Safer Policy Search

CompSci 590.2
Ron Parr
Duke University

Content from Kakade & Langford ICML 02, and Schulman et al. ICML 15

Issues addressed by Kakade & Langford

• Hard to guarantee monotonic improvement with existing approximate RL methods

• On-policy nature of policy gradient methods can lead to arbitrarily slow learning
Exploration Issue

- Figure from Kakade & Langford:

  ![Diagram](image)

  - Suppose initial policy has uniform distribution over 3 actions
  - Expected time to reach rightmost state from leftmost is exponential in n

Solution: Different distributions

- Optimization distribution may be part of the problem, e.g., suppose problem starts in leftmost state; we are forced to optimize for this case
- Create a “restart” distribution for training, e.g., pick states closer to goal for training
- Issues:
  - How do you pick this for big problems?
  - Can improve exploration at the expense of weaker bounds
- Not the focus of this lecture
Notation (from Schulman et al.)

\[ \eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t) \right], \text{ where} \]
\[ s_0 \sim \rho_0(s_0), \ a_t \sim \pi(a_t|s_t), \ s_{t+1} \sim P(s_{t+1}|s_t, a_t) \]
\[ Q_\pi(s_t, a_t) = \mathbb{E}_{s_{t+1}, a_{t+1}, \ldots} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_{t+t}) \right], \]
\[ V_\pi(s_t) = \mathbb{E}_{a_t, s_{t+1}, \ldots} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_{t+t}) \right], \]
\[ A_\pi(s, a) = Q_\pi(s, a) - V_\pi(s), \text{ where} \]
\[ a_t \sim \pi(a_t|s_t), s_{t+1} \sim P(s_{t+1}|s_t, a_t) \text{ for } t \geq 0. \]

Express value of one policy in terms of Advantages over another

- In terms of expectations:
  \[ \eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s_0, a_0, \ldots} \tilde{\pi} \left[ \sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t) \right] \]

- In terms of state distribution:
  \[ = \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a). \]
Issues

- For a stochastic policy function, advantages may not be uniformly positive – picking better actions at some states could cause us to pick worse actions at others.

- How do we know if it’s safe to adopt the greedy policy?

\[
\eta(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a)
\]

We don’t know this until we try the new policy

Workarounds

- Approximate improvement using current state distribution:

\[
L_\pi(\pi) = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a)
\]

- This is exact when argument and subscript match:

\[
L_{\pi_0}\theta \left(\pi_0\right) = \eta(\pi_0),
\]

\[
\nabla_\theta L_{\pi_0}\theta \left(\pi_0\right)\bigg|_{\theta=\theta_0} = \nabla_\theta \eta(\pi_0)\bigg|_{\theta=\theta_0}
\]

- Optimizing \( L \) near \( \pi_0 \) also optimizes \( \eta \) for small step sizes.
How to make small changes

\[ \pi_{\text{new}}(a|s) = (1 - \alpha)\pi_{\text{old}}(a|s) + \alpha\pi'(a|s) \]

- How to pick \( \alpha \)

\[
\eta(\pi_{\text{new}}) \geq L_{\pi_{\text{old}}}(\pi_{\text{new}}) - \frac{2\epsilon\gamma}{(1 - \gamma)^2} \alpha^2
\]

where \( \epsilon = \max_s |E_{a\sim\pi'(a|s)} [A_\pi(s, a)]| \)

- Can solve for \( \alpha \) to maximize policy improvement bound

Turning this into an algorithm

- Repeat
  - Estimate advantages using rollouts and regression
  - Make a conservative policy update step

- Kakade & Langford provide sample complexity bounds for this under the assumption of bounded error in the advantage estimation
- Not necessarily a practical algorithm – no experimental results
Kakade & Langford conclusions

- Clever approach to addressing the step size challenge in policy gradient methods form theoretical standpoint
- Not quite practical

Trust Region Policy Optimization (TRPO)

- Generalize Kakade & Langford results from a mixture between policies to KL divergence between resulting distributions over actions:

\[ \eta(\tilde{\pi}) \geq L_\mu(\tilde{\pi}) - CD_{KL}^{\max}(\pi, \tilde{\pi}), \]

where \( C = \frac{4\epsilon \gamma}{(1 - \gamma)^2} \).

