Advanced Policy Gradients

CS 285: Deep Reinforcement Learning, Decision Making, and Control
Sergey Levine
Class Notes

1. Homework 2 due today (11:59 pm)!
   • Don’t be late!

2. Homework 3 comes out this week
   • Start early! Q-learning takes a while to run
Today’s Lecture

1. Why does policy gradient work?
2. Policy gradient is a type of policy iteration
3. Policy gradient as a constrained optimization
4. From constrained optimization to natural gradient
5. Natural gradients and trust regions

• Goals:
  • Understand the policy iteration view of policy gradient
  • Understand how to analyze policy gradient improvement
  • Understand what natural gradient does and how to use it
Recap: policy gradients

REINFORCE algorithm:
1. sample \{τ^i\} from \(π_θ(a_t|s_t)\) (run the policy)
2. \(∇_θ J(θ) \approx \sum_i \left( \sum_{t=1}^T ∇_θ \log π_θ(a^i_t|s^i_t) \left( \sum_{t'=t}^T r(s^i_{t'}, a^i_{t'}) \right) \right)\)
3. \(θ ← θ + α∇_θ J(θ)\)

\[\nabla_θ J(θ) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_θ \log π_θ(a_{i,t}|s_{i,t}) \hat{Q}^π_{i,t}\]

“reward to go”

can also use function approximation here
Why does policy gradient work?

\[ \nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_{i,t}|s_{i,t}) \hat{A}^\pi_{i,t} \]

1. Estimate \( \hat{A}^\pi(s_t, a_t) \) for current policy \( \pi \)
2. Use \( \hat{A}^\pi(s_t, a_t) \) to get improved policy \( \pi' \)

look familiar?

policy iteration algorithm:

1. evaluate \( A^\pi(s, a) \)
2. set \( \pi \leftarrow \pi' \)
Policy gradient as policy iteration

\[ J(\theta') - J(\theta) = J(\theta') - E_{s_0 \sim p(s_0)} [V^{\pi_\theta}(s_0)] \]

\[ = J(\theta') - E_{\tau \sim p_\theta(\tau)} [V^{\pi_\theta}(s_0)] \]

claim: \[ J(\theta') - J(\theta) = E_{\tau \sim p_\theta(\tau)} \left[ \sum_{t} \gamma^t r(s_t, a_t) \right] \]

\[ = J(\theta') - E_{\tau \sim p_\theta(\tau)} \left[ \sum_{t=0}^{\infty} \gamma^t V^{\pi_\theta}(s_t) - \sum_{t=1}^{\infty} \gamma^t V^{\pi_\theta}(s_t) \right] \]

\[ = J(\theta') + E_{\tau \sim p_\theta(\tau)} \left[ \sum_{t=0}^{\infty} \gamma^t (\gamma V^{\pi_\theta}(s_{t+1}) - V^{\pi_\theta}(s_t)) \right] \]

\[ = E_{\tau \sim p_\theta(\tau)} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] + E_{\tau \sim p_\theta(\tau)} \left[ \sum_{t=0}^{\infty} \gamma^t (\gamma V^{\pi_\theta}(s_{t+1}) - V^{\pi_\theta}(s_t)) \right] \]

\[ = E_{\tau \sim p_\theta(\tau)} \left[ \sum_{t=0}^{\infty} \gamma^t (r(s_t, a_t) + V^{\pi_\theta}(s_{t+1}) - V^{\pi_\theta}(s_t)) \right] \]

\[ = E_{\tau \sim p_\theta(\tau)} \left[ \sum_{t=0}^{\infty} \gamma^t A^{\pi_\theta}(s_t, a_t) \right] \]
Policy gradient as **policy iteration**

\[ J(\theta') - J(\theta) = E_{\tau \sim p_{\theta'}}(\tau) \left[ \sum_{t} \gamma^t A_{\pi_{\theta}}(s_t, a_t) \right] \]

- Expectation under \( \pi_{\theta'} \)
- Advantage under \( \pi_{\theta} \)

