Policy Gradient Methods: Estimation

Lucas Janson

CS/Stat 184(0): Introduction to Reinforcement Learning

Fall 2024

Today

- Feedback from last lecture
- Recap
- Estimation: REINFORCE
- Variance Reduction
    - Other Gradient Expressions
    - Baselines and Advantages
- Examples

Feedback from feedback forms

Feedback from feedback forms

1. Thank you to everyone who filled out the forms!

Feedback from feedback forms

1. Thank you to everyone who filled out the forms!

Today

- Feedback from last lecture
- Recap
- Estimation: REINFORCE
- Variance Reduction
  - Other Gradient Expressions
  - Baselines and Advantages
- Examples

The Learning Setting:

We don't know the MDP, but we can obtain trajectories.

The Finite Horizon, Learning Setting. We can obtain trajectories as follows:

- We start at $s_0 \sim \mu$.
- We can act for H steps and observe the trajectory $\tau = \{s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1}\}$

Note that with a simulator, we can sample trajectories as specified in the above.

Optimization Objective

- Consider a parameterized class of policies:

$$\{\pi_\theta(a \mid s) \mid \theta \in \mathbb{R}^d\}$$

(why do we make it stochastic?)

- Objective $\max_\theta J(\theta)$, where

$$J(\theta) := \mathbb{E}_{s_0 \sim \mu}\left[V^{\pi_\theta}(s_0)\right] = \mathbb{E}_{\tau \sim \rho_{\pi_\theta}}\left[\sum_{h=0}^{H-1} r(s_h, a_h)\right]$$

- Policy Gradient Descent:

$$\theta^{k+1} = \theta^k + \eta \nabla J(\theta^k)$$

Example Policy Parameterizations

Recall that we consider parameterized policy $\pi_\theta(\cdot \mid s) \in \Delta(A), \forall s$

1. Softmax linear Policy

Feature vector $\phi(s, a, h) \in \mathbb{R}^d$, and parameter $\theta \in \mathbb{R}^d$

$$\pi_\theta(a \mid s, h) = \frac{\exp(\theta^\top \phi(s, a, h))}{\sum_{a'} \exp(\theta^\top \phi(s, a', h))}$$

Example Policy Parameterizations

Recall that we consider parameterized policy $\pi_\theta(\cdot \mid s) \in \Delta(A), \forall s$

1. Softmax linear Policy

Feature vector $\phi(s, a, h) \in \mathbb{R}^d$, and parameter $\theta \in \mathbb{R}^d$

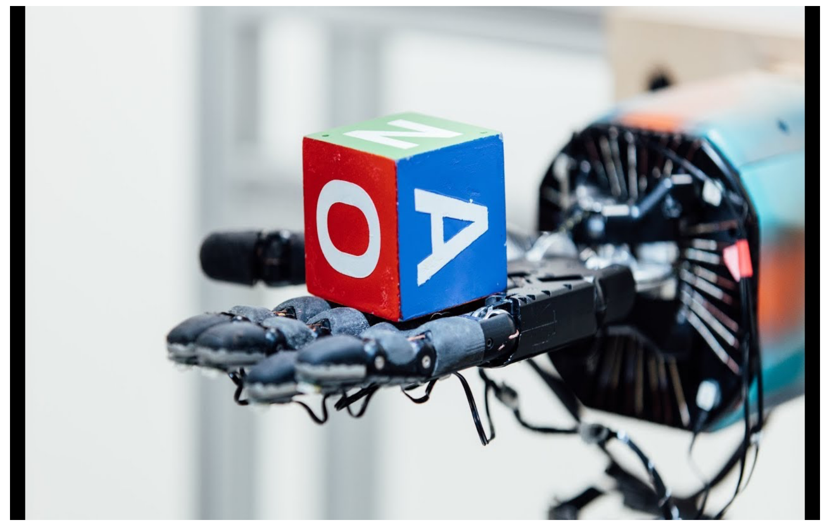
$$\pi_\theta(a \mid s, h) = \frac{\exp(\theta^\top \phi(s, a, h))}{\sum_{a'} \exp(\theta^\top \phi(s, a', h))}$$

2. Neural Policy:

Neural network
$f_\theta : S \times A \times [H] \mapsto \mathbb{R}$

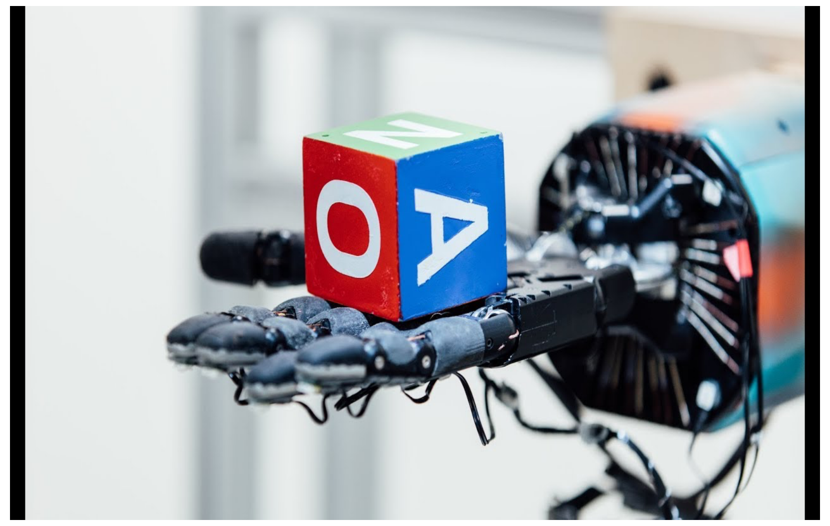
$$\pi_\theta(a \mid s, h) = \frac{\exp(f_\theta(s, a, h))}{\sum_{a'} \exp(f_\theta(s, a', h))}$$

Example Policy Parameterization for “Controls”



Suppose $a \in R^k$, as it might be for a control problem.

Example Policy Parameterization for "Controls"



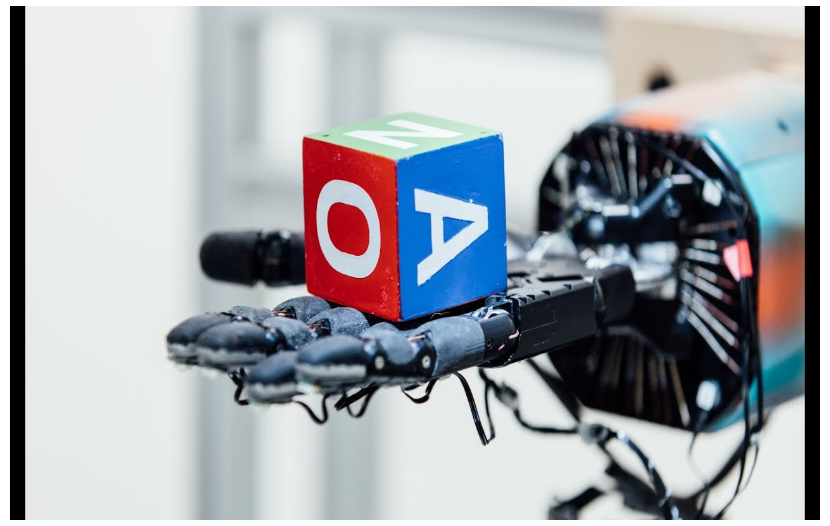
Suppose $a \in R^k$, as it might be for a control problem.

