Nightmares of Policy Optimization

Sanjiban Choudhury
Can we just focus on finding a good policy?

\[ \pi_\theta : S_t \rightarrow a_t \]

Learn a mapping from states to actions

Roll-out policies in the real-world to estimate value
We assumed black-box policies ...
Have we redacted too much?

SUBJECT: 

I. [Redacted] is being s [Redacted] in the areas of [Redacted] until [Redacted]. The pro [Redacted]

2. This [Redacted] c [Redacted] into the [Redacted] areas, [Redacted] and [Redacted] will continue to be [Redacted].

3. The [Redacted] estimated to be in the [Redacted] required, under [Redacted].
Black-box vs White-box vs Gray-box

Black Box

White Box

\[ S_0 \xrightarrow{\pi_\theta} a_0 \xrightarrow{f(\cdot)} S_1 \xrightarrow{\pi_\theta} a_1 \xrightarrow{f(\cdot)} \ldots \xrightarrow{\pi_\theta} J(\theta) \]
Black-box vs White-box vs Gray-box
How can we take gradients if we don’t know the dynamics?
The Likelihood Ratio Trick!
Algorithm 20: The REINFORCE algorithm.

Start with an arbitrary initial policy $\pi_\theta$

while not converged do

Run simulator with $\pi_\theta$ to collect $\{\xi^{(i)}\}^{N}_{i=1}$

Compute estimated gradient

$$\tilde{\nabla}_\theta J = \frac{1}{N} \sum_{i=1}^{N} \left[ \left( \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta \left( a_t^{(i)} | s_t^{(i)} \right) \right) R(\xi^{(i)}) \right]$$

Update parameters $\theta \leftarrow \theta + \alpha \tilde{\nabla}_\theta J$

return $\pi_\theta$
Tetris Policy

\[
\pi_\theta(a|s) = \frac{\exp \left( \theta^\top f(s, a) \right)}{\sum_{a'} \exp \left( \theta^\top f(s, a') \right)}
\]

\[f_1(s, a) = \# \text{ number of holes}\]

\[f_2(s, a) = \# \text{ max height}\]
Chugging through the gradient ..

\[
\nabla_\theta \log \pi_\theta(a|s) = \nabla_\theta \left[ \theta^\top f(s,a) - \log \sum_{a'} \exp \left( \theta^\top f(s,a') \right) \right] \\
= f(s,a) - \frac{\sum_{a'} f(s,a') \exp \left( \theta^\top f(s,a') \right)}{\sum_{a'} \exp \left( \theta^\top f(s,a') \right)} \\
= f(s,a) - \sum_{a'} f(s,a') \pi_\theta(a'|s) \\
= f(s,a) - E_{\pi_\theta(a'|s)} \left[ f(s,a') \right]
\]

Understanding the REINFORCE update

Let \( f_1(s, a) = \# \text{ holes} \).

\[
\begin{align*}
R = +1 & \Rightarrow \sum_{s' \in S} f_1(s, a) - \mathbb{E}_{a' \sim \pi_\theta} f_1(s, a') = -5 \\
R = +1 & \Rightarrow \sum_{s' \in S} f_1(s, a) - \mathbb{E}_{a' \sim \pi_\theta} f_1(s, a') = +3
\end{align*}
\]

\[
\Theta_1 = \Theta_1 + \sum_{s \in S} \left( \nabla \log \pi_\theta(a|s) \right) R(s) = \Theta_1 + \alpha \left( -5 \times (+1) + 3 \times (-1) \right) = \Theta_1 - \alpha 8 \quad (\text{Bump down this feature})
\]
Algorithm 20: The REINFORCE algorithm.

Start with an arbitrary initial policy $\pi_\theta$

while not converged do

Run simulator with $\pi_\theta$ to collect $\{\xi^{(i)}\}_{i=1}^N$

Compute estimated gradient

$$\tilde{\nabla}_\theta J = \frac{1}{N} \sum_{i=1}^N \left[ \left( \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta \left( a_t^{(i)} | s_t^{(i)} \right) \right) R(\xi^{(i)}) \right]$$