**importance sampling**

\[
E_{x \sim p(x)}[f(x)] = \int p(x)f(x)dx \\
= \int \frac{q(x)}{q(x)} p(x)f(x)dx \\
= \int q(x) \frac{p(x)}{q(x)} f(x)dx \\
= E_{x \sim q(x)} \left[ \frac{p(x)}{q(x)} f(x) \right]
\]

\[
E_{\tau \sim p_{\theta'}}(\tau) \left[ \sum_{t} \gamma^t A_{\pi_{\theta}}(s_t, a_t) \right] = \sum_{t} E_{s_t \sim p_{\theta'}(s_t)} \left[ E_{a_t \sim \pi_{\theta'}(a_t | s_t)} \left[ \gamma^t A_{\pi_{\theta}}(s_t, a_t) \right] \right] \\
= \sum_{t} E_{s_t \sim p_{\theta'}(s_t)} \left[ E_{a_t \sim \pi_{\theta}(a_t | s_t)} \left[ \frac{\pi_{\theta'}(a_t | s_t)}{\pi_{\theta}(a_t | s_t)} \gamma^t A_{\pi_{\theta}}(s_t, a_t) \right] \right]
\]

is it OK to use \( p_{\theta}(s_t) \) instead?
Ignoring distribution mismatch?

\[
\sum_t E_{s_t \sim p_{\theta'}(s_t)} \left[ E_{a_t \sim \pi_\theta(a_t|s_t)} \left[ \frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)} \gamma^t A^{\pi_\theta}(s_t, a_t) \right] \right] \approx \sum_t E_{s_t \sim p_{\theta}(s_t)} \left[ E_{a_t \sim \pi_\theta(a_t|s_t)} \left[ \frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)} \gamma^t A^{\pi_\theta}(s_t, a_t) \right] \right]
\]

why do we want this to be true?

\[ J(\theta') - J(\theta) \approx \bar{A}(\theta') \quad \Rightarrow \quad \theta' \leftarrow \arg \max_{\theta'} \bar{A}(\theta) \]

2. Use \( \hat{A}^{\pi}(s_t, a_t) \) to get improved policy \( \pi' \)

is it true? and when?

Claim: \( p_{\theta}(s_t) \) is close to \( p_{\theta'}(s_t) \) when \( \pi_\theta \) is close to \( \pi_{\theta'} \).
Bounding the distribution change

Claim: \( p_\theta(s_t) \) is close to \( p_{\theta'}(s_t) \) when \( \pi_\theta \) is close to \( \pi_{\theta'} \)

Simple case: assume \( \pi_\theta \) is a deterministic policy \( a_t = \pi_\theta(s_t) \)

\( \pi_{\theta'} \) is close to \( \pi_\theta \) if \( \pi_{\theta'}(a_t \neq \pi_\theta(s_t)|s_t) \leq \epsilon \)

\[
p_{\theta'}(s_t) = (1 - \epsilon)^t p_{\theta}(s_t) + (1 - (1 - \epsilon)^t))p_{\text{mistake}}(s_t)
\]


probability we made no mistakes some other distribution

\[
|p_{\theta'}(s_t) - p_{\theta}(s_t)| = (1 - (1 - \epsilon)^t)|p_{\text{mistake}}(s_t) - p_{\theta}(s_t)| \leq 2(1 - (1 - \epsilon)^t)
\]

useful identity: \( (1 - \epsilon)^t \geq 1 - \epsilon t \) for \( \epsilon \in [0,1] \)

\[
\leq 2\epsilon t
\]

not a great bound, but a bound!
Bounding the distribution change

Claim: $p_\theta(s_t)$ is close to $p_{\theta'}(s_t)$ when $\pi_\theta$ is close to $\pi_{\theta'}$.

General case: assume $\pi_\theta$ is an arbitrary distribution

$\pi_{\theta'}$ is close to $\pi_\theta$ if $|\pi_{\theta'}(a_t|s_t) - \pi_\theta(a_t|s_t)| \leq \epsilon$ for all $s_t$

Useful lemma: if $|p_X(x) - p_Y(x)| = \epsilon$, exists $p(x, y)$ such that $p(x) = p_X(x)$ and $p(y) = p_Y(y)$ and $p(x = y) = 1 - \epsilon$

$\Rightarrow p_X(x)$ “agrees” with $p_Y(y)$ with probability $\epsilon$

$\Rightarrow \pi_{\theta'}(a_t|s_t)$ takes a different action than $\pi_\theta(a_t|s_t)$ with probability at most $\epsilon$

$|p_{\theta'}(s_t) - p_\theta(s_t)| = (1 - (1 - \epsilon)^t)|p_{\text{mistake}}(s_t) - p_\theta(s_t)| \leq 2(1 - (1 - \epsilon)^t) \leq 2\epsilon t$