3. Gaussian + Linear Model

- Feature vector: $\phi(s, h) \in \mathbb{R}^d$,
- Parameters: $\theta \in \mathbb{R}^{k \times d}$,
  (and maybe $\sigma \in R^+$)

- Policy: sample action from a (multivariate) Normal with mean $\theta \cdot \phi(s, h)$ and variance $\sigma^2 I$, i.e.
  $$\pi_{\theta,\sigma}(\cdot \mid s, h) = \mathcal{N}\left(\theta \cdot \phi(s, h), \sigma^2 I\right)$$
- Sampling:
  $$a = \theta \cdot \phi(s, h) + \eta, \text{ where } \eta \sim \mathcal{N}(0, \sigma^2 I)$$

Example Policy Parameterization for "Controls"



Suppose $a \in R^k$, as it might be for a control problem.

3. Gaussian + Linear Model

- Feature vector: $\phi(s,h) \in \mathbb{R}^d$,
- Parameters: $\theta \in \mathbb{R}^{k \times d}$,
  (and maybe $\sigma \in R^+$)

- Policy: sample action from a (multivariate) Normal with mean $\theta \cdot \phi(s,h)$ and variance $\sigma^2 I$, i.e.
$$\pi_{\theta,\sigma}(\cdot \,|\, s,h) = \mathcal{N}\left(\theta \cdot \phi(s,h), \sigma^2 I\right)$$
- Sampling:
$$a = \theta \cdot \phi(s,h) + \eta, \text{ where } \eta \sim \mathcal{N}(0,\sigma^2 I)$$

4. Gaussian + Neural Model

- Neural network $g_\theta : S \times [H] \mapsto \mathbb{R}^k$
- Parameters: $\theta \in R^d$,
  (and maybe $\sigma \in R^+$)

- Policy: a (multivariate) Normal with mean $g_\theta(s)$ and variance $\sigma^2 I$, i.e.
$$\pi_{\theta,\sigma}(\cdot \,|\, s,h) = \mathcal{N}(g_\theta(s,h), \sigma^2 I)$$
- Sampling:
$$a = g_\theta(s,h) + \eta, \text{ where } \eta \sim \mathcal{N}(0,\sigma^2 I)$$

The Likelihood Ratio Method

The Likelihood Ratio Method

- Suppose $J(\theta) = \mathbb{E}_{x\sim P_\theta}\left[f(x)\right] = \sum_x P_\theta(x) f(x)$, and our objective is $\max_\theta J(\theta)$.

The Likelihood Ratio Method

- Suppose $J(\theta) = \mathbb{E}_{x \sim P_\theta}\left[f(x)\right] = \sum_x P_\theta(x) f(x)$, and our objective is $\max_\theta J(\theta)$.
- Computing $\nabla_\theta J(\theta)$ exactly may be difficult (due to the sum over x=trajectories)

The Likelihood Ratio Method

- Suppose $J(\theta) = \mathbb{E}_{x \sim P_\theta}\left[f(x)\right] = \sum_x P_\theta(x) f(x)$, and our objective is $\max_\theta J(\theta)$.
- Computing $\nabla_\theta J(\theta)$ exactly may be difficult (due to the sum over x=trajectories)
  - So GD not an option—what about SGD?

The Likelihood Ratio Method

- Suppose $J(\theta) = \mathbb{E}_{x \sim P_\theta}\left[f(x)\right] = \sum_x P_\theta(x) f(x)$, and our objective is $\max_\theta J(\theta)$.
- Computing $\nabla_\theta J(\theta)$ exactly may be difficult (due to the sum over x=trajectories)
  - So GD not an option—what about SGD?
  - In supervised learning, stochastic gradient was just gradient on one sample—will that work here?

The Likelihood Ratio Method

- Suppose $J(\theta) = \mathbb{E}_{x \sim P_\theta}\left[f(x)\right] = \sum_x P_\theta(x) f(x)$, and our objective is $\max_\theta J(\theta)$.
- Computing $\nabla_\theta J(\theta)$ exactly may be difficult (due to the sum over x=trajectories)
  - So GD not an option—what about SGD?
  - In supervised learning, stochastic gradient was just gradient on one sample—will that work here?
  - Won't work: θ-dependence is inside the distribution, not inside the expectation

The Likelihood Ratio Method

- Suppose $J(\theta) = \mathbb{E}_{x \sim P_\theta}\left[f(x)\right] = \sum_x P_\theta(x) f(x)$, and our objective is $\max_\theta J(\theta)$.
- Computing $\nabla_\theta J(\theta)$ exactly may be difficult (due to the sum over x=trajectories)
  - So GD not an option—what about SGD?
  - In supervised learning, stochastic gradient was just gradient on one sample—will that work here?
  - Won't work: θ-dependence is inside the distribution, not inside the expectation
  - So how can we unbiasedly estimate $\nabla_\theta J(\theta)$?

The Likelihood Ratio Method

- Suppose $J(\theta) = \mathbb{E}_{x \sim P_\theta}\left[f(x)\right] = \sum_x P_\theta(x) f(x)$, and our objective is $\max_\theta J(\theta)$.
- Computing $\nabla_\theta J(\theta)$ exactly may be difficult (due to the sum over x=trajectories)
  - So GD not an option—what about SGD?
  - In supervised learning, stochastic gradient was just gradient on one sample—will that work here?
  - Won't work: θ-dependence is inside the distribution, not inside the expectation
  - So how can we unbiasedly estimate $\nabla_\theta J(\theta)$?
- Suppose we can compute $f(x)$, $P_\theta(x)$, and $\nabla P_\theta(x)$, and we can sample $x \sim P_\theta$

The Likelihood Ratio Method

- Suppose $J(\theta) = \mathbb{E}_{x \sim P_\theta}\left[f(x)\right] = \sum_x P_\theta(x) f(x)$, and our objective is $\max_\theta J(\theta)$.
- Computing $\nabla_\theta J(\theta)$ exactly may be difficult (due to the sum over x=trajectories)
  - So GD not an option—what about SGD?
  - In supervised learning, stochastic gradient was just gradient on one sample—will that work here?
  - Won't work: θ-dependence is inside the distribution, not inside the expectation
  - So how can we unbiasedly estimate $\nabla_\theta J(\theta)$?
- Suppose we can compute $f(x)$, $P_\theta(x)$, and $\nabla P_\theta(x)$, and we can sample $x \sim P_\theta$
- We have that:

$$\nabla_\theta J(\theta) = \mathbb{E}_{x \sim P_\theta(x)}\left[\nabla_\theta \log P_\theta(x)\, f(x)\right]$$

