Off-Policy Policy Gradient

CS 185/285

Instructor: Sergey Levine
UC Berkeley

Part 1:
Back to policy gradients

Why do we *like* policy gradients?

a.k.a. on-policy batch actor-critic

policy gradient with GAE:

1. sample $\{\tau^{(i)}\}$ from π_θ (run the policy)
2. evaluate $y_t^{(i)} = r(\mathbf{s}_t^{(i)}, \mathbf{a}_t^{(i)}) + \hat{V}_\phi^\pi(\mathbf{s}_{t+1}^{(i)})$
3. fit $\hat{V}_\phi^\pi(\mathbf{s})$ to targets $\{y_t^{(i)}\}$
4. evaluate $\hat{A}^\pi_{\text{GAE}}(\mathbf{s}_t^{(i)}, \mathbf{a}_t(i)) = \sum_{t'=t}^{\infty} (\gamma\lambda)^{t'-t} \delta_{t'}^{(i)}$
5. $\nabla_\theta J(\theta) \approx \sum_i \sum_t \nabla_\theta \log \pi_\theta(\mathbf{a}_t^{(i)} | \mathbf{s}_t^{(i)}) \hat{A}^\pi(\mathbf{s}_t^{(i)}, \mathbf{a}_t^{(i)})$
6. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

This makes policy gradients a great choice if we want a stable, reliable RL method, and we are not that concerned about sample efficiency

Particularly good for sim-to-real robotics and for training LLMs

Q-function methods can be hard to tune and might not converge

With on-policy methods, we can use Monte Carlo advantage estimators

$$\lambda = 1$$

$$\hat{A}^\pi_{\text{MC}}(\mathbf{s}_t, \mathbf{a}_t) = \sum_{t'=t}^{\infty} \gamma^{t'-t} r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) - \hat{V}_\phi^\pi(\mathbf{s}_t)$$

This gives us a provably **unbiased** policy gradient estimator

no matter how “wrong” the critic is!

The problem with policy gradients

a.k.a. on-policy batch actor-critic

policy gradient with GAE:

1. sample $\{\tau^{(i)}\}$ from π_θ (run the policy)
2. evaluate $y_t^{(i)} = r(\mathbf{s}_t^{(i)}, \mathbf{a}_t^{(i)}) + \hat{V}_\phi^\pi(\mathbf{s}_{t+1}^{(i)})$
3. fit $\hat{V}_\phi^\pi(\mathbf{s})$ to targets $\{y_t^{(i)}\}$
4. evaluate $\hat{A}_{\text{GAE}}^\pi(\mathbf{s}_t^{(i)}, \mathbf{a}_t(i)) = \sum_{t'=t}^{\infty} (\gamma\lambda)^{t'-t} \delta_{t'}^{(i)}$
5. $\nabla_\theta J(\theta) \approx \sum_i \sum_t \nabla_\theta \log \pi_\theta(\mathbf{a}_t^{(i)} | \mathbf{s}_t^{(i)}) \hat{A}^\pi(\mathbf{s}_t^{(i)}, \mathbf{a}_t^{(i)})$
6. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

each iteration takes **one** gradient step

each iteration requires generating new samples

high variance means we usually want **bigger** batches than we would use with regular supervised learning, and smaller learning rates

this quickly becomes intractable

The kind of algorithm we would like to see

fictional "multi-step" policy gradient:

1. sample $\{\tau^{(i)}\}$ from π_θ (run the policy)
2. evaluate $y_t^{(i)} = r(\mathbf{s}_t^{(i)}, \mathbf{a}_t^{(i)}) + \hat{V}_\phi^\pi(\mathbf{s}_{t+1}^{(i)})$
3. fit $\hat{V}_\phi^\pi(\mathbf{s})$ to targets $\{y_t^{(i)}\}$
4. evaluate $\hat{A}^\pi_{\text{GAE}}(\mathbf{s}_t^{(i)}, \mathbf{a}_t(i)) = \sum_{t'=t}^{\infty} (\gamma\lambda)^{t'-t} \delta_{t'}^{(i)}$
5. $\nabla_\theta J(\theta) \approx \sum_i \sum_t \nabla_\theta \log \pi_\theta(\mathbf{a}_t^{(i)}|\mathbf{s}_t^{(i)}) \hat{A}^\pi(\mathbf{s}_t^{(i)}, \mathbf{a}_t^{(i)})$
6. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

