Trust Region Policy Optimization

Tri Nguyen

Summer Reading 2021 - Oregon State University

September 3, 2021

Roadmap

- The method belong to policy gradient class, where policy is parametrizied
- Use another objective function (avoid using the trick of Policy Gradient)
- Propose an optimization method to solve that objective function *approximately*
- Experimental result

The proposed method is pleasingly complicated :)))

Trust Region Policy Optimization

John Schulman JOSCHU@EECS.BERKELEY.EDU
Sergey Levine SLEVINE@EECS.BERKELEY.EDU
Philipp Moritz PCMORITZ@EECS.BERKELEY.EDU
Michael Jordan JORDAN@CS.BERKELEY.EDU
Pieter Abbeel PABBEEL@CS.BERKELEY.EDU
University of California, Berkeley, Department of Electrical Engineering and Computer Sciences

- Kakade, Sham M. "A natural policy gradient." Advances in neural information processing systems 14 (2001). (910 citations)
- **Schulman, John, et al. "Trust region policy optimization." International conference on machine learning. PMLR, 2015. (3826 citations)**
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347. (5066 citaions)

Notation

- $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the transition probability distribution.
- $r : \mathcal{S} \to \mathbb{R}$ is the reward function.
- $\rho_0 : \mathcal{S} \to \mathbb{R}$ is the distribution of the initial state s_0.
- $\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots} \left[\sum_{t=1}^{\infty} \gamma^t r(s_t) \right]$ is the expected discounted reward, where

$$s_0 \sim \rho_0(s_0), a_t \sim \pi(a_t | s_t), s_{t+1} \sim P(s_{t+1} | s_t, a_t).$$

- $Q_\pi(s_t, a_t) = \mathbb{E}_{s_{t+1}, a_{t+1}, \ldots} \left[\sum_{\ell=0}^{\infty} \gamma^\ell r(s_{t+1}) \right]$
- $V_\pi(s_t) = \mathbb{E}_{a_t, s_{t+1}, \ldots} \left[\sum_{\ell=0}^{\infty} \gamma^\ell r(s_{t+1}) \right]$
- $A_\pi(s, a) = Q_\pi(s, a) - V_\pi(s)$ is the advantage function.

Starting Point

- Kakade & Langford (2002) showed a relation between any policy π and $\widetilde{\pi}$.

$$\eta(\widetilde{\pi}) = \eta(\pi) + \mathbb{E}_{s_0, a_0, \ldots \sim \widetilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t) \right] \tag{1}$$

Proof.

Let $\tau \sim \widetilde{\pi}$ be a trajectory sampled using $\widetilde{\pi}$.

$$\mathbb{E}_{\tau|\widetilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t) \right]$$

$$= \mathbb{E}_{\tau|\widetilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t (r(s_t) + \gamma V_\pi(s_{t+1}) - V_\pi(s_t)) \right]$$

$$= \mathbb{E}_{\tau|\widetilde{\pi}} \left[r(s_0) + \gamma V_\pi(s_1) - V_\pi(s_0) + \gamma (r(s_1) + \gamma V_\pi(s_2) - V_\pi(s_1)) + \gamma^2(.) + \ldots \right]$$

$$= \mathbb{E}_{\tau|\widetilde{\pi}} \left[-V_\pi(s_0) + \sum_{t=0}^{\infty} \gamma^t r(s_t) \right] = -\mathbb{E}_{s_0}\left[V_\pi(s_0)\right] + \mathbb{E}_{\tau|\widetilde{\pi}} \gamma^t r(s_t)$$

$$= -\eta_\pi + \eta_{\widetilde{\pi}}$$

- Define ρ_π is the discounted visitation frequencies, $\rho_\pi = \sum_{t=0}^{\infty} \gamma^t P(s_t = s)$. Note that $s_0 \sim \rho_0$, the others depend on π and the environment.
- Rewrite (1)

$$\begin{aligned}
\eta(\widetilde{\pi}) &= \eta(\pi) + \mathbb{E}_{s_0, a_0, \ldots \sim \widetilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t) \right] \\
&= \eta(\pi) + \sum_{t=0}^{\infty} \sum_{s} P(s_t = s | \widetilde{\pi}) \sum_{a} \widetilde{\pi}(a|s) \gamma^t A_\pi(s_t, a_t) \\
&= \eta(\pi) + \sum_{s} \sum_{t=0}^{\infty} \gamma^t P(s_t = s | \widetilde{\pi}) \sum_{a} \widetilde{\pi}(a|s) A_\pi(s_t, a_t) \\
&= \eta(\pi) + \sum_{s} \rho_{\widetilde{\pi}(s)} \color{blue}{\sum_{a} \widetilde{\pi}(a|s) A_\pi(s_t, a_t)}
\end{aligned}$$

- If the blue term is nonnegative at every state, then $\widetilde{\pi}$ is better or equal π
- In deterministics setting, it reduces to policy improvement, i.e., $\widetilde{\pi}(s) = \arg\max_a A_\pi(s, a)$
- Maximizing the RHS respect to paremeters of $\widetilde{\pi}$ would result the best policy

The first approximation

- Recall
$$\eta(\widetilde{\pi}) = \eta(\pi) + \sum_s \rho_{\widetilde{\pi}}(s) \sum_a \widetilde{\pi}(a|s) A_\pi(s_t, a_t)$$
- Define
$$L_\pi(\widetilde{\pi}) := \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \widetilde{\pi}(a|s) A_\pi(s_t, a_t)$$
then $L_\pi(\widetilde{\pi}) \approx \eta(\widetilde{\pi})$ locally in a sense that
$$L_{\pi_{\theta_0}}(\pi_{\theta_0}) = \eta(\pi_{\theta_0}), \quad \text{and } \nabla_\theta L_{\pi_{\theta_0}}(\pi_\theta)\mid_{\theta=\theta_0} = \nabla_\theta \eta(\pi_\theta)\mid_{\theta=\theta_0}$$
- The first equality holds since $\underbrace{\sum_s \rho_\pi(s) \sum_a \pi(a|s)}_{\mathbb{E}}(Q_\pi(a,s) - V_\pi(s)) = 0$
- An improvement is guaranteed if using the following updating rule
$$\pi_{\text{new}}(a|s) = (1-\alpha)\pi_{\text{old}}(a|s) + \alpha\pi'(a|s),$$
where $\pi' = \arg\max_\pi L_{\pi_{\text{old}}}(\pi)$ and it is bounded by
$$\eta(\pi_{\text{new}}) \geq L_{\pi_{\text{old}}}(\pi_{\text{new}}) - \frac{2\epsilon\gamma}{(1-\gamma)^2}\alpha^2$$
- Maximizing $L_{\pi_{\text{old}}}(\pi_\theta)$ respect to θ is guaranteed to improve over π_{old}

New lower bound

- Define $D_{\mathrm{TV}}(p||q) = \frac{1}{2} \sum_{i=1} |p_i - q_i|$ (called total variation divergence), and
- $D^{\mathrm{max}}_{\mathrm{TV}}(\pi, \widetilde{\pi}) = \max_s D_{\mathrm{TV}}(\pi(a|s), \widetilde{\pi}(a|s))$

Theorem

Let $\alpha = D^{\max}_{\mathrm{TV}}(\pi_{\mathrm{old}}, \pi_{\mathrm{new}})$. *Then the following bound holds*

$$\eta(\pi_{\mathrm{new}}) \geq L_{\pi_{\mathrm{old}}}(\pi_{\mathrm{new}}) - \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha^2,$$

where $\epsilon = \max_{a,s} |A_\pi(s, a)|$

The improvement is guaranteed to general stochastic policy.

Algorithm

- Define $D_{\mathrm{KL}}^{\max}(\pi, \widetilde{\pi}) = \max_s D_{\mathrm{KL}}(\pi(a|s), \widetilde{\pi}(a|s))$
- Since $D_{\mathrm{TV}}(p||q)^2 \leq D_{\mathrm{KL}}(p||q)$

$$\eta(\pi_{\mathrm{new}}) \geq L_{\pi_{\mathrm{old}}}(\pi_{\mathrm{new}}) - \frac{4\epsilon\gamma}{(1-\gamma)^2} D_{\mathrm{KL}}^{\max}(\pi_{\mathrm{old}}, \pi_{\mathrm{new}}) \tag{2}$$

Algorithm 1 Policy iteration algorithm guaranteeing non-decreasing expected return η

Initialize π_0.
for $i = 0, 1, 2, \dots$ until convergence **do**
  Compute all advantage values $A_{\pi_i}(s, a)$.
  Solve the constrained optimization problem

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

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

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

end for

Policy improvement guarantee:

- Let $M_i(\pi) = L_{\pi_i}(\pi) - C D_{\mathrm{KL}}^{\max}(\pi_i, \pi)$

$$\begin{aligned} \eta(\pi_{i+1}) &\geq M_i(\pi_{i+1}) \quad \text{(by 2)} \\ &\geq M_i(\pi_i) \quad \text{(by updating rule)} \\ &= \eta(\pi_i) \end{aligned}$$

- The algorithm design is a type of minorization-minimization, where M_i is the surrogate function
- It is slow if C large
- Optimization problem:

$$\max_{\theta} L_{\theta_{old}}(\theta)$$
$$\text{subject to } D_{\mathrm{KL}}^{\max}(\theta_{\mathrm{old}}, \theta) \leq \delta$$

The second approximation

Let $\overline{D}_{\mathrm{KL}}^{\rho} = \mathbb{E}_{s\sim\rho}[D_{\mathrm{KL}}](\pi_{\theta_1}(\cdot|s)|\pi_{\theta_2}(\cdot|s))$

- Relax constrain to $\overline{D}_{\mathrm{KL}}^{\rho_{\mathrm{old}}}(\theta_{\mathrm{old}}, \theta) \leq \delta$
- Rewrite the objective function in expectation form

$$
\begin{aligned}
&\arg\max_{\theta} \sum_s \rho_{\theta_{\mathrm{old}}}(s) \sum_a \pi_\theta(a|s) A_{\theta_{\mathrm{old}}}(s,a) \\
&= \arg\max_{\theta} \left(\sum_s \rho_{\theta_{\mathrm{old}}}(s) \sum_a \pi_\theta(a|s) Q_{\theta_{\mathrm{old}}}(s,a) - \sum_s \rho_{\theta_{\mathrm{old}}(s)} \sum_a \pi_\theta(a|s) V_{\theta_{\mathrm{old}}}(s) \right) \\
&= \arg\max_{\theta} \sum_s \rho_{\theta_{\mathrm{old}}}(s) \mathbb{E}_{a\sim q} \frac{\pi_\theta(a|s)}{q(a|s)} Q_{\theta_{\mathrm{old}}} \\
&= \arg\max_{\theta} \ \mathbb{E}_{s\sim\rho_{\mathrm{old}}, a\sim q} \frac{\pi_\theta(a|s)}{q(a|s)} Q_{\theta_{\mathrm{old}}}
\end{aligned}
$$

- Final optimization problem

$$
\min_{\theta} \ \mathbb{E}_{s\sim\rho_{\mathrm{old}}, a\sim q} \left[\frac{\pi_\theta(a|s)}{q(a|s)} Q_{\theta_{\mathrm{old}}} \right]
$$

$$
\text{subject to } \mathbb{E}_{s\sim\rho_{\theta_{\mathrm{old}}}}[D_{\mathrm{KL}}(\pi_{\mathrm{old}}(\cdot|s)||\pi_{\mathrm{new}}(\cdot|s))] \leq \delta
$$

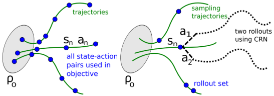
where q is behaviour policy, $q = \pi_{\theta_{\mathrm{old}}}$

Pratical algorithm - The third approximation

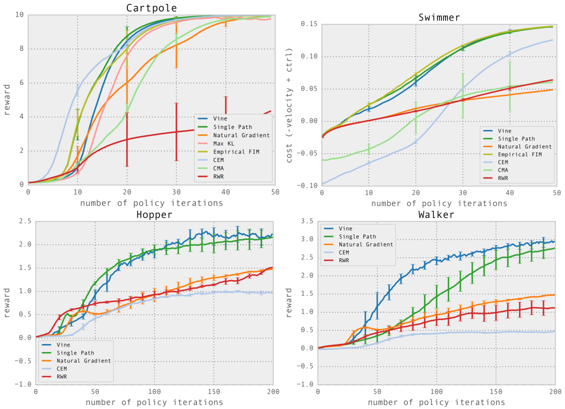
$$
\begin{aligned}
&\min_{\theta} \; \mathbb{E}_{s\sim\rho_{\text{old}}, a\sim q}\left[\frac{\pi_\theta(a|s)}{q(a|s)} Q_{\theta_{\text{old}}}\right] \\
&\text{subject to } \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}}\left[D_{\text{KL}}(\pi_{\text{old}}(\cdot|s)||\pi_{\text{new}}(\cdot|s))\right] \leq \delta
\end{aligned} \tag{3}
$$

Repeat the following steps:

- Use Monte Carlo simulation to collect trajectories.
  - Single path: generate 1 episode, then move to step 2
  - Vine: generate a number of trajectories, then choose a subset of states and samples actions from these state to generate new branching trajectories.
- Construct the estimated objective and constraint of Problem (3)
- Approximately solve this optimization problem using conjugate gradient algorithm followed by a line search.



Experiment result



Experiment result

	B. Rider	*Breakout*	*Enduro*	*Pong*	*Q*bert*	*Seaquest*	*S. Invaders*
Random	354	1.2	0	-20.4	157	110	179
Human (Mnih et al., 2013)	7456	31.0	368	-3.0	18900	28010	3690
Deep Q Learning (Mnih et al., 2013)	4092	168.0	470	20.0	1952	1705	581
UCC-I (Guo et al., 2014)	5702	380	741	21	20025	2995	692
TRPO - single path	1425.2	10.8	534.6	20.9	1973.5	1908.6	568.4
TRPO - vine	859.5	34.2	430.8	20.9	7732.5	788.4	450.2

Table 1. Performance comparison for vision-based RL algorithms on the Atari domain. Our algorithms (bottom rows) were run once on each task, with the same architecture and parameters. Performance varies substantially from run to run (with different random initializations of the policy), but we could not obtain error statistics due to time constraints.

Conclusion

- Theorem 1 justifies for the surrogate objective function.
- Proposing using hard constraint instead of using penalty objective.
- Single path or vine using MC for estimating the minimization problem.
- Using conjugate gradient method for search direction, and line search to ensure the current step satisfies constraint.

Optimization method

$$\max_{\theta} L_{\theta_{old}}(\theta)$$
$$\text{subject to } \overline{D}_{\mathrm{KL}}(\theta_{\mathrm{old}}, \theta) \leq \delta$$

Step 1: find the optimal direction to update

- The fisher information matrix is defined as $\boldsymbol{F} = \mathbb{E}_{x \sim p(x;\theta)}[\nabla \log p(x;\theta) \nabla \log p(x;\theta)^T]$
- Fact: $\boldsymbol{F} = -\boldsymbol{H}_{\theta}[\log \nabla p(x;\theta)]$
- A derived fact: $\boldsymbol{F} = \boldsymbol{H}_{D_{\mathrm{KL}}}$
- A derived approximation: $D_{\mathrm{KL}}(p_{\theta} || p_{\theta'}) \approx \frac{1}{2}(\theta' - \theta)^T \boldsymbol{F} (\theta' - \theta)$

Then we can dirive the optimal direction by

- Linear approximation of the objective:

$$L_{\theta_{\text{old}}}(\theta) \approx L_{\theta_{\text{old}}} + \nabla_{\theta} L_{\theta_{\text{old}}}(\theta) \mid_{\theta = \theta_{\text{old}}} [\theta - \theta_{\text{old}}]$$

- Quadratic approximation of the constrain:

$$D_{\mathrm{KL}}(\theta_{\text{old}}, \theta) \approx \frac{\lambda}{2}(\theta - \theta_{\text{old}})^T \boldsymbol{F} (\theta - \theta_{\text{old}}),$$

  where $\boldsymbol{F}$ is Fisher information matrix.
- Lagrangian form

$$f(\theta) := L_{\theta_{\text{old}}} + \nabla_{\theta} L_{\theta_{\text{old}}}(\theta) \mid_{\theta = \theta_{\text{old}}} [\theta - \theta_{\text{old}}] + \frac{\lambda}{2}(\theta - \theta_{\text{old}})^T \boldsymbol{F} (\theta - \theta_{\text{old}})$$

Find optimal direction

- To find optimal f^*, we can find θ^* such as $\nabla f(\theta^*) = 0$,

  $$0 = \nabla_\theta L_{\text{old}}(\theta) \mid_{\theta=\theta_{\text{old}}} + \lambda \boldsymbol{F}(\theta^* - \theta_{\text{old}}) \Leftrightarrow \theta^* = \theta_{\text{old}} - \lambda \boldsymbol{F}^{-1} \nabla_\theta L_{\theta_{\text{old}}}(\theta) \mid_{\theta=\theta_{\text{old}}}$$

- Therefore, the optimal direction is $\boldsymbol{y} = \boldsymbol{F}^{-1} \nabla_\theta L_{\theta_{\text{old}}}(\theta)|_{\theta=\theta_{\text{old}}}$
- Conjugate gradient can be used to solve $\boldsymbol{F}\boldsymbol{y} = \nabla_\theta L_{\theta_{\text{old}}}(\theta) \mid_{\theta=\theta_{\text{old}}}$

Connection to other methods

- Standard policy gradient

$$\max_{\theta} \left[\nabla_\theta L_{\theta_{\text{old}}}(\theta) \mid_{\theta=\theta_{\text{old}}} \cdot(\theta - \theta_{\text{old}})\right]$$
$$\text{subject to } \frac{1}{2}\left\|\theta - \theta_{\text{old}}\right\|^2 \leq \delta$$

- Natural policy gradient (Kakade, 2002)

$$\max_{\theta} \left[\nabla_\theta L_{\theta_{\text{old}}}(\theta) \mid_{\theta=\theta_{\text{old}}} \cdot(\theta - \theta_{\text{old}})\right]$$
$$\text{subject to } \frac{1}{2}(\theta_{\text{old}} - \theta)^T \boldsymbol{F}(\theta_{\text{old}} - \theta) \leq \delta$$

Proximal policy optimization

"Proximal policy optimization algorithms" (PPO) improved upon this by using only first-order derivative.

- Recall the objective in TRPO is

$$\mathbb{E}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} Q_{\theta_{\text{old}}}\right] = \mathbb{E}[r(\theta) Q_{\theta_{\text{old}}}]$$

- In PPO, the objective is

$$\mathbb{E}\left[\min(r(\theta) Q_{\theta_{\text{old}}}, \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon) Q_{\theta_{\text{old}}})\right]$$

PPO experiment result

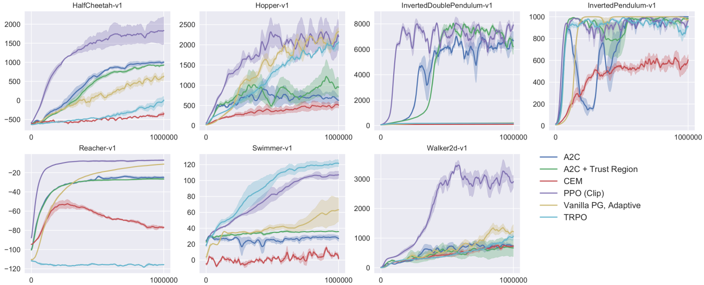


Figure 3: Comparison of several algorithms on several MuJoCo environments, training for one million timesteps.