Constrained Policy Optimization

B. Ravindran

- In this lecture we will be work with on-policy algorithms
- Some issues with such algorithms
  - Difficult to choose step size
    - Step too big ==> Bad policy ==> Data collected under bad policy ==> Hard to recover
    - Step too small ==> Inefficient use of experience

Consider a family of policies with parametrization:

$$\pi_\theta(a) = \begin{cases} \sigma(\theta) & a = 1 \\ 1 - \sigma(\theta) & a = 2 \end{cases}$$

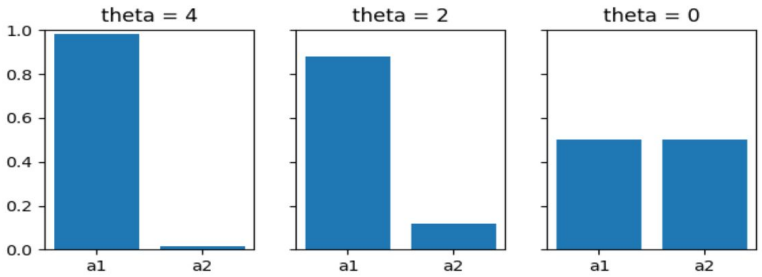


Figure: Small changes in the policy parameters can unexpectedly lead to **big** changes in the policy.

Constrained Policy Optimization

- Take the largest possible step to improve policy performance, while satisfying the constraints.

Constrained Policy Optimization

- Take the largest possible step to improve policy performance, while satisfying the constraints.

- One way to define a constraint is to limit the extent to which the updated policy can differ from the current policy. This constraint can help maintain stability during the learning process and balance exploration and exploitation.

Trust Region Policy Optimization (TRPO)

- Trust Region constraint is defined by setting a limit on the **KL divergence** between the current (π) and the update ($\tilde{\pi}$) policies.

Trust Region Policy Optimization (TRPO)

- Trust Region constraint is defined by setting a limit on the **KL divergence** between the current (π) and the update ($\tilde{\pi}$) policies.

- This ensures that the updated policy is not too different from the current policy, which helps in maintaining stability during the learning process.

Trust Region Policy Optimization (TRPO)

- Trust Region constraint is defined by setting a limit on the **KL divergence** between the current (π) and the update ($\tilde{\pi}$) policies.

- This ensures that the updated policy is not too different from the current policy, which helps in maintaining stability during the learning process.

$$\tilde{\pi} = arg\max_{\tilde{\pi}} \mathop{\mathbb{E}}_{s,a\sim\rho_\pi} \left[\frac{\tilde{\pi}(a|s)}{\pi(a|s)} A_\pi(s,a)\right]$$

$$s.t. \mathop{\mathbb{E}}_{s\sim\rho_\pi} [D_{KL}(\pi(.|s)||\tilde{\pi}(.|s))] \leq \delta$$

Understanding TRPO

- We define value of a policy as:

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t) \right], \text{ where}$$

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

Understanding TRPO

- We define value of a policy as:

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t) \right], \text{ where}$$
$$s_0 \sim \rho_0(s_0),\ a_t \sim \pi(a_t|s_t),\ s_{t+1} \sim P(s_{t+1}|s_t, a_t)$$

- One can also rewrite the value of a new policy w.r.t another policy as:

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

Understanding TRPO

And it can be re-written as :

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

$$= \eta(\pi) + \sum_{t=0}^{\infty} \sum_{s} P(s_t = s|\tilde{\pi}) \sum_{a} \tilde{\pi}(a|s) \gamma^t A_\pi(s, a)$$

$$= \eta(\pi) + \sum_{s} \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\tilde{\pi}) \sum_{a} \tilde{\pi}(a|s) A_\pi(s, a)$$

$$= \eta(\pi) + \sum_{s} \rho_{\tilde{\pi}}(s) \sum_{a} \tilde{\pi}(a|s) A_\pi(s, a)$$

Understanding TRPO

And it can be re-written as :

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

$$= \eta(\pi) + \sum_{t=0}^{\infty} \sum_{s} P(s_t = s|\tilde{\pi}) \sum_{a} \tilde{\pi}(a|s) \gamma^t A_\pi(s, a)$$

$$= \eta(\pi) + \sum_{s} \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\tilde{\pi}) \sum_{a} \tilde{\pi}(a|s) A_\pi(s, a)$$

$$= \eta(\pi) + \sum_{s} \rho_{\tilde{\pi}}(s) \sum_{a} \tilde{\pi}(a|s) A_\pi(s, a)$$

Understanding TRPO

And it can be re-written as :

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

$$= \eta(\pi) + \sum_{t=0}^{\infty} \sum_{s} P(s_t = s|\tilde{\pi}) \sum_{a} \tilde{\pi}(a|s) \gamma^t A_\pi(s, a)$$

$$= \eta(\pi) + \sum_{s} \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\tilde{\pi}) \sum_{a} \tilde{\pi}(a|s) A_\pi(s, a)$$

$$= \eta(\pi) + \sum_{s} \rho_{\tilde{\pi}}(s) \sum_{a} \tilde{\pi}(a|s) A_\pi(s, a)$$

Understanding TRPO

And it can be re-written as :

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

Policy Iteration: Notice that by choosing $\tilde{\pi}(s) = \arg\max_a A_\pi(s, a)$, we can guarantee that $\eta(\tilde{\pi}) \geq \eta(\pi)$

Understanding TRPO

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

- The dependency of $\rho_{\tilde{\pi}}$ on $\tilde{\pi}$ make the above equation difficult to find better policy directly.

Understanding TRPO

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

- The dependency of $\rho_{\tilde{\pi}}$ on $\tilde{\pi}$ make the above equation difficult to find better policy directly.
- Instead we define an approximation as:

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

Understanding TRPO

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

- The dependency of $\rho_{\tilde{\pi}}$ on $\tilde{\pi}$ make the above equation difficult to find better policy directly.
- Instead we define an approximation as:

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

- Improving the lower bound of $\eta(\tilde{\pi})$ in turn helps improve the policy

$$\eta(\tilde{\pi}) \geq L_\pi(\tilde{\pi}) - C D_{\mathrm{KL}}^{\max}(\pi, \tilde{\pi}),$$

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

$$D_{KL}^{max}(\pi, \tilde{\pi}) