update to the parameter for action a_2) to support your answer.

Clearly this can easily be greater than 1, assume you have a uniform policy where each pi = ⅓ then it is already 5/3

$$\bar{r} = 4\,\pi_\theta(a_1) + 1 \cdot \pi_\theta(a_2) + 0 \cdot \pi_\theta(a_3) = 4\,\pi_\theta(a_1) + \pi_\theta(a_2).$$

Therefore the answer is false it can be negative

Thank You

Good Luck on the Drone race and Final