Bounding the objective value

\( \pi_{\theta'} \) is close to \( \pi_\theta \) if \( |\pi_{\theta'}(a_t|s_t) - \pi_\theta(a_t|s_t)| \leq \epsilon \) for all \( s_t \)

\[ |p_{\theta'}(s_t) - p_\theta(s_t)| \leq 2\epsilon t \]

\[
E_{p_{\theta'}(s_t)}[f(s_t)] = \sum_{s_t} p_{\theta'}(s_t)f(s_t) \geq \sum_{s_t} p_\theta(s_t)f(s_t) - |p_\theta(s_t) - p_{\theta'}(s_t)| \max_{s_t} f(s_t)
\]

\[
\geq E_{p_\theta(s_t)}[f(s_t)] - 2\epsilon t \max_{s_t} f(s_t)
\]

\[
\sum_t E_{s_t \sim p_{\theta'}(s_t)} \left[ E_{a_t \sim \pi_\theta(a_t|s_t)} \left[ \frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)} \gamma^t A^\pi_\theta(s_t,a_t) \right] \right] \geq O(Tr_{\max}) \text{ or } O\left(\frac{r_{\max}}{1-\gamma}\right)
\]

\[
\sum_t E_{s_t \sim p_\theta(s_t)} \left[ E_{a_t \sim \pi_\theta(a_t|s_t)} \left[ \frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)} \gamma^t A^\pi_\theta(s_t,a_t) \right] \right] - \sum_t 2\epsilon t C
\]

maximizing this maximizes a bound on the thing we want!
Where are we at so far?

\[ \theta' \leftarrow \arg \max_{\theta'} \sum_t E_{s_t \sim p_\theta(s_t)} \left[ E_{a_t \sim \pi_\theta(a_t|s_t)} \left[ \frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)} \gamma^t A^\pi_\theta(s_t, a_t) \right] \right] \]

such that \[ |\pi_{\theta'}(a_t|s_t) - \pi_\theta(a_t|s_t)| \leq \epsilon \]

for small enough \( \epsilon \), this is guaranteed to improve \( J(\theta') - J(\theta) \)
Break
A more convenient bound

Claim: \( p_\theta(s_t) \) is close to \( p_{\theta'}(s_t) \) when \( \pi_\theta \) is close to \( \pi_{\theta'} \)

\( \pi_{\theta'} \) is close to \( \pi_\theta \) if \( |\pi_{\theta'}(a_t|s_t) - \pi_\theta(a_t|s_t)| \leq \epsilon \) for all \( s_t \)

\[ |p_{\theta'}(s_t) - p_{\theta}(s_t)| \leq 2\epsilon t \]

A more convenient bound:

\[ |\pi_{\theta'}(a_t|s_t) - \pi_\theta(a_t|s_t)| \leq \sqrt{\frac{1}{2} D_{KL}(\pi_{\theta'}(a_t|s_t)||\pi_\theta(a_t|s_t))} \]

\[ \Rightarrow D_{KL}(\pi_{\theta'}(a_t|s_t)||\pi_\theta(a_t|s_t)) \text{ bounds state marginal difference} \]

\[ D_{KL}(p_1(x)||p_2(x)) = E_{x \sim p_1(x)} \left[ \log \frac{p_1(x)}{p_2(x)} \right] \]

KL divergence has some very convenient properties that make it much easier to approximate!
How do we optimize the objective?

\[
\theta' \leftarrow \arg \max_{\theta'} \sum_t E_{s_t \sim p_\theta(s_t)} \left[ E_{a_t \sim \pi_\theta(a_t|s_t)} \left[ \frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)} \gamma^t A_\pi(\theta, s_t, a_t) \right] \right]
\]

such that \( D_{KL}(\pi_{\theta'}(a_t|s_t) \parallel \pi_\theta(a_t|s_t)) \leq \epsilon \)

for small enough \( \epsilon \), this is guaranteed to improve \( J(\theta') - J(\theta) \)
How do we enforce the constraint?