$K\times$

why is this not allowed?

$$\mathbb{E}_{\tau \sim p_\theta(\mathbf{s}_t, \mathbf{a}_t)} [\nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) \hat{A}^\pi(\mathbf{s}_t, \mathbf{a}_t)]$$

$$= \sum_{\mathbf{s}_t, \mathbf{a}_t} p_\theta(\mathbf{s}_t, \mathbf{a}_t) \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) \hat{A}^\pi(\mathbf{s}_t, \mathbf{a}_t)$$

can we somehow *correct* this so that we can take multiple steps?

Part 2:
Importance sampling

Off-policy learning & importance sampling

$$\theta^\star = \arg\max_\theta J(\theta)$$

$$J(\theta) = E_{\tau\sim p_\theta(\tau)}[r(\tau)]$$

what if we don't have samples from $p_\theta(\tau)$?

(we have samples from some $\bar{p}(\tau)$ instead)

$$J(\theta) = E_{\tau\sim\bar{p}(\tau)}\left[\frac{p_\theta(\tau)}{\bar{p}(\tau)}r(\tau)\right]$$

$$p_\theta(\tau) = p(\mathbf{s}_1)\prod_{t=1}^{H}\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)p(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{a}_t)$$

$$\frac{p_\theta(\tau)}{\bar{p}(\tau)} = \frac{\cancel{p(\mathbf{s}_1)}\prod_{t=1}^{H}\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)\cancel{p(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{a}_t)}}{\cancel{p(\mathbf{s}_1)}\prod_{t=1}^{H}\bar{\pi}(\mathbf{a}_t|\mathbf{s}_t)\cancel{p(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{a}_t)}} = \frac{\prod_{t=1}^{H}\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}{\prod_{t=1}^{H}\bar{\pi}(\mathbf{a}_t|\mathbf{s}_t)}$$

importance sampling

$$\begin{aligned}E_{x\sim p(x)}[f(x)] &= \int p(x)f(x)dx \\ &= \int \frac{q(x)}{q(x)}p(x)f(x)dx \\ &= \int q(x)\frac{p(x)}{q(x)}f(x)dx \\ &= E_{x\sim q(x)}\left[\frac{p(x)}{q(x)}f(x)\right]\end{aligned}$$

Deriving the policy gradient with IS

$$\theta^\star = \arg\max_\theta J(\theta)$$

$$J(\theta) = E_{\tau\sim p_\theta(\tau)}[r(\tau)]$$

a convenient identity

$$p_\theta(\tau)\nabla_\theta \log p_\theta(\tau) = \nabla_\theta p_\theta(\tau)$$

can we estimate the value of some *new* parameters θ'?

$$J(\theta') = E_{\tau\sim p_\theta(\tau)}\left[\frac{p_{\theta'}(\tau)}{p_\theta(\tau)} r(\tau)\right]$$

the only bit that depends on θ'

$$\nabla_{\theta'} J(\theta') = E_{\tau\sim p_\theta(\tau)}\left[\frac{\nabla_{\theta'} p_{\theta'}(\tau)}{p_\theta(\tau)} r(\tau)\right] = E_{\tau\sim p_\theta(\tau)}\left[\frac{p_{\theta'}(\tau)}{p_\theta(\tau)} \nabla_{\theta'} \log p_{\theta'}(\tau) r(\tau)\right]$$

now estimate locally, at $\theta = \theta'$: $\nabla_\theta J(\theta) = E_{\tau\sim p_\theta(\tau)}[\nabla_\theta \log p_\theta(\tau) r(\tau)]$