- Why is this useful?
- Action distributions are really the thing that matters –compare with natural policy gradient
- Potential for larger steps in policy space
Revised algorithm (Schulman et al.)

**Algorithm 1** Policy iteration algorithm guaranteeing non-decreasing expected return \( \eta \)

Initialize \( \pi_0 \).

for \( i = 0, 1, 2, \ldots \) until convergence do

   Compute all advantage values \( A_{\pi_i}(s, a) \).

   Solve the constrained optimization problem

   \[
   \pi_{i+1} = \arg \max_{\pi} \left[ L_{\pi_i}(\pi) - C \mathcal{D}_{KL}^{\max}(\pi, \pi) \right]
   \]

   where \( C = 4\epsilon\gamma/(1-\gamma)^2 \)

   and \( L_{\pi_i}(\pi) = \eta(\pi_i) + \sum_s \rho_{\pi}(s) \sum_a \pi(a|s)A_{\pi_i}(s, a) \)

end for

Issues

- How to do the optimization?
- How to estimate the advantages, etc.
- \( \mathcal{D}^{\text{MAX}}_{\text{KL}} \) not practical to compute because it is a max over all states
- Step sizes still small
- Still not practical
Making it practical

- Use ordinary (weighted average) $D_{KL}$ instead of max
- Replace optimization with hard constraint on $max D_{KL}$ between old and new policies to avoid decreasing performance (still obtained from bound), allowing bigger steps

$$\max_{\theta} \sum_{s} \rho_{\theta_{old}}(s) \sum_{a} \pi_{\theta}(a|s) A_{\theta_{old}}(s, a)$$

subject to $D_{KL}^{\rho_{\theta_{old}}}(\theta_{old}, \theta) \leq \delta$.

Remaining Issues

$$\max_{\theta} \sum_{s} \rho_{\theta_{old}}(s) \sum_{a} \pi_{\theta}(a|s) A_{\theta_{old}}(s, a)$$

subject to $D_{KL}^{\rho_{\theta_{old}}}(\theta_{old}, \theta) \leq \delta$.

- Not practical to evaluate this over entire state space
- Would like to use trajectories to approximate this
- Would like to avoid summing over all actions:
  - Sample actions from some distribution $q$
  - Use importance weights
  - Use $Q$ values instead of advantages (shifts objective by a constant)
With sampling, an importance weights

$$
\begin{align*}
&\text{maximize } \sum_s \rho_{\theta_{\text{old}}} (s) \sum_a \pi_{\theta}(a|s) A_{\theta_{\text{old}}} (s,a) \\
&\quad \text{subject to } D_{\text{KL}}^{\rho_{\theta_{\text{old}}} \ast} (\theta_{\text{old}}, \theta) \leq \delta.
\end{align*}
$$

$$
\begin{align*}
&\text{maximize } \mathbb{E}_{s \sim \rho_{\theta_{\text{old}}}, a \sim q} \left[ \frac{\pi_{\theta}(a|s)}{q(a|s)} Q_{\theta_{\text{old}}} (s,a) \right] \\
&\quad \text{subject to } \mathbb{E}_{s \sim \rho_{\theta_{\text{old}}}} [D_{\text{KL}} (\pi_{\theta_{\text{old}}} (\cdot|s) \parallel \pi_{\theta} (\cdot|s))] \leq \delta.
\end{align*}
$$

Two approaches to rollouts (Figure from Schulman et al.)
TRPO in practice

• No guarantees because of various approximations used
• Repeat:
  • For a given set of policy parameters
  • Collect trajectories
  • Form optimization problem
  • Approximately solve optimization problem using conjugate gradient and line search

• Works very well for control problems
• Not obviously better than DQN for Atari games but not bad

Conclusions

• Historically, policy gradient methods have been limited by high variance and small step sizes, making them slow/unreliable
• Accumulation of insights over past ~15 years have provided a set of tools for reducing variance and increasing step sizes

• Policy gradient methods generally perceived to be best way to go for control problems