$$\theta' \leftarrow \arg \max_{\theta'} \sum_{t} E_{s_t \sim p_{\theta}(s_t)} \left[ E_{a_t \sim \pi_{\theta}(a_t|s_t)} \left[ \frac{\pi_{\theta'}(a_t|s_t)}{\pi_{\theta}(a_t|s_t)} \gamma^t A_{\theta}(s_t, a_t) \right] \right]$$

such that $$D_{\text{KL}}(\pi_{\theta'}(a_t|s_t) || \pi_{\theta}(a_t|s_t)) \leq \epsilon$$

$$L(\theta', \lambda) = \sum_{t} E_{s_t \sim p_{\theta}(s_t)} \left[ E_{a_t \sim \pi_{\theta}(a_t|s_t)} \left[ \frac{\pi_{\theta'}(a_t|s_t)}{\pi_{\theta}(a_t|s_t)} \gamma^t A_{\theta}(s_t, a_t) \right] \right] - \lambda(D_{\text{KL}}(\pi_{\theta'}(a_t|s_t) || \pi_{\theta}(a_t|s_t)) - \epsilon)$$

1. Maximize $$L(\theta', \lambda)$$ with respect to $$\theta'$$

2. $$\lambda \leftarrow \lambda + \alpha(D_{\text{KL}}(\pi_{\theta'}(a_t|s_t) || \pi_{\theta}(a_t|s_t)) - \epsilon)$$

Intuition: raise $$\lambda$$ if constraint violated too much, else lower it

an instance of dual gradient descent (more on this later!)
How (else) do we optimize the objective?

\[ \theta' \leftarrow \arg \max_{\theta'} \sum_t E_{s_t \sim \rho(s_t)} \left[ E_{a_t \sim \pi_\theta(a_t|s_t)} \left[ \frac{\pi_\theta'(a_t|s_t)}{\pi_\theta(a_t|s_t)} \gamma^t A^{\pi_\theta}(s_t, a_t) \right] \right] \]

such that \( D_{KL}(\pi_\theta'(a_t|s_t)||\pi_\theta(a_t|s_t)) \leq \epsilon \)

for small enough \( \epsilon \), this is guaranteed to improve \( J(\theta') - J(\theta) \)

\[ \theta' \leftarrow \arg \max_{\theta'} \nabla_\theta \tilde{A}(\theta)^T (\theta' - \theta) \]

such that \( D_{KL}(\pi_\theta'(a_t|s_t)||\pi_\theta(a_t|s_t)) \leq \epsilon \)

Use first order Taylor approximation for objective (a.k.a., linearization)
How do we optimize the objective?

\[ \theta' \leftarrow \arg \max_{\theta'} \sum_t E_{s_t \sim p_{\theta}(s_t)} \left[ E_{a_t \sim \pi_{\theta}(a_t|s_t)} \left[ \frac{\pi_{\theta'}(a_t|s_t)}{\pi_{\theta}(a_t|s_t)} \gamma^t A^\theta(s_t, a_t) \right] \right] \]

such that \( D_{\text{KL}}(\pi_{\theta'}(a_t|s_t) \| \pi_{\theta}(a_t|s_t)) \leq \epsilon \)

\[ \theta' \leftarrow \arg \max_{\theta'} \nabla_{\theta'} \bar{A}(\theta)^T (\theta' - \theta) \]

such that \( D_{\text{KL}}(\pi_{\theta'}(a_t|s_t) \| \pi_{\theta}(a_t|s_t)) \leq \epsilon \)

\[ \nabla_{\theta'} \bar{A}(\theta') = \sum_t E_{s_t \sim p_{\theta}(s_t)} \left[ E_{a_t \sim \pi_{\theta}(a_t|s_t)} \left[ \frac{\pi_{\theta'}(a_t|s_t)}{\pi_{\theta}(a_t|s_t)} \gamma^t \nabla_{\theta'} \log \pi_{\theta'}(a_t|s_t) A^\theta(s_t, a_t) \right] \right] \]

(see policy gradient lecture for derivation)

\[ \nabla_{\theta} \bar{A}(\theta) = \sum_t E_{s_t \sim p_{\theta}(s_t)} \left[ E_{a_t \sim \pi_{\theta}(a_t|s_t)} \left[ \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta}(a_t|s_t)} \gamma^t \nabla_{\theta} \log \pi_{\theta}(a_t|s_t) A^\theta(s_t, a_t) \right] \right] \]

\[ \nabla_{\theta} \bar{A}(\theta) = \sum_t E_{s_t \sim p_{\theta}(s_t)} \left[ E_{a_t \sim \pi_{\theta}(a_t|s_t)} \left[ \gamma^t \nabla_{\theta} \log \pi_{\theta}(a_t|s_t) A^\theta(s_t, a_t) \right] \right] = \nabla_{\theta} J(\theta) \]

exactly the normal policy gradient!
Can we just use the gradient then?