Update parameters $\theta \leftarrow \theta + \alpha \tilde{\nabla}_\theta J$

return $\pi_\theta$
Causality: Can actions affect the past?
The Policy Gradient Theorem

$$
\nabla_\theta J = E_{p(\xi|\theta)} \left[ \sum_{t=0}^{T-1} \left( \nabla_\theta \log \pi_\theta(a_t|s_t) \left( \sum_{t'=0}^{t-1} r(s_{t'},a_{t'}) + \sum_{t'=t}^{T-1} r(s_{t'},a_{t'}) \right) \right) \right]
$$

$$
= E_{p(\xi|\theta)} \left[ \sum_{t=0}^{T-1} \left( \nabla_\theta \log \pi_\theta(a_t|s_t) \sum_{t'=t}^{T-1} r(s_{t'},a_{t'}) \right) \right],
$$

$$
\nabla_\theta J = E_{p(\xi|\theta)} \left[ \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) Q^{\pi_\theta}(s_t,a_t) \right]
$$
Life is good!

This solves everything ...
The Three Nightmares of Policy Optimization
Nightmare 1:

Local Optima
Activity!
Consider the following MDP

Let’s say I picked actions uniformly.
How long would it take me to get to the state with reward=1?

From Kakade and Langford
Think-Pair-Share

Think (30 sec): How long would it take me to get to the state with reward = 1? What does this imply if I run policy gradients?

Pair: Find a partner

Share (45 sec): Partners exchange ideas
Problem: Lack of exploration

Optimal policy
Problem: Lack of exploration

Optimal policy

Transition probabilities:

\[
\begin{align*}
\Pr(S_0) &= \frac{1}{10} \\
\Pr(S_1) &= \frac{1}{10} \\
\Pr(S_2) &= \frac{1}{10} \\
\Pr(S_3) &= \frac{7}{10}
\end{align*}
\]
Problem: Lack of exploration

Optimal policy

Random Policy starting from $s_0$

$r = +100$

$r = +100$

$\frac{1}{10}$ $\frac{1}{10}$ $\frac{1}{10}$ $\frac{7}{10}$

$\frac{6}{10}$ $\frac{3}{10}$ $\frac{1}{10}$ $\frac{0}{10}$
Problem: Lack of exploration

Optimal policy

After many rounds of policy iteration
Solution: Demand improvement from all states

Choosing an uniform start state distribution

After policy iteration
Key Idea: Use a good “restart” distribution

Choose a restart distribution $\mu(s)$ instead of start state distribution $\mu(s)$

Try your best to “cover” states the expert will visit

Suffer at most a penalty of $\| \frac{d_{\pi^*}}{\mu} \|_{\infty}$
Nightmare 2: Distribution Shift
Approximate Policy Iteration

Estimate advantage

\[ A^\pi(s, a) \]

Greedily improve policy

\[ \pi' = \arg \min_{\pi'} A^\pi(s, \pi'(s)) \]
The problem of distribution shift
The problem of distribution shift
The problem of distribution shift
The problem of distribution shift
How does distribution shift manifest?

The true performance difference

\[
V^{\pi'}(s) - V^{\pi}(s) = \frac{1}{1 - \gamma} \mathbb{E}_{s \sim d^{t}_{\pi}} A^{\pi}(s, \pi'(s))
\]

(New) (Old)

What our estimator currently approximates

\[
\frac{1}{1 - \gamma} \mathbb{E}_{s \sim d^{t}_{\pi}} A^{\pi}(s, \pi'(s))
\]
Be stable

Slowly change policies

Keep $d^t_\pi$ close to $d^t_{\pi'}$
Idea 1: Conservative Policy Iteration (CPI)

\[ \pi' = (1 - \alpha)\pi + \alpha\pi_{\text{greedy}} \]

Mix in old policy and greedy policy

Can prove that performance difference is bounded by

\[ V^{\pi'}(s) - V^{\pi}(s) \geq \alpha A_{\text{greedy}} - 2\alpha^2 \frac{\gamma}{1 - \gamma} \]

How much greedy policy improves based on estimate

How much distribution shift hurts!
Idea 2: Update distributions slowly

CPI requires keeping around all the policies you have seen thus far, which is not scalable ... 

Instead can we change policies slowly?

Does this simply mean do gradient descent with a small step size?
Nightmare 2: Distribution Shift
Correlated Features
Activity!
What happens if we have correlated features?

**Parameterization 1:** $f_1 = \# \text{ of Holes after the placement}, f_2 = \text{Height after the placement}$. We use $\theta$ to denote the parameter for this parameterization.

