Trust Region Policy Optimization & The Natural Policy Gradient

Lucas Janson and Sham Kakade

CS/Stat 184: Introduction to Reinforcement Learning

Fall 2023

Today

- Recap
- Algorithms:
  - Trust Region Policy Optimization (TRPO)
  - The Natural Policy Gradient (NPG)
  - Proximal Policy Optimization (PPO)

Recap

(M=1) PG with a Learned Baseline:

1. Initialize θ_0, parameters: $\eta_1, \eta_2, \ldots$
2. For k = 0, … :
   1. Sup. Learning: Using N trajectories sampled under π_{θ^k}, estimate a baseline $\widetilde{b}_h$
      $\widetilde{b}(s) \approx V_h^{\theta^k}(s)$
   2. Obtain a trajectory $\tau \sim \rho_{\theta^k}$
      Set $\widetilde{\nabla}_\theta J(\theta^k) = \sum_{h=0}^{H-1} \nabla \ln \pi_{\theta^k}(a_h | s_h)\left(R_h(\tau) - \widetilde{b}(s_h)\right)$
   3. Update: $\theta^{k+1} = \theta^k + \eta^k \widetilde{\nabla}_\theta J(\theta^k)$

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

The Performance Difference Lemma (PDL)

- Let $\rho_{\widetilde{\pi},s}$ be the distribution of trajectories from starting state s acting under π. (we are making the starting distribution explicit now).
- For any two policies π and $\widetilde{\pi}$ and any state s,

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

Comments:

- Helps us think about error analysis, instabilities of fitted PI, and sub-optimality.
- Helps to understand algorithm design (TRPO, NPG, PPO)
- This also motivates the use of “local” methods (e.g. policy gradient descent)

Back to Approximate Policy Iteration (API)

- Suppose π^k gets updated to π^{k+1}. How much worse could π^{k+1} be?
- Suppose at some state s, π^{k+1} choose an action which has a negative advantage for π^k.
  - Since $\widetilde{A}^k(s, a, h) \approx A_h^{\pi^k}(s, a, h)$, we expect some error.
  - In the worst case, let us consider the most negative advantage:

$$\Delta_\infty := \min_{s \in S} \; A_h^{\pi^k}(s, \pi^{k+1}(s))$$

  - Here, if $\Delta_\infty < 0$, it is possible that degradation may occur:

$$V^{\pi^{k+1}}(s_0) \geq V^{\pi^k}(s_0) - H \cdot |\Delta_\infty|$$

Proof sketch:

- Fitted PI does not enforce that the trajectory distributions, ρ_{π^k} and $\rho_{\pi^{k+1}}$, be close to each other.
- Suppose the $\rho_{\pi^{k+1}}$ has full support on these worst case states s (i.e. we get trapped at this state where we made a bad choice).

Trust Region Policy Optimization (TRPO)

1. Init π_0
2. For $k = 0, \ldots K$:

$$\theta^{k+1} = \arg\max_{\theta} \mathbb{E}_{s_0,\ldots s_{H-1} \sim \rho_{\pi^k}} \left[\sum_{h=0}^{H-1} \mathbb{E}_{a_h \sim \pi_\theta(s_h)} A^{\pi^k}(s_h, a_h) \right]$$

$$\text{s.t. } KL\left(\rho_{\pi^k} | \rho_{\pi_\theta}\right) \leq \delta$$

3. Return π_K

- We want to maximize local advantage against π_{θ^k}, but we want the new policy to be close to π_{θ^k} (in the KL sense)
- **How do we implement this with sampled trajectories?**

KL-divergence: measures the distance between two distributions

Given two distributions P & Q, where $P \in \Delta(X), Q \in \Delta(X)$,
KL Divergence is defined as:

$$KL(P|Q) = \mathbb{E}_{x\sim P}\left[\ln \frac{P(x)}{Q(x)}\right]$$

Examples:

If $Q = P$, then $KL(P|Q) = KL(Q|P) = 0$

If $P = \mathcal{N}(\mu_1, \sigma^2 I), Q = \mathcal{N}(\mu_2, \sigma^2 I)$, then $KL(P|Q) = \frac{1}{2\sigma^2}\|\mu_1 - \mu_2\|^2$

Fact:

$KL(P|Q) \geq 0$, and being 0 if and only if $P = Q$

Estimating TRPO: optional slide

(see PPO & Importance sampling for derivation)