The off-policy policy gradient

$$\theta^\star = \arg\max_\theta J(\theta) \qquad J(\theta) = E_{\tau\sim p_\theta(\tau)}[r(\tau)]$$

$$\nabla_{\theta'} J(\theta') = E_{\tau\sim p_\theta(\tau)}\left[\frac{p_{\theta'}(\tau)}{p_\theta(\tau)}\nabla_{\theta'}\log\pi_{\theta'}(\tau)r(\tau)\right] \quad \text{when } \theta \neq \theta'$$

$$\frac{p_{\theta'}(\tau)}{p_\theta(\tau)} = \frac{\prod_{t=1}^H \pi_{\theta'}(\mathbf{a}_t|\mathbf{s}_t)}{\prod_{t=1}^H \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}$$

$$= E_{\tau\sim p_\theta(\tau)}\left[\left(\prod_{t=1}^H \frac{\pi_{\theta'}(\mathbf{a}_t|\mathbf{s}_t)}{\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}\right)\left(\sum_{t=1}^H \nabla_{\theta'}\log\pi_{\theta'}(\mathbf{a}_t|\mathbf{s}_t)\right)\left(\sum_{t=1}^H r(\mathbf{s}_t,\mathbf{a}_t)\right)\right] \text{ what about causality?}$$

$$= E_{\tau\sim p_\theta(\tau)}\left[\sum_{t=1}^H \nabla_{\theta'}\log\pi_{\theta'}(\mathbf{a}_t|\mathbf{s}_t)\left(\underline{\prod_{t'=1}^t \frac{\pi_{\theta'}(\mathbf{a}_{t'}|\mathbf{s}_{t'})}{\pi_\theta(\mathbf{a}_{t'}|\mathbf{s}_{t'})}}\right)\left(\sum_{t'=t}^H r(\mathbf{s}_{t'},\mathbf{a}_{t'})\left(\cancel{\prod_{t''=t}^{t'} \frac{\pi_{\theta'}(\mathbf{a}_{t''}|\mathbf{s}_{t''})}{\pi_\theta(\mathbf{a}_{t''}|\mathbf{s}_{t''})}}\right)\right)\right]$$

future actions don't affect current weight

if we ignore this, we get
a policy iteration algorithm
(more on this next time)

A first-order approximation for IS

$$\nabla_{\theta'} J(\theta') = E_{\tau \sim p_\theta(\tau)} \left[\sum_{t=1}^{H} \nabla_{\theta'} \log \pi_{\theta'}(\mathbf{a}_t|\mathbf{s}_t) \left(\prod_{t'=1}^{t} \frac{\pi_{\theta'}(\mathbf{a}_{t'}|\mathbf{s}_{t'})}{\pi_\theta(\mathbf{a}_{t'}|\mathbf{s}_{t'})} \right) \left(\sum_{t'=t}^{H} r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) \right) \right]$$

exponential in T...

let's write the objective a bit differently...

on-policy policy gradient:

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{H} \nabla_\theta \log \pi_\theta(\mathbf{a}_t^{(i)}|\mathbf{s}_t^{(i)}) \hat{A}_t^{(i)}$$

$(\mathbf{s}_t^{(i)}, \mathbf{a}_t^{(i)}) \sim \pi_\theta(\mathbf{s}_t, \mathbf{a}_t)$

off-policy policy gradient:

$$\nabla_{\theta'} J(\theta') \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{H} \frac{\pi_{\theta'}(\mathbf{s}_t^{(i)}, \mathbf{a}_t^{(i)})}{\pi_\theta(\mathbf{s}_t^{(i)}, \mathbf{a}_t^{(i)})} \nabla_{\theta'} \log \pi_{\theta'}(\mathbf{a}_t^{(i)}|\mathbf{s}_t^{(i)}) \hat{A}_t^{(i)}$$

this is "a little bit wrong"
we'll see next time why
it's mostly ok to do this

$$= \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{H} \frac{\pi_{\theta'}(\mathbf{s}_t^{(i)})}{\pi_\theta(\mathbf{s}_t^{(i)})} \frac{\pi_{\theta'}(\mathbf{a}_t^{(i)}|\mathbf{s}_t^{(i)})}{\pi_\theta(\mathbf{a}_t^{(i)}|\mathbf{s}_t^{(i)})} \nabla_{\theta'} \log \pi_{\theta'}(\mathbf{a}_t^{(i)}|\mathbf{s}_t^{(i)}) \hat{A}_t^{(i)}$$

ignore this part

The kind of algorithm we would like to see

actual "multi-step" policy gradient:

1. sample $\{\tau^{(i)}\}$ from π_θ (run the policy)
2. evaluate $y_t^{(i)} = r(\mathbf{s}_t^{(i)}, \mathbf{a}_t^{(i)}) + \hat{V}_\phi^\pi(\mathbf{s}_{t+1}^{(i)})$
3. fit $\hat{V}_\phi^\pi(\mathbf{s})$ to targets $\{y_t^{(i)}\}$
4. evaluate $\hat{A}^\pi_{\text{GAE}}(\mathbf{s}_t^{(i)}, \mathbf{a}_t(i)) = \sum_{t'=t}^{\infty} (\gamma\lambda)^{t'-t} \delta_{t'}^{(i)}$
5. $\theta' \leftarrow \theta$
6. $\nabla_{\theta'} J(\theta') \approx \sum_{i=1}^{N} \sum_{t=1}^{H} \frac{\pi_{\theta'}(\mathbf{a}_t^{(i)}|\mathbf{s}_t^{(i)})}{\pi_\theta(\mathbf{a}_t^{(i)}|\mathbf{s}_t^{(i)})} \nabla_{\theta'} \log \pi_{\theta'}(\mathbf{a}_t^{(i)}|\mathbf{s}_t^{(i)}) \hat{A}_t^{(i)}$ ($K\times$)
7. $\theta' \leftarrow \theta' + \alpha \nabla_{\theta'} J(\theta')$
6. $\theta \leftarrow \theta'$

Part 3:
Making importance sampling work

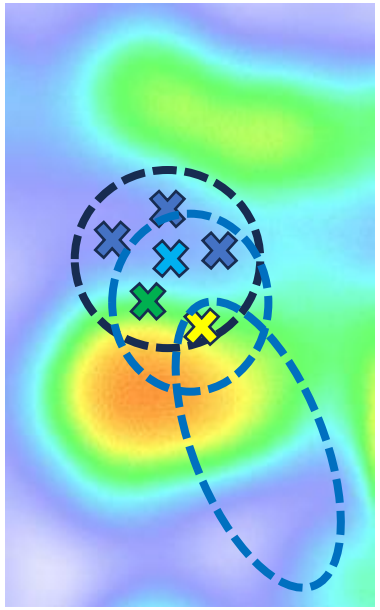
There is still a problem

actual "multi-step" policy gradient:

1. sample $\{\tau^{(i)}\}$ from π_θ (run the policy)
2. evaluate $y_t^{(i)} = r(\mathbf{s}_t^{(i)}, \mathbf{a}_t^{(i)}) + \hat{V}_\phi^\pi(\mathbf{s}_{t+1}^{(i)})$
3. fit $\hat{V}_\phi^\pi(\mathbf{s})$ to targets $\{y_t^{(i)}\}$
4. evaluate $\hat{A}^\pi_{\text{GAE}}(\mathbf{s}_t^{(i)}, \mathbf{a}_t(i)) = \sum_{t'=t}^{\infty} (\gamma\lambda)^{t'-t} \delta_{t'}^{(i)}$
5. $\theta' \leftarrow \theta$
6. $\nabla_{\theta'} J(\theta') \approx \sum_{i=1}^{N} \sum_{t=1}^{H} \frac{\pi_{\theta'}(\mathbf{a}_t^{(i)}|\mathbf{s}_t^{(i)})}{\pi_\theta(\mathbf{a}_t^{(i)}|\mathbf{s}_t^{(i)})} \nabla_{\theta'} \log \pi_{\theta'}(\mathbf{a}_t^{(i)}|\mathbf{s}_t^{(i)}) \hat{A}_t^{(i)}$
7. $\theta' \leftarrow \theta' + \alpha \nabla_{\theta'} J(\theta')$