\[ \theta' \leftarrow \arg \max_{\theta'} \nabla_\theta J(\theta)^T (\theta' - \theta) \]

such that \( D_{KL}(\pi_{\theta'}(a_t|s_t) \parallel \pi_\theta(a_t|s_t)) \leq \epsilon \)

\[ \theta \leftarrow \theta + \alpha \nabla_\theta J(\theta) \]

\[ \pi_\theta(a_t|s_t) \]

some parameters change probabilities a lot more than others!

Claim: gradient ascent does this:

\[ \theta' \leftarrow \arg \max_{\theta'} \nabla_\theta J(\theta)^T (\theta' - \theta) \]

such that \( \|\theta - \theta'\|^2 \leq \epsilon \)

\[ \theta' = \theta + \sqrt{\frac{\epsilon}{\|\nabla_\theta J(\theta)\|^2}} \nabla_\theta J(\theta) \]
Can we just use the gradient then?

\[ \theta' \leftarrow \arg \max_{\theta'} \nabla_\theta J(\theta)^T (\theta' - \theta) \]

such that \( D_{KL}(\pi_{\theta'}(a_t|s_t) || \pi_\theta(a_t|s_t)) \leq \epsilon \)

not the same!

\[ \theta' \leftarrow \arg \max_{\theta'} \nabla_\theta J(\theta)^T (\theta' - \theta) \]

such that \( ||\theta - \theta'||^2 \leq \epsilon \)

second order Taylor expansion

\[ D_{KL}(\pi_{\theta'} || \pi_\theta) \approx \frac{1}{2} (\theta' - \theta)^T F(\theta' - \theta) \]

Fisher-information matrix

\[ F = E_{\pi_\theta} [\nabla_\theta \log \pi_\theta(a|s) \nabla_\theta \log \pi_\theta(a|s)^T] \]

can estimate with samples
Can we just use the gradient then?

\[ \theta' \leftarrow \arg \max_{\theta'} \nabla_\theta J(\theta)^T (\theta' - \theta) \]

such that \( D_{KL}(\pi_{\theta'}(a_t|s_t)||\pi_\theta(a_t|s_t)) \leq \epsilon \)

\[ D_{KL}(\pi_{\theta'}||\pi_\theta) \approx \frac{1}{2} (\theta' - \theta)^T F (\theta' - \theta) \]

\[ \theta' = \theta + \alpha F^{-1} \nabla_\theta J(\theta) \]

\[ \alpha = \sqrt{\frac{2\epsilon}{\nabla_\theta J(\theta)^T F \nabla_\theta J(\theta)}} \]

natural gradient
Is this even a problem in practice?

\[ r(s_t, a_t) = -s_t^2 - a_t^2 \]

\[ \log \pi_\theta(a_t | s_t) = -\frac{1}{2\sigma^2} (k s_t - a_t)^2 + \text{const} \quad \theta = (k, \sigma) \]

(a) 'Vanilla' policy gradients  
(b) Natural policy gradients

(image from Peters & Schaal 2008)

Essentially the same problem as this:
Practical methods and notes

• Natural policy gradient
  • Generally a good choice to stabilize policy gradient training
  • See this paper for details:
    • Peters, Schaal. Reinforcement learning of motor skills with policy gradients.
  • Practical implementation: requires efficient Fisher-vector products, a bit non-trivial to do without computing the full matrix
    • See: Schulman et al. Trust region policy optimization

\[ \theta' = \theta + \alpha F^{-1} \nabla_{\theta} J(\theta) \]

• Trust region policy optimization
• Just use the IS objective directly
  • Use regularization to stay close to old policy
  • See: Proximal policy optimization

\[ \alpha = \sqrt{\frac{2\epsilon}{\nabla_{\theta} J(\theta)^T F \nabla_{\theta} J(\theta)}} \]
Review

• Policy gradient = policy iteration
• Optimize advantage under new policy state distribution
• Using old policy state distribution optimizes a bound, if the policies are close enough
• Results in constrained optimization problem
• First order approximation to objective = gradient ascent
• Regular gradient ascent has the wrong constraint, use natural gradient
• Practical algorithms
  • Natural policy gradient
  • Trust region policy optimization