1. Initialize staring policy π_0, samples size M
2. For $k = 0, \ldots K$:
    1. Using N trajectories sampled under ρ^k to learn a $\widetilde{b}_h$
       $\widetilde{b}(s,h) \approx V_h^{\pi^k}(s)$
    2. Obtain M NEW trajectories $\tau_1, \ldots \tau_M \sim \rho^k$
       Solve the following optimization problem to obtain π_{k+1}:

$$\max_{\theta} \frac{1}{M} \sum_{m=1}^{M} \sum_{h=0}^{H-1} \frac{\pi_\theta(s_h)}{\pi^k(s_h)} \left(R_h(\tau^m) - \widetilde{b}(s_h, h) \right)$$

$$\text{s.t.} \sum_{m=1}^{M} \sum_{h=0}^{H-1} \ln \frac{\pi_{\theta_k}(a_h^m \mid s_h^m)}{\pi_\theta(a_h^m \mid s_h^m)} \le \delta$$

Today:

Today

- Recap
- Algorithms:
  - Trust Region Policy Optimization (TRPO)
  - The Natural Policy Gradient (NPG)
  - Proximal Policy Optimization (PPO)

TRPO is locally equivalent to the NPG

TRPO at iteration k:

$$\max_{\theta} \mathbb{E}_{s_0,\ldots s_{H-1}\sim\rho_{\pi^k}} \left[\sum_{h=0}^{H-1} \mathbb{E}_{a_h\sim\pi_\theta(s_h)} A^{\pi^k}(s_h, a_h) \right]$$

→ First-order Taylor expansion at θ^k

$$\text{s.t. } KL\left(\rho_{\pi^k} | \rho_{\pi_\theta}\right) \leq \delta$$

→ second-order Taylor expansion at θ^k

Intuition: maximize local adv subject to being incremental (in KL);

$$\max_{\theta} \nabla_\theta J(\pi_{\theta^k})^\top (\theta - \theta^k)$$

$$\text{s.t. } (\theta - \theta^k)^\top F_{\theta^k}(\theta - \theta^k) \leq \delta$$

(Where F_{θ^k} is the “Fisher Information Matrix”)

NPG: A “leading order” equivalent program to TRPO:

1. Init π_0
2. For $k = 0, \ldots K$:

$$\theta^{k+1} = \arg\max_{\theta} \nabla_\theta J(\pi_{\theta^k})^\top (\theta - \theta^k)$$

s.t. $(\theta - \theta^k)^\top F_{\theta^k} (\theta - \theta^k) \leq \delta$

3. Return π_K

- Where $\nabla_\theta J(\pi_{\theta^k})$ is the gradient at θ^k and
- F_θ is (basically) the Fisher information matrix at $\theta \in \mathbb{R}^d$, defined as:

$$F_\theta := \mathbb{E}_{\tau \sim \rho_\theta} \left[\nabla_\theta \ln \rho_\theta(\tau) \left(\nabla_\theta \ln \rho_\theta(\tau) \right)^\top \right] \in \mathbb{R}^{d \times d}$$

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

There is a closed form update:

Linear objective and quadratic convex constraint, we can solve it optimally!

Indeed this gives us:

$$\theta^{k+1} = \theta^k + \eta F_{\theta^k}^{-1} \nabla_\theta J(\pi_{\theta^k})$$

Where $\eta = \sqrt{\dfrac{\delta}{\nabla_\theta J(\pi_{\theta^k})^\top F_{\theta^k}^{-1} \nabla_\theta J(\pi_{\theta^k})}}$

An Implementation: Sample Based NPG

1. Init π_0
2. For $k = 0, \ldots K$:
   - Estimate PG $\nabla_\theta J(\pi_{\theta^k})$
   - Estimate Fisher info-matrix: $F_{\theta^k} = \mathbb{E}_{\tau \sim \rho_{\theta^k}} \left[\sum_{h=0}^{H-1} \nabla \ln \pi_{\theta^k}(a_h | s_h) \left(\nabla \ln \pi_{\theta^k}(a_h | s_h) \right)^\top \right]$
   - Natural Gradient Ascent: $\theta^{k+1} = \theta^k + \eta \, \widehat{F_{\theta^k}}^{-1} \widehat{\nabla_\theta J(\pi_{\theta^k})}$
3. Return π_K

(We will implement it in HW4 on Cartpole)

NPG Derivation

First Order Expansion on the Objective Function

$$\max_{\theta} \mathbb{E}_{s_0,\ldots s_{H-1}\sim\rho_{\theta_k}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a\sim\pi_{\theta}(s)}A^{\pi_{\theta_k}}(s,a)\right]$$

Let's look at a first order Taylor expansion around $\theta = \theta^k$:

$$\mathbb{E}_{s_0,\ldots s_{H-1}\sim\rho_{\theta_k}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a\sim\pi_{\theta}(s)}A^{\pi_{\theta_k}}(s,a)\right] \approx \mathbb{E}_{s_0,\ldots s_{H-1}\sim\rho_{\theta_k}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a\sim\pi_{\theta_k}(s)}A^{\pi_{\theta_k}}(s,a)\right]$$

$$+\underbrace{\mathbb{E}_{s_0,\ldots s_{H-1}\sim\rho_{\theta_k}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a\sim\pi_{\theta_k}(s)}\nabla_{\theta}\ln\pi_{\theta_k}(a\,|\,s)A^{\pi_{\theta_k}}(s,a)\right]}_{\nabla_{\theta}J(\pi_{\theta_k})}\cdot(\theta-\theta_k)$$

$$= \text{"constant"} + \nabla_{\theta}J(\pi_{\theta_k})^{\top}(\theta-\theta_k)$$

Taylor Expansion on the Constraint

(we need it to be second-order. Why?)

$$\ell(\theta) := KL(\rho_{\widetilde{\theta}} | \rho_{\theta})$$

$$\ell(\theta) \approx \ell(\widetilde{\theta}) + \nabla \ell(\widetilde{\theta})^{\top}(\theta - \widetilde{\theta}) + \frac{1}{2}(\theta - \widetilde{\theta})^{\top} \nabla_{\theta}^{2} \ell(\widetilde{\theta})(\theta - \widetilde{\theta})$$

$$\ell(\widetilde{\theta}) = KL(\rho_{\widetilde{\theta}} | \rho_{\widetilde{\theta}}) = 0$$

We will show that $\nabla_{\theta} \ell(\widetilde{\theta}) = 0$, and $\nabla^{2} \ell(\widetilde{\theta})$ has the claimed form!

The gradient of the KL-divergence is zero at θ^k

Change from trajectory distribution to state-action distribution:

$$\ell(\theta) := KL\left(\rho_{\tilde{\theta}} | \rho_\theta\right) = \mathbb{E}_{\tau\sim\rho_{\tilde{\theta}}}\left[\ln \frac{\rho_{\tilde{\theta}}(\tau)}{\rho_\theta(\tau)}\right]$$

$$\nabla_\theta \ell(\theta)\Big|_{\theta=\tilde{\theta}} = -\mathbb{E}_{\tau\sim\rho_{\tilde{\theta}}}\left[\nabla_\theta \ln \rho_{\tilde{\theta}}(\tau)\right]$$

$$= -\sum_\tau \rho_{\tilde{\theta}}(\tau)\frac{\nabla_\theta \rho_{\tilde{\theta}}(\tau)}{\rho_{\tilde{\theta}}(\tau)}$$

$$= 0$$

Let's compute the Hessian of the KL-divergence at θ^k

$$\ell(\theta) := KL\left(\rho_{\pi_{\theta^k}} | \rho_{\pi_\theta}\right) = \mathbb{E}_{\tau \sim \rho_{\pi_{\theta^k}}}\left[\ln \frac{\rho_{\pi_{\theta^k}}(\tau)}{\rho_{\pi_\theta}(\tau)}\right]$$

$$\nabla^2_\theta \ell(\theta)\Big|_{\theta=\widetilde{\theta}} = -\mathbb{E}_{\tau \sim \rho_{\widetilde{\theta}}}\left[\nabla^2_\theta \ln \rho_{\widetilde{\theta}}(\tau)\right]$$

$$= -\sum_\tau \rho_{\widetilde{\theta}}(\tau)\left(\frac{\nabla^2_\theta \rho_{\widetilde{\theta}}(\tau)}{\rho_{\widetilde{\theta}}(\tau)} - \frac{\nabla_\theta \rho_{\widetilde{\theta}}(\tau)\, \nabla_\theta \rho_{\widetilde{\theta}}(\tau)^\top}{\left(\rho_{\widetilde{\theta}}(\tau)\right)^2}\right)$$

$$= \mathbb{E}_{\tau \sim \rho_{\widetilde{\theta}}}\left[\nabla \ln \rho_{\widetilde{\theta}}(\tau)\left(\nabla_\theta \ln \rho_{\widetilde{\theta}}(\tau)\right)^\top\right] \in \mathbb{R}^{d \times d}$$

It's called the Fisher Information Matrix!

Today

- Recap
- Algorithms:
    - Trust Region Policy Optimization (TRPO)
    - The Natural Policy Gradient (NPG)
    - Proximal Policy Optimization (PPO)

Back to TRPO/NPG

1. Init π_0
2. For $k = 0, \ldots K$:

$$\theta^{k+1} = \arg\max_{\theta} \mathbb{E}_{s_0,\ldots s_{H-1} \sim \rho_{\pi^k}} \left[\sum_{h=0}^{H-1} \mathbb{E}_{a_h \sim \pi_\theta(s_h)} A^{\pi^k}(s_h, a_h) \right]$$

$$\text{s.t. } KL\left(\rho_{\pi^k} | \rho_{\pi_\theta}\right) \leq \delta$$

3. Return π_K

- The difficulty with TRPO and NPG is that they could be computationally costly. Need to solve constrained optimization or matrix inversion ("second order") problems.
- Can we use a method which only uses gradients?

Let's try to use a "Lagrangian relaxation" of TRPO

Proximal Policy Optimization (PPO)

1. Init π_0, choose λ
2. For $k = 0, \ldots K$:

$$\theta^{k+1} = \arg\max_{\theta} \mathbb{E}_{s_0, \ldots s_{H-1} \sim \rho_{\pi^k}} \left[\sum_{h=0}^{H-1} \mathbb{E}_{a_h \sim \pi_\theta(s_h)} A^{\pi^k}(s_h, a_h) \right] \underbrace{- \lambda KL \left(\rho_{\pi^k} | \rho_{\pi_\theta} \right)}_{\text{regularization}}$$

3. Return π_K

The regularization term is:

$$KL\left(\rho_{\pi_{\theta^k}} | \rho_{\pi_\theta}\right) = \mathbb{E}_{\tau \sim \rho_{\pi_{\theta^k}}}\left[\ln \frac{\rho_{\pi_{\theta^k}}(\tau)}{\rho_{\pi_\theta}(\tau)}\right]$$

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

$$= \mathbb{E}_{\tau \sim \rho_{\pi_{\theta^k}}}\left[\sum_{h=0}^{H-1} \ln \frac{1}{\pi_\theta(a_h | s_h)}\right] + \left[\text{term not a function of } \theta\right]$$

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

Proximal Policy Optimization (PPO)

1. Init π_0, choose λ
2. For $k = 0, \ldots K$:
   use SGD to optimize:
   $$\theta^{k+1} \approx \arg\max_{\theta} \ell^k(\theta)$$
   where:
   $$\ell^k(\theta) := \mathbb{E}_{s_0,\ldots s_{H-1} \sim \rho_{\pi^k}} \left[\sum_{h=0}^{H-1} \mathbb{E}_{a_h \sim \pi_\theta(s_h)} A^{\pi^k}(s_h, a_h) \right] - \lambda \mathbb{E}_{\tau \sim \rho_{\pi^k}} \left[\sum_{h=0}^{H-1} \ln \frac{1}{\pi_\theta(a_h \mid s_h)} \right]$$
3. Return π_K

How do we estimate this objective?

Back to Estimating $\ell^k(\theta)$

We want to estimate,

$$\mathbb{E}_{s_0,\ldots s_{H-1}\sim\rho_{\pi^k}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi_{\theta}(s_h)}A^{\pi^k}(s_h,a_h)\right]$$

We will use importance sampling:

$$= \mathbb{E}_{s_0,\ldots s_{H-1}\sim\rho_{\pi^k}}\left[\sum_{h=0}^{H-1}\mathbb{E}_{a_h\sim\pi^k(s_h)}\left[\frac{\pi_{\theta}(s_h)}{\pi^k(s_h)}A^{\pi^k}(s_h,a_h)\right]\right]$$

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

Estimating $\ell^k(\theta)$

1. Using N trajectories sampled under ρ^k to learn a $\widetilde{b}_h$

   $\widetilde{b}(s,h) \approx V_h^{\pi^k}(s)$

2. Obtain M NEW trajectories $\tau_1, \ldots \tau_M \sim \rho^k$

   Set $\widehat{\ell}^k(\theta) = \frac{1}{M}\sum_{m=1}^{M}\sum_{h=0}^{H-1}\left(\frac{\pi_\theta(s_h)}{\pi^k(s_h)}\left(R_h(\tau^m) - \widetilde{b}(s_h,h)\right) - \lambda \ln \frac{1}{\pi_\theta(a_h \mid s_h)}\right)$

Summary:

1. NPG: a simpler way to do TRPO, a “pre-conditioned” gradient method.
2. PPO: “first order” approx to TRPO

Attendance:

bit.ly/3RcTC9T

Feedback:

bit.ly/3RHtlxy