($K\times$ steps 6–7)

6. $\theta \leftarrow \theta'$



once the weights are too far from 1.0,
the estimator is no longer useful

formally: its variance becomes too high

Clipped importance weights

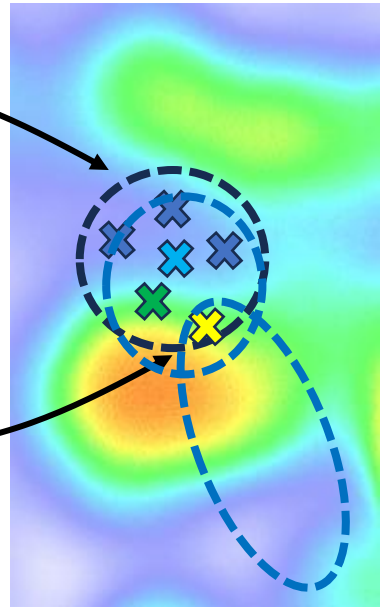
$$J(\theta') = E_{\tau \sim p_\theta(\tau)} \left[\overbrace{\frac{p_{\theta'}(\tau)}{p_\theta(\tau)}}^{w(\tau)} r(\tau) \right]$$

"surrogate objective"

$$\nabla_{\theta'} J(\theta') = E_{\tau \sim p_\theta(\tau)} \left[\frac{p_{\theta'}(\tau)}{p_\theta(\tau)} \nabla_{\theta'} \log p_{\theta'}(\tau) r(\tau) \right]$$

$w(\tau)$ keeps decreasing here

$w(\tau)$ keeps increasing here



try 0.1

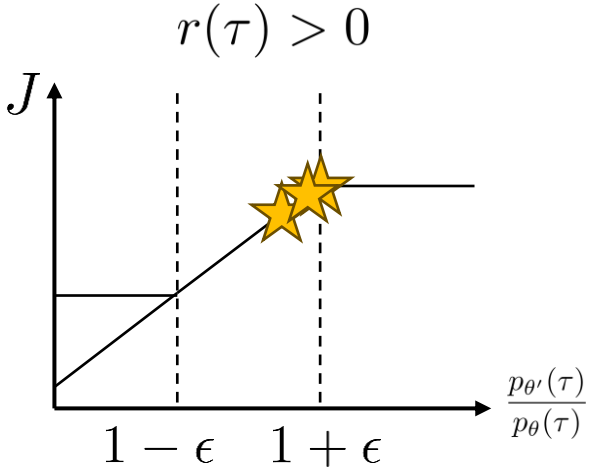
$$\text{redefine} \quad w(\tau) = \max \left\{ 1 - \epsilon, \min \left\{ 1 + \epsilon, \frac{p_{\theta'}(\tau)}{p_\theta(\tau)} \right\} \right\}$$

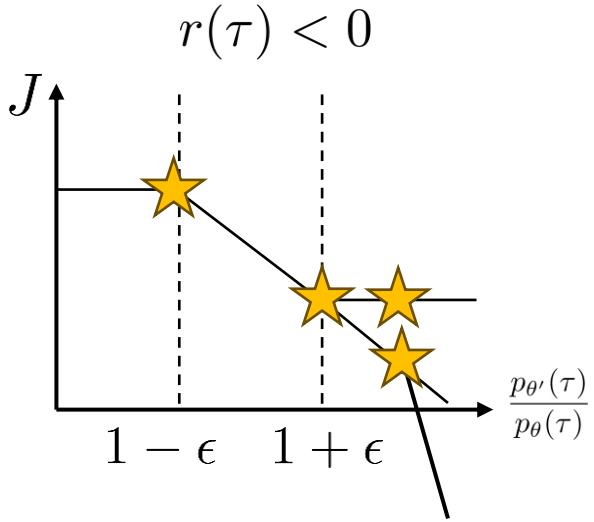
puts a limit on how much probabilities can *increase* or *decrease*

One more detail

$$J(\theta') = E_{\tau \sim p_\theta(\tau)} \left[w(\tau) r(\tau) \right]$$

$$w(\tau) = \max \left\{ 1 - \epsilon, \min \left\{ 1 + \epsilon, \frac{p_{\theta'}(\tau)}{p_\theta(\tau)} \right\} \right\}$$





not paying any price for drifting away from data here!

$$J(\theta') = E_{\tau \sim p_\theta(\tau)} \left[\min\{ w(\tau) r(\tau), w_c(\tau) r(\tau) \} \right]$$

$$w(\tau) = \frac{p_{\theta'}(\tau)}{p_\theta(\tau)}$$

$$w_c(\tau) = \max \left\{ 1 - \epsilon, \min \left\{ 1 + \epsilon, \frac{p_{\theta'}(\tau)}{p_\theta(\tau)} \right\} \right\}$$

Clipped importance sampling policy gradient

policy gradient with clipped IS (PPO):

1. sample $\{\tau^{(i)}\}$ from π_θ (run the policy)
2. evaluate $y_t^{(i)} = r(\mathbf{s}_t^{(i)}, \mathbf{a}_t^{(i)}) + \hat{V}_\phi^\pi(\mathbf{s}_{t+1}^{(i)})$
3. fit $\hat{V}_\phi^\pi(\mathbf{s})$ to targets $\{y_t^{(i)}\}$
4. evaluate $\hat{A}^\pi_{\text{GAE}}(\mathbf{s}_t^{(i)}, \mathbf{a}_t(i)) = \sum_{t'=t}^{\infty} (\gamma\lambda)^{t'-t} \delta_{t'}^{(i)}$
5. $\theta' \leftarrow \theta$
6. $\nabla_{\theta'} J(\theta') \approx \nabla_{\theta'} \mathcal{L}_{\text{CLIP}}(\theta')$ ($K\times$)
7. $\theta' \leftarrow \theta' + \alpha \nabla_{\theta'} J(\theta')$
6. $\theta \leftarrow \theta'$

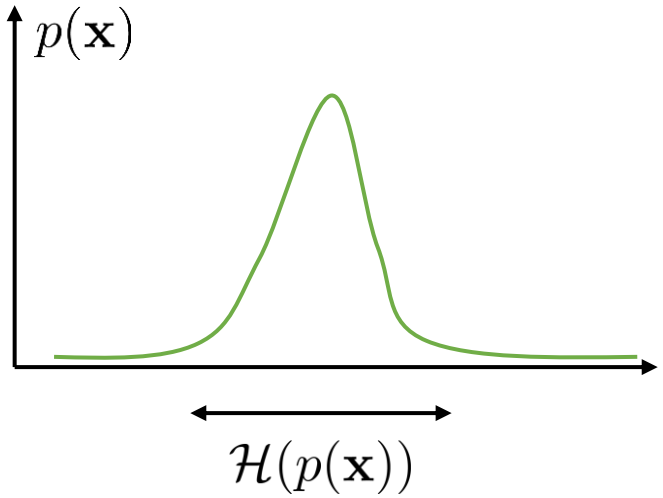
this results in the widely used PPO algorithm

Some more details:

use GAE for $y_t^{(i)}$ as well

add $\sum_i \sum_t \mathcal{H}(\pi_{\theta'}(\cdot|\mathbf{s}_t^{(i)}))$ to objective

"entropy regularization"



$$\mathcal{L}_{\text{CLIP}}(\theta') = \sum_{i=1}^{N} \sum_{t=1}^{H} \min \left\{ \frac{\pi_{\theta'}(\mathbf{a}_t^{(i)}|\mathbf{s}_t^{(i)})}{\pi_\theta(\mathbf{a}_t^{(i)}|\mathbf{s}_t^{(i)})} \hat{A}_t^{(i)}, \text{clip}\left(\frac{\pi_{\theta'}(\mathbf{a}_t^{(i)}|\mathbf{s}_t^{(i)})}{\pi_\theta(\mathbf{a}_t^{(i)}|\mathbf{s}_t^{(i)})}, 1-\epsilon, 1+\epsilon \right) \hat{A}_t^{(i)} \right\}$$