The Likelihood Ratio Method

- Suppose $J(\theta) = \mathbb{E}_{x \sim P_\theta}\left[f(x)\right] = \sum_x P_\theta(x) f(x)$, and our objective is $\max_\theta J(\theta)$.
- Computing $\nabla_\theta J(\theta)$ exactly may be difficult (due to the sum over x=trajectories)
  - So GD not an option—what about SGD?
  - In supervised learning, stochastic gradient was just gradient on one sample—will that work here?
  - Won't work: θ-dependence is inside the distribution, not inside the expectation
  - So how can we unbiasedly estimate $\nabla_\theta J(\theta)$?
- Suppose we can compute $f(x)$, $P_\theta(x)$, and $\nabla P_\theta(x)$, and we can sample $x \sim P_\theta$
- We have that:

$$\nabla_\theta J(\theta) = \mathbb{E}_{x \sim P_\theta(x)}\left[\nabla_\theta \log P_\theta(x)\, f(x)\right]$$

Proof:

The Likelihood Ratio Method

- Suppose $J(\theta) = \mathbb{E}_{x \sim P_\theta}\left[f(x)\right] = \sum_x P_\theta(x) f(x)$, and our objective is $\max_\theta J(\theta)$.
- Computing $\nabla_\theta J(\theta)$ exactly may be difficult (due to the sum over x=trajectories)
  - So GD not an option—what about SGD?
  - In supervised learning, stochastic gradient was just gradient on one sample—will that work here?
  - Won't work: θ-dependence is inside the distribution, not inside the expectation
  - So how can we unbiasedly estimate $\nabla_\theta J(\theta)$?
- Suppose we can compute $f(x)$, $P_\theta(x)$, and $\nabla P_\theta(x)$, and we can sample $x \sim P_\theta$
- We have that:

$$\nabla_\theta J(\theta) = \mathbb{E}_{x \sim P_\theta(x)}\left[\nabla_\theta \log P_\theta(x)\, f(x)\right]$$

Proof:

$$\nabla_\theta J(\theta) = \sum_x \nabla_\theta P_\theta(x) f(x)$$

The Likelihood Ratio Method

- Suppose $J(\theta) = \mathbb{E}_{x\sim P_\theta}\left[f(x)\right] = \sum_x P_\theta(x)f(x)$, and our objective is $\max_\theta J(\theta)$.
- Computing $\nabla_\theta J(\theta)$ exactly may be difficult (due to the sum over x=trajectories)
  - So GD not an option—what about SGD?
  - In supervised learning, stochastic gradient was just gradient on one sample—will that work here?
  - Won't work: θ-dependence is inside the distribution, not inside the expectation
  - So how can we unbiasedly estimate $\nabla_\theta J(\theta)$?
- Suppose we can compute $f(x)$, $P_\theta(x)$, and $\nabla P_\theta(x)$, and we can sample $x \sim P_\theta$
- We have that:

$$\nabla_\theta J(\theta) = \mathbb{E}_{x\sim P_\theta(x)}\left[\nabla_\theta \log P_\theta(x)\, f(x)\right]$$

Proof:

$$\begin{aligned}\nabla_\theta J(\theta) &= \sum_x \nabla_\theta P_\theta(x) f(x)\\ &= \sum_x P_\theta(x)\frac{\nabla_\theta P_\theta(x)}{P_\theta(x)} f(x)\end{aligned}$$

The Likelihood Ratio Method

- Suppose $J(\theta) = \mathbb{E}_{x \sim P_\theta}\left[f(x)\right] = \sum_x P_\theta(x) f(x)$, and our objective is $\max_\theta J(\theta)$.
- Computing $\nabla_\theta J(\theta)$ exactly may be difficult (due to the sum over x=trajectories)
  - So GD not an option—what about SGD?
  - In supervised learning, stochastic gradient was just gradient on one sample—will that work here?
  - Won't work: θ-dependence is inside the distribution, not inside the expectation
  - So how can we unbiasedly estimate $\nabla_\theta J(\theta)$?
- Suppose we can compute $f(x)$, $P_\theta(x)$, and $\nabla P_\theta(x)$, and we can sample $x \sim P_\theta$
- We have that:

$$\nabla_\theta J(\theta) = \mathbb{E}_{x \sim P_\theta(x)}\left[\nabla_\theta \log P_\theta(x)\, f(x)\right]$$

Proof:

$$\begin{aligned}\nabla_\theta J(\theta) &= \sum_x \nabla_\theta P_\theta(x) f(x) \\ &= \sum_x P_\theta(x) \frac{\nabla_\theta P_\theta(x)}{P_\theta(x)} f(x) \\ &= \sum_x P_\theta(x)\, \nabla_\theta \log P_\theta(x) f(x)\end{aligned}$$

The Likelihood Ratio Method, continued

The Likelihood Ratio Method, continued

- We have:

$$\nabla_\theta J(\theta) = \mathbb{E}_{x \sim P_\theta(x)} \left[\nabla_\theta \log P_\theta(x)\, f(x) \right]$$

The Likelihood Ratio Method, continued

- We have:

$$\nabla_\theta J(\theta) = \mathbb{E}_{x \sim P_\theta(x)} \left[\nabla_\theta \log P_\theta(x) \, f(x) \right]$$

- An unbiased estimate is given by:

$\widehat{\nabla}_\theta J(\theta) = \nabla_\theta \log P_\theta(x) \cdot f(x)$, where $x \sim P_\theta$

The Likelihood Ratio Method, continued

- We have:

$$\nabla_\theta J(\theta) = \mathbb{E}_{x \sim P_\theta(x)} \left[\nabla_\theta \log P_\theta(x)\, f(x) \right]$$

- An unbiased estimate is given by:

$\widehat{\nabla}_\theta J(\theta) = \nabla_\theta \log P_\theta(x) \cdot f(x)$, where $x \sim P_\theta$

- We can lower variance by drawing N i.i.d. samples from P_θ and averaging:

$$\widehat{\nabla}_\theta J(\theta) = \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \log P_\theta(x_i)\, f(x_i)$$

Today

- Feedback from last lecture
- Recap
- Estimation: REINFORCE
- Variance Reduction
  - Other Gradient Expressions
  - Baselines and Advantages
- Examples

Apply likelihood ratio method to policy gradient

Apply likelihood ratio method to policy gradient

- Let $\rho_\theta(\tau)$ be the probability of a trajectory $\tau = \{s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1}\}$, i.e.