**Parameterization 2:** $g_1 = \ldots = g_{100} = \# \text{ of Holes after the placement}$, $g_{101} = \text{Height after the placement}$. We use $\phi$ to denote the parameter for this parameterization.

Then, for Parameterization 1, we have,

$$\theta^\top f(x,a) = \theta_1 \times \# \text{ of Holes}(x,a) + \theta_2 \times \text{Height}(x,a).$$

While for Parameterization 2, we have,

$$\phi^\top g = \left(\sum_{i=1}^{100} \phi_i\right) \times \# \text{ of Holes}(x,a) + \phi_{101} \times \text{Height}(x,a).$$
Think-Pair-Share

Think (30 sec): What would happen if we ran policy gradient with Feature Set 1 vs Feature Set 2? How can we fix it?

Pair: Find a partner

Then, for Parameterization 1, we have,

$$\theta^T f(x,a) = \theta_1 \times \text{# of Holes}(x,a) + \theta_2 \times \text{Height}(x,a).$$

While for Parameterization 2, we have,

$$\phi^T g = \left( \sum_{i=1}^{100} \phi_i \right) \times \text{# of Holes}(x,a) + \phi_{101} \times \text{Height}(x,a).$$

Share (45 sec): Partners exchange ideas
Gradient Descent as Steepest Descent

Gradient Descent is simply Steepest Descent with L2 norm

$$\max_{\Delta \theta} J(\theta + \Delta \theta) \quad \text{s.t.} \quad \|\Delta \theta\| \leq \epsilon$$

An alternative norm: KL Divergence! Gives rise to Fisher Information Matrix

$$G(\theta) = E_{p_\theta} \left[ \nabla_\theta \log(p_\theta) \nabla_\theta \log(p_\theta)^\top \right]$$

$$\Delta \theta = \frac{1}{2\lambda} \tilde{G}^{-1}(\theta) \tilde{\nabla}_\theta J.$$
Natural Gradient Descent

Estimate Fisher Information Matrix

\[ \tilde{G}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left[ \nabla_{\theta} \log \pi_\theta(a_i|s_i) \nabla_{\theta} \log \pi_\theta(a_i|s_i)^\top \right] \]

Parameter update:

\[ \Delta \theta = \frac{1}{2\lambda} \tilde{G}^{-1}(\theta) \tilde{\nabla}_\theta J. \]

Modern variants known as TRPO, PPO
Nightmare 3: Variance
What happens when Q values for all rollouts are similar?

\[ \nabla_{\theta} J = E_{p(\xi|\theta)} \left[ \sum_{t=0}^{T-1} \nabla_{\theta} \log \pi_{\theta}(a_t|s_t) Q^{\pi_{\theta}}(s_t, a_t) \right] \]

Recall that one of the reasons for the high variance is that the algorithm does not know how well the trajectories perform compared to other trajectories. Therefore, by introducing a baseline for the total reward (or reward to go), we can update the policy based on how well the policy performs compared to a baseline.
Solution: Subtract a baseline!

\[ \nabla_\theta J = E_{d^{\pi_\theta}(s)} E_{\pi_\theta(a|s)} \left[ \nabla_\theta \log(\pi_\theta(a|s)) \left( Q^{\pi_\theta}(s, a) - V^{\pi_\theta}(s) \right) \right]. \]

We can prove that this does not change the gradient

But turns Q values into advantage (which is lower magnitude)
The Policy Gradient Theorem

$$\nabla_\theta J = E_{p(\zeta|\theta)} \left[ \sum_{t=0}^{T-1} \left( \nabla_\theta \log \pi_\theta(a_t|s_t) \left( \sum_{r=0}^{T-1} r(s_r,a_r) + \sum_{r=t}^{T-1} r(s_r,a_r) \right) \right) \right]$$

$$= E_{p(\zeta|\theta)} \left[ \sum_{t=0}^{T-1} \left( \nabla_\theta \log \pi_\theta(a_t|s_t) \sum_{r=t}^{T-1} r(s_r,a_r) \right) \right],$$

$$\nabla_\theta J = E_{p(\zeta|\theta)} \left[ \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) Q^\pi_\theta(s_t,a_t) \right]$$

1. Local Optima: Use Exploration Distribution
2. Distribution Shift: *Natural* Gradient Descent
3. High Variance: Subtract baseline