$$\rho_\theta(\tau) = \mu(s_0)\pi_\theta(a_0 \,|\, s_0)P(s_1 \,|\, s_0, a_0)\ldots P(s_{H-1} \,|\, s_{H-2}, a_{H-2})\pi_\theta(a_{H-1} \,|\, s_{H-1})$$

Apply likelihood ratio method to policy gradient

- Let $\rho_\theta(\tau)$ be the probability of a trajectory $\tau = \{s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1}\}$, i.e.

$$\rho_\theta(\tau) = \mu(s_0)\pi_\theta(a_0 \mid s_0)P(s_1 \mid s_0, a_0)\ldots P(s_{H-1} \mid s_{H-2}, a_{H-2})\pi_\theta(a_{H-1} \mid s_{H-1})$$

- Let $R(\tau)$ be the cumulative reward on trajectory τ, i.e. $R(\tau) := \sum_{h=0}^{H-1} r(s_h, a_h)$

Apply likelihood ratio method to policy gradient

- Let $\rho_\theta(\tau)$ be the probability of a trajectory $\tau = \{s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1}\}$, i.e.

$$\rho_\theta(\tau) = \mu(s_0)\pi_\theta(a_0 \mid s_0)P(s_1 \mid s_0, a_0)\ldots P(s_{H-1} \mid s_{H-2}, a_{H-2})\pi_\theta(a_{H-1} \mid s_{H-1})$$

- Let $R(\tau)$ be the cumulative reward on trajectory τ, i.e. $R(\tau) := \sum_{h=0}^{H-1} r(s_h, a_h)$

- Our objective function is:

$$J(\theta) \ = \mathbb{E}_{\tau \sim \rho_\theta}\Big[R(\tau)\Big]$$

Apply likelihood ratio method to policy gradient

- Let $\rho_\theta(\tau)$ be the probability of a trajectory $\tau = \{s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1}\}$, i.e.

$$\rho_\theta(\tau) = \mu(s_0)\pi_\theta(a_0 \mid s_0)P(s_1 \mid s_0, a_0)\ldots P(s_{H-1} \mid s_{H-2}, a_{H-2})\pi_\theta(a_{H-1} \mid s_{H-1})$$

- Let $R(\tau)$ be the cumulative reward on trajectory τ, i.e. $R(\tau) := \sum_{h=0}^{H-1} r(s_h, a_h)$

- Our objective function is:

$$J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta}\left[R(\tau)\right]$$

- From the likelihood ratio method, we have:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta}\left[\nabla_\theta \ln \rho_\theta(\tau)\ R(\tau)\right]$$

Apply likelihood ratio method to policy gradient

- Let $\rho_\theta(\tau)$ be the probability of a trajectory $\tau = \{s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1}\}$, i.e.

$$\rho_\theta(\tau) = \mu(s_0)\pi_\theta(a_0 \mid s_0)P(s_1 \mid s_0, a_0)\ldots P(s_{H-1} \mid s_{H-2}, a_{H-2})\pi_\theta(a_{H-1} \mid s_{H-1})$$

- Let $R(\tau)$ be the cumulative reward on trajectory τ, i.e. $R(\tau) := \sum_{h=0}^{H-1} r(s_h, a_h)$

- Our objective function is:

$$J(\theta) = \mathbb{E}_{\tau\sim\rho_\theta}\left[R(\tau)\right]$$

- From the likelihood ratio method, we have:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\rho_\theta}\left[\nabla_\theta \ln \rho_\theta(\tau)\ R(\tau)\right]$$

- But $\rho_\theta(\tau)$ involves the dynamics P, which we assumed we don't know!

REINFORCE: A Policy Gradient Algorithm

REINFORCE: A Policy Gradient Algorithm

- The REINFORCE Policy Gradient expression:

$$\nabla_\theta \ln \rho_\theta(\tau)\ R(\tau) = \left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) \right) R(\tau)$$

REINFORCE: A Policy Gradient Algorithm

- The REINFORCE Policy Gradient expression:

$$\nabla_\theta \ln \rho_\theta(\tau)\ R(\tau) = \left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) \right) R(\tau)$$

- Proof:

$$\nabla_\theta \ln \rho_\theta(\tau) = \nabla_\theta \big(\ln \mu(s_0) + \ln \pi_\theta(a_0 \mid s_0) + \ln P(s_1 \mid s_0, a_0) + \ldots \big)$$

REINFORCE: A Policy Gradient Algorithm

- The REINFORCE Policy Gradient expression:

$$\nabla_\theta \ln \rho_\theta(\tau)\ R(\tau) = \left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) \right) R(\tau)$$

- Proof:

$$\nabla_\theta \ln \rho_\theta(\tau) = \nabla_\theta \Big(\ln \mu(s_0) + \ln \pi_\theta(a_0 \mid s_0) + \ln P(s_1 \mid s_0, a_0) + \ldots \Big)$$

$$= \nabla_\theta \Big(\ln \pi_\theta(a_0 \mid s_0) + \ln \pi_\theta(a_1 \mid s_1) \ldots \Big)$$

REINFORCE: A Policy Gradient Algorithm

- The REINFORCE Policy Gradient expression:

$$\nabla_\theta \ln \rho_\theta(\tau)\ R(\tau) = \left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) \right) R(\tau)$$

- Proof:

$$\nabla_\theta \ln \rho_\theta(\tau) = \nabla_\theta \Big(\ln \mu(s_0) + \ln \pi_\theta(a_0 \mid s_0) + \ln P(s_1 \mid s_0, a_0) + \ldots \Big)$$

$$= \nabla_\theta \Big(\ln \pi_\theta(a_0 \mid s_0) + \ln \pi_\theta(a_1 \mid s_1) \ldots \Big)$$

$$= \left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) \right)$$

Obtaining an Unbiased Gradient Estimate at θ

$$\nabla_\theta J(\theta) := \mathbb{E}_{\tau \sim \rho_\theta}\left[\left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\right) R(\tau)\right]$$

Obtaining an Unbiased Gradient Estimate at θ

$$\nabla_\theta J(\theta) := \mathbb{E}_{\tau \sim \rho_\theta}\left[\left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\right) R(\tau)\right]$$

1. Obtain a trajectory $\tau \sim \rho_\theta$
   (which we can do in our learning setting)

Obtaining an Unbiased Gradient Estimate at θ

$$\nabla_\theta J(\theta) := \mathbb{E}_{\tau \sim \rho_\theta}\left[\left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\right) R(\tau)\right]$$

1. Obtain a trajectory $\tau \sim \rho_\theta$
   (which we can do in our learning setting)
2. Set:

$$g(\theta, \tau) := \left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\right) R(\tau)$$

Obtaining an Unbiased Gradient Estimate at θ

$$\nabla_\theta J(\theta) := \mathbb{E}_{\tau \sim \rho_\theta}\left[\left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\right) R(\tau)\right]$$

1. Obtain a trajectory $\tau \sim \rho_\theta$
   (which we can do in our learning setting)
2. Set:

$$g(\theta, \tau) := \left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\right) R(\tau)$$

We have: $\mathbb{E}[g(\theta, \tau)] = \nabla_\theta J(\theta)$

PG with REINFORCE:

PG with REINFORCE:

1. Initialize θ^0, step size parameters: $\eta^1, \eta^2, \ldots$

PG with REINFORCE:

1. Initialize θ^0, step size parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:

PG with REINFORCE:

1. Initialize θ^0, step size parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
    1. Obtain a trajectory $\tau \sim \rho_{\theta^k}$

       Compute $g(\theta^k, \tau)$

PG with REINFORCE:

1. Initialize θ^0, step size parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
    1. Obtain a trajectory $\tau \sim \rho_{\theta^k}$
       Compute $g(\theta^k, \tau)$
    2. Update: $\theta^{k+1} = \theta^k + \eta^k g(\theta^k, \tau)$

The (mini-batch) PG procedure with REINFORCE

(reducing variance using batch sizes of M)

The (mini-batch) PG procedure with REINFORCE

(reducing variance using batch sizes of M)

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$

The (mini-batch) PG procedure with REINFORCE

(reducing variance using batch sizes of M)

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:

The (mini-batch) PG procedure with REINFORCE

(reducing variance using batch sizes of M)

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
    1. Init $G = 0$ and do M times:

       Obtain a trajectory $\tau \sim \rho_{\theta^k}$

       Update: $G \leftarrow G + g(\theta^k, \tau)$

The (mini-batch) PG procedure with REINFORCE

(reducing variance using batch sizes of M)

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
   1. Init $G = 0$ and do M times:
      Obtain a trajectory $\tau \sim \rho_{\theta^k}$
      Update: $G \leftarrow G + g(\theta^k, \tau)$
   2. Set $g := \frac{1}{M} G$

The (mini-batch) PG procedure with REINFORCE

(reducing variance using batch sizes of M)

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
   1. Init $G = 0$ and do M times:

      Obtain a trajectory $\tau \sim \rho_{\theta^k}$

      Update: $G \leftarrow G + g(\theta^k, \tau)$
   2. Set $g := \dfrac{1}{M} G$
   3. Update: $\theta^{k+1} = \theta^k + \eta^k g$

The (mini-batch) PG procedure with REINFORCE

(reducing variance using batch sizes of M)

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
   1. Init $G = 0$ and do M times:
      Obtain a trajectory $\tau \sim \rho_{\theta^k}$
      Update: $G \leftarrow G + g(\theta^k, \tau)$
   2. Set $g := \frac{1}{M} G$
   3. Update: $\theta^{k+1} = \theta^k + \eta^k g$

We still have that at the kth step, g is unbiased for $\nabla_\theta J(\theta)$ evaluated at θ^k

Today

- Feedback from last lecture
- Recap
- Estimation: REINFORCE
- Variance Reduction
  - Other Gradient Expressions
  - Baselines and Advantages
- Examples

Other PG formulas
(that are lower variance for sampling)

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta}\left[\left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\right) R(\tau)\right] \qquad \text{(REINFORCE)}$$

Other PG formulas (that are lower variance for sampling)

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta}\left[\left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\right) R(\tau)\right]$$

(REINFORCE)

$$= \mathbb{E}_{\tau \sim \rho_\theta}\left[\sum_{h=0}^{H-1} \left(\nabla_\theta \ln \pi_\theta(a_h \mid s_h) \sum_{t=h}^{H-1} r_t \right)\right]$$

Other PG formulas (that are lower variance for sampling)

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta}\left[\left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\right) R(\tau)\right]$$

(REINFORCE)

$$= \mathbb{E}_{\tau \sim \rho_\theta}\left[\sum_{h=0}^{H-1}\left(\nabla_\theta \ln \pi_\theta(a_h \mid s_h) \sum_{t=h}^{H-1} r_t\right)\right]$$

$$= \mathbb{E}_{\tau \sim \rho_\theta}\left[\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) Q_h^{\pi_\theta}(s_h, a_h)\right]$$

Other PG formulas
(that are lower variance for sampling)

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta}\left[\left(\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\right) R(\tau)\right] \qquad \text{(REINFORCE)}$$

$$= \mathbb{E}_{\tau \sim \rho_\theta}\left[\sum_{h=0}^{H-1}\left(\nabla_\theta \ln \pi_\theta(a_h \mid s_h) \sum_{t=h}^{H-1} r_t\right)\right]$$

$$= \mathbb{E}_{\tau \sim \rho_\theta}\left[\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) Q_h^{\pi_\theta}(s_h, a_h)\right]$$

Intuition: Changing the action distribution at h only affects rewards later on…

HW: You will show these simplified version are also valid PG expressions

An improved policy gradient procedure:

An improved policy gradient procedure:

On a trajectory τ, define:

$$R_h(\tau) = \sum_{t=h}^{H-1} r_t$$

And define:

$$g'(\theta, \tau) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) R_h(\tau)$$

An improved policy gradient procedure:

On a trajectory τ, define:

$$R_h(\tau) = \sum_{t=h}^{H-1} r_t$$

And define:

$$g'(\theta, \tau) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) R_h(\tau)$$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$

An improved policy gradient procedure:

On a trajectory τ, define:

$$R_h(\tau) = \sum_{t=h}^{H-1} r_t$$

And define:

$$g'(\theta, \tau) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h | s_h) R_h(\tau)$$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:

An improved policy gradient procedure:

On a trajectory τ, define:

$$R_h(\tau) = \sum_{t=h}^{H-1} r_t$$

And define:

$$g'(\theta, \tau) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) R_h(\tau)$$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
    1. Obtain a trajectory $\tau \sim \rho_{\theta^k}$
       Set $g'(\theta^k, \tau)$

An improved policy gradient procedure:

On a trajectory τ, define:

$$R_h(\tau) = \sum_{t=h}^{H-1} r_t$$

And define:

$$g'(\theta, \tau) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h | s_h) R_h(\tau)$$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
   1. Obtain a trajectory $\tau \sim \rho_{\theta^k}$
      Set $g'(\theta^k, \tau)$
   2. Update: $\theta^{k+1} = \theta^k + \eta^k g'(\theta^k, \tau)$

An improved policy gradient procedure:

On a trajectory τ, define:

$$R_h(\tau) = \sum_{t=h}^{H-1} r_t$$

And define:

$$g'(\theta, \tau) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) R_h(\tau)$$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
   1. Obtain a trajectory $\tau \sim \rho_{\theta^k}$

      Set $g'(\theta^k, \tau)$
   2. Update: $\theta^{k+1} = \theta^k + \eta^k g'(\theta^k, \tau)$

Comments:

- We still have unbiased gradient estimates.
- Easy to use a mini-batch algorithm to reduce variance.
- Easy to compute the gradient in “one pass” over the data.

Today

- Feedback from last lecture
- Recap
- Estimation: REINFORCE
- Variance Reduction
  - Other Gradient Expressions
  - Baselines and Advantages
- Examples

With a “baseline” function:

With a "baseline" function:

For any function only of the state, $b_h : S \to \mathbb{R}$, we have:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta}\left[\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\big(R_h(\tau) - b_h(s_h)\big)\right]$$

$$= \mathbb{E}_{\tau \sim \rho_\theta}\left[\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\Big(Q_h^{\pi_\theta}(s_h, a_h) - b_h(s_h)\Big)\right]$$

With a “baseline” function:

For any function only of the state, $b_h : S \to \mathbb{R}$, we have:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta}\left[\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \,|\, s_h)\big(R_h(\tau) - b_h(s_h)\big)\right]$$

$$= \mathbb{E}_{\tau \sim \rho_\theta}\left[\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \,|\, s_h)\Big(Q_h^{\pi_\theta}(s_h, a_h) - b_h(s_h)\Big)\right]$$

This is (basically) the method of control variates.

Proof:

Proof:

- To see this, first note:
  $$\mathbb{E}_{x \sim P_\theta}\left[\nabla_\theta \log P_\theta(x)\ c\right] \overset{\text{LRM}}{=} \nabla_\theta \mathbb{E}_{x\sim P_\theta}[c] = \nabla_\theta c = 0$$

Proof:

- To see this, first note:
  $\mathbb{E}_{x \sim P_\theta}\left[\nabla_\theta \log P_\theta(x)\ c\right] = 0$

- Thus for any constant c,
  $\mathbb{E}_{x \sim P_\theta}\left[\nabla_\theta \log P_\theta(x)\left(f(x) - c\right)\right] = \mathbb{E}_{x \sim P_\theta}\left[\nabla_\theta \log P_\theta(x) f(x)\right]$

Proof:

- To see this, first note:
  $$\mathbb{E}_{x\sim P_\theta}\left[\nabla_\theta \log P_\theta(x)\ c\right] =$$

- Thus for any constant c,
  $$\mathbb{E}_{x\sim P_\theta}\left[\nabla_\theta \log P_\theta(x)\big(f(x)-c\big)\right] = \mathbb{E}_{x\sim P_\theta}\left[\nabla_\theta \log P_\theta(x) f(x)\right]$$

- Returning to RL, we have:
  $$\mathbb{E}_{\tau\sim\rho_\theta}\left[\sum_{h=0}^{H-1}\nabla_\theta \ln \pi_\theta(a_h|s_h)\big(R_h(\tau)-b_h(s_h)\big)\right] = \sum_{h=0}^{H-1}\mathbb{E}_{s_h\sim\rho_\theta}\left[\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[\nabla_\theta \ln \pi_\theta(a_h|s_h)\big(R_h(\tau)-b_h(s_h)\big)\right]\right]$$
  $$= \sum_{h=0}^{H-1}\mathbb{E}_{s_h\sim\rho_\theta}\left[\mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h)}\left[\nabla_\theta \ln \pi_\theta(a_h|s_h) R_h(\tau)\right]\right]$$

(where $s_h \sim \rho_\theta$ is a sample from the marginal state distribution at time h)

PG with a Naive (constant) Baseline:

PG with a Naive (constant) Baseline:

- Lets try to use a constant (time-dependent) baseline: $b_h^{\theta} = \mathbb{E}_{\tau \sim \rho_{\theta}(\tau)} \left[R_h(\tau) \right]$

PG with a Naive (constant) Baseline:

- Lets try to use a constant (time-dependent) baseline: $b_h^{\theta} = \mathbb{E}_{\tau \sim \rho_{\theta}(\tau)} \left[R_h(\tau) \right]$

$$g'(\theta, \tau, b) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) \left(R_h(\tau) - b_h \right)$$

PG with a Naive (constant) Baseline:

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$

- Lets try to use a constant (time-dependent) baseline: $b_h^{\theta} = \mathbb{E}_{\tau \sim \rho_{\theta}(\tau)} \left[R_h(\tau) \right]$

$$g'(\theta, \tau, b) := \sum_{h=0}^{H-1} \nabla_{\theta} \ln \pi_{\theta}(a_h \mid s_h) \left(R_h(\tau) - b_h \right)$$

PG with a Naive (constant) Baseline:

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:

- Lets try to use a constant (time-dependent) baseline: $b_h^{\theta} = \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \left[R_h(\tau) \right]$

$$g'(\theta, \tau, b) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h | s_h) \left(R_h(\tau) - b_h \right)$$

PG with a Naive (constant) Baseline:

- Lets try to use a constant (time-dependent) baseline: $b_h^{\theta} = \mathbb{E}_{\tau \sim \rho_{\theta}(\tau)} \left[R_h(\tau) \right]$

$$g'(\theta, \tau, b) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) \left(R_h(\tau) - b_h \right)$$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
    1. Sample M trajectories, $\tau_1, \ldots, \tau_M \sim \rho_{\theta^k}$. Set:

       $\widetilde{b} = (\tilde{b}_0, \ldots, \tilde{b}_{H-1})$, where $\widetilde{b}_h = \frac{1}{M} \sum_{i=1}^{M} R_h(\tau_i)$

PG with a Naive (constant) Baseline:

- Lets try to use a constant (time-dependent) baseline: $b_h^{\theta} = \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \left[R_h(\tau) \right]$

$$g'(\theta, \tau, b) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) \left(R_h(\tau) - b_h \right)$$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
   1. Sample M trajectories, $\tau_1, \ldots, \tau_M \sim \rho_{\theta^k}$. Set:
   $\widetilde{b} = (\tilde{b}_0, \ldots, \tilde{b}_{H-1})$, where $\widetilde{b}_h = \frac{1}{M} \sum_{i=1}^{M} R_h(\tau_i)$
   2. Obtain a trajectory $\tau \sim \rho_{\theta^k}$
   Compute $g'(\theta^k, \tau, \widetilde{b})$

PG with a Naive (constant) Baseline:

- Lets try to use a constant (time-dependent) baseline: $b_h^{\theta} = \mathbb{E}_{\tau \sim \rho_\theta(\tau)}\left[R_h(\tau)\right]$

$$g'(\theta, \tau, b) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\left(R_h(\tau) - b_h\right)$$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
   1. Sample M trajectories, $\tau_1, \ldots, \tau_M \sim \rho_{\theta^k}$. Set:
      $\widetilde{b} = (\tilde{b}_0, \ldots, \tilde{b}_{H-1})$, where $\widetilde{b}_h = \frac{1}{M}\sum_{i=1}^{M} R_h(\tau_i)$
   2. Obtain a trajectory $\tau \sim \rho_{\theta^k}$
      Compute $g'(\theta^k, \tau, \widetilde{b})$
   3. Update: $\theta^{k+1} = \theta^k + \eta^k g'(\theta^k, \tau, \widetilde{b})$

The Advantage Function (finite horizon)

$$V_h^\pi(s) = \mathbb{E}\left[\sum_{t=h}^{H-1} r(s_t, a_t) \middle| s_h = s\right] \qquad Q_h^\pi(s, a) = \mathbb{E}\left[\sum_{t=h}^{H-1} r(s_t, a_t) \middle| (s_h, a_h) = (s, a)\right]$$

The Advantage Function (finite horizon)

$$V_h^\pi(s) = \mathbb{E}\left[\sum_{t=h}^{H-1} r(s_t, a_t) \middle| s_h = s\right] \qquad Q_h^\pi(s, a) = \mathbb{E}\left[\sum_{t=h}^{H-1} r(s_t, a_t) \middle| (s_h, a_h) = (s, a)\right]$$

- The Advantage function is defined as:

$$A_h^\pi(s, a) = Q_h^\pi(s, a) - V_h^\pi(s)$$

The Advantage Function (finite horizon)

$$V_h^\pi(s) = \mathbb{E}\left[\sum_{t=h}^{H-1} r(s_t, a_t) \middle| s_h = s\right] \qquad Q_h^\pi(s,a) = \mathbb{E}\left[\sum_{t=h}^{H-1} r(s_t, a_t) \middle| (s_h, a_h) = (s,a)\right]$$

- The Advantage function is defined as:

$$A_h^\pi(s,a) = Q_h^\pi(s,a) - V_h^\pi(s)$$

- We have that:

$$\mathbb{E}_{a\sim\pi(\cdot|s)}\left[A_h^\pi(s,a) \middle| s, h\right] = \sum_a \pi(a\,|\,s) A_h^\pi(s,a) = \;??$$

The Advantage Function (finite horizon)

$$V_h^\pi(s) = \mathbb{E}\left[\sum_{t=h}^{H-1} r(s_t, a_t) \,\middle|\, s_h = s\right] \qquad Q_h^\pi(s,a) = \mathbb{E}\left[\sum_{t=h}^{H-1} r(s_t, a_t) \,\middle|\, (s_h, a_h) = (s,a)\right]$$

- The Advantage function is defined as:

$$A_h^\pi(s,a) = Q_h^\pi(s,a) - V_h^\pi(s)$$

- We have that:

$$\mathbb{E}_{a\sim\pi(\cdot|s)}\left[A_h^\pi(s,a) \,\middle|\, s, h\right] = \sum_a \pi(a\,|\,s) A_h^\pi(s,a) = \ ??$$

- What do we know about $A_h^{\pi^\star}(s,a)$? $\leq 0 \quad \forall a, s$

The Advantage Function (finite horizon)

$$V_h^\pi(s) = \mathbb{E}\left[\sum_{t=h}^{H-1} r(s_t, a_t) \middle| s_h = s\right] \qquad Q_h^\pi(s,a) = \mathbb{E}\left[\sum_{t=h}^{H-1} r(s_t, a_t) \middle| (s_h, a_h) = (s,a)\right]$$

- The Advantage function is defined as:

$$A_h^\pi(s,a) = Q_h^\pi(s,a) - V_h^\pi(s)$$

- We have that:

$$\mathbb{E}_{a\sim\pi(\cdot|s)}\left[A_h^\pi(s,a) \middle| s, h\right] = \sum_a \pi(a\,|\,s) A_h^\pi(s,a) = \ ??$$

- What do we know about $A_h^{\pi^\star}(s,a)$?

- For the discounted case, $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$

The Advantage-based PG:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \left[\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) \left(Q_h^{\pi_\theta}(s_h, a_h) - b_h(s_h) \right) \right]$$

The Advantage-based PG:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \left[\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) \left(Q_h^{\pi_\theta}(s_h, a_h) - b_h(s_h) \right) \right]$$

$$= \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \left[\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) A_h^{\pi_\theta}(s_h, a_h) \right]$$

The Advantage-based PG:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \left[\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) \left(Q_h^{\pi_\theta}(s_h, a_h) - b_h(s_h) \right) \right]$$

$$= \mathbb{E}_{\tau \sim \rho_\theta(\tau)} \left[\sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h) A_h^{\pi_\theta}(s_h, a_h) \right]$$

- The second step follows by choosing $b_h(s) = V_h^\pi(s)$.

The Advantage-based PG:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\rho_\theta(\tau)}\left[\sum_{h=0}^{H-1}\nabla_\theta \ln \pi_\theta(a_h\,|\,s_h)\Big(Q_h^{\pi_\theta}(s_h,a_h) - b_h(s_h)\Big)\right]$$

$$= \mathbb{E}_{\tau\sim\rho_\theta(\tau)}\left[\sum_{h=0}^{H-1}\nabla_\theta \ln \pi_\theta(a_h\,|\,s_h) A_h^{\pi_\theta}(s_h,a_h)\right]$$

- The second step follows by choosing $b_h(s) = V_h^\pi(s)$.
- In practice, the most common approach is to use $b_h(s)$ that's an estimate of $V_h^\pi(s)$.

PG with a Learned Baseline:

PG with a Learned Baseline:

Let $g'(\theta, \tau, b()) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\big(R_h(\tau) - b(s_h, h)\big)$

PG with a Learned Baseline:

Let $g'(\theta, \tau, b()) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\big(R_h(\tau) - b(s_h, h)\big)$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$

PG with a Learned Baseline:

Let $g'(\theta, \tau, b()) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\left(R_h(\tau) - b(s_h, h)\right)$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:

PG with a Learned Baseline:

$$\text{Let } g'(\theta, \tau, b()) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h | s_h)\big(R_h(\tau) - b(s_h, h)\big)$$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0,\ldots$:
    1. Supervised Learning: Using N trajectories sampled under π_{θ^k}, estimate a baseline $\widetilde{b}$
    $\widetilde{b}(s, h) \approx V_h^{\theta^k}(s)$

PG with a Learned Baseline:

$$\text{Let } g'(\theta, \tau, b()) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h \mid s_h)\big(R_h(\tau) - b(s_h, h)\big)$$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
    1. Supervised Learning: Using N trajectories sampled under π_{θ^k}, estimate a baseline $\widetilde{b}$
       $\widetilde{b}(s, h) \approx V_h^{\theta^k}(s)$
    2. Obtain a trajectory $\tau \sim \rho_{\theta^k}$
       Compute $g'(\theta^k, \tau, \widetilde{b}())$

PG with a Learned Baseline:

$$\text{Let } g'(\theta, \tau, b()) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h | s_h)\big(R_h(\tau) - b(s_h, h)\big)$$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
   1. Supervised Learning: Using N trajectories sampled under π_{θ^k}, estimate a baseline $\widetilde{b}$
      $\widetilde{b}(s, h) \approx V_h^{\theta^k}(s)$
   2. Obtain a trajectory $\tau \sim \rho_{\theta^k}$
      Compute $g'(\theta^k, \tau, \widetilde{b}())$
   3. Update: $\theta^{k+1} = \theta^k + \eta^k g'(\theta^k, \tau, \widetilde{b}())$

PG with a Learned Baseline:

$$\text{Let } g'(\theta, \tau, b()) := \sum_{h=0}^{H-1} \nabla_\theta \ln \pi_\theta(a_h | s_h)\big(R_h(\tau) - b(s_h, h)\big)$$

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
   1. Supervised Learning: Using N trajectories sampled under π_{θ^k}, estimate a baseline $\widetilde{b}$
      $\widetilde{b}(s, h) \approx V_h^{\theta^k}(s)$
   2. Obtain a trajectory $\tau \sim \rho_{\theta^k}$
      Compute $g'(\theta^k, \tau, \widetilde{b}())$
   3. Update: $\theta^{k+1} = \theta^k + \eta^k g'(\theta^k, \tau, \widetilde{b}())$

Note that regardless of our choice of $\widetilde{b}$, we still get unbiased gradient estimates.

(minibatch) PG with a Learned Baseline:

(minibatch) PG with a Learned Baseline:

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$

(minibatch) PG with a Learned Baseline:

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:

(minibatch) PG with a Learned Baseline:

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
    1. Supervised Learning: Using N trajectories sampled under π_{θ^k}, estimate a baseline $\widetilde{b}$
       $\widetilde{b}(s,h) \approx V_h^{\theta^k}(s)$

(minibatch) PG with a Learned Baseline:

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
   1. Supervised Learning: Using N trajectories sampled under π_{θ^k}, estimate a baseline $\widetilde{b}$
      $\widetilde{b}(s, h) \approx V_h^{\theta^k}(s)$
   2. Obtain M trajectories $\tau_1, \ldots \tau_M \sim \rho_{\theta^k}$

      Compute $g = \frac{1}{M} \sum_{m=1}^{M} g'(\theta^k, \tau_m, \widetilde{b}())$

(minibatch) PG with a Learned Baseline:

1. Initialize θ^0, parameters: $\eta^1, \eta^2, \ldots$
2. For $k = 0, \ldots$:
    1. Supervised Learning: Using N trajectories sampled under π_{θ^k}, estimate a baseline $\widetilde{b}$
       $\widetilde{b}(s, h) \approx V_h^{\theta^k}(s)$
    2. Obtain M trajectories $\tau_1, \ldots \tau_M \sim \rho_{\theta^k}$

       Compute $g = \frac{1}{M} \sum_{m=1}^{M} g'(\theta^k, \tau_m, \widetilde{b}())$
    3. Update: $\theta^{k+1} = \theta^k + \eta^k g$

Today

- Feedback from last lecture
- Recap
- Estimation: REINFORCE
- Variance Reduction
    - Other Gradient Expressions
    - Baselines and Advantages
- Examples

Policy Parameterizations

Recall that we consider parameterized policy $\pi_\theta(\cdot \mid s) \in \Delta(A), \forall s$

1. Softmax linear Policy

Feature vector $\phi(s,a) \in \mathbb{R}^d$, and parameter $\theta \in \mathbb{R}^d$

$$\pi_\theta(a \mid s) = \frac{\exp(\theta^\top \phi(s,a))}{\sum_{a'} \exp(\theta^\top \phi(s,a'))}$$

2. Neural Policy:

Neural network $f_\theta : S \times A \mapsto \mathbb{R}$

$$\pi_\theta(a \mid s) = \frac{\exp(f_\theta(s,a))}{\sum_{a'} \exp(f_\theta(s,a'))}$$

Softmax Policy Properties

$$\pi_\theta(a \mid s) = \frac{\exp(\theta^\top \phi(s, a))}{\sum_{a'} \exp(\theta^\top \phi(s, a'))}$$

Softmax Policy Properties

$$\pi_\theta(a \mid s) = \frac{\exp(\theta^\top \phi(s,a))}{\sum_{a'} \exp(\theta^\top \phi(s,a'))}$$

Two properties (see HW):

Softmax Policy Properties

$$\pi_\theta(a \mid s) = \frac{\exp(\theta^\top \phi(s, a))}{\sum_{a'} \exp(\theta^\top \phi(s, a'))}$$

Two properties (see HW):

- More probable actions have features which align with θ. Precisely,
  $\pi_\theta(a \mid s) \geq \pi_\theta(a' \mid s)$ if and only if $\theta^\top \phi(s, a) \geq \theta^\top \phi(s, a')$

Softmax Policy Properties

$$\pi_\theta(a \mid s) = \frac{\exp(\theta^\top \phi(s,a))}{\sum_{a'} \exp(\theta^\top \phi(s,a'))}$$

Two properties (see HW):

- More probable actions have features which align with θ. Precisely,
  $\pi_\theta(a \mid s) \geq \pi_\theta(a' \mid s)$ if and only if $\theta^\top \phi(s,a) \geq \theta^\top \phi(s,a')$
- The gradient of the log policy is:
  $\nabla_\theta \log(\pi_\theta(a \mid s)) = \phi(s,a) - \mathbb{E}_{a' \sim \pi_\theta(\cdot \mid s)}[\phi(s,a')]$

Softmax Policy Properties

$$\pi_\theta(a \mid s) = \frac{\exp(\theta^\top \phi(s,a))}{\sum_{a'} \exp(\theta^\top \phi(s,a'))}$$

Two properties (see HW):

- More probable actions have features which align with θ. Precisely,
  $\pi_\theta(a \mid s) \geq \pi_\theta(a' \mid s)$ if and only if $\theta^\top \phi(s,a) \geq \theta^\top \phi(s,a')$
- The gradient of the log policy is:
  $\nabla_\theta \log(\pi_\theta(a \mid s)) = \phi(s,a) - \mathbb{E}_{a' \sim \pi_\theta(\cdot \mid s)}[\phi(s,a')]$

- We have:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \rho_\theta}\left[\sum_{h=0}^{H-1} Q_h^{\pi_\theta}(s_h, a_h)\left(\phi(s_h, a_h) - \mathbb{E}_{a' \sim \pi_\theta(\cdot \mid s_h)}[\phi(s_h, a')]\right)\right]$$

$$= \mathbb{E}_{\tau \sim \rho_\theta}\left[\sum_{h=0}^{H-1} A_h^{\pi_\theta}(s_h, a_h)\phi(s_h, a_h)\right]$$

Summary:

1. REINFORCE (a direct application of the likelihood ratio method)
2. Variance Reduction: with baselines

Attendance:

bit.ly/3RcTC9T

Feedback:

bit.ly/3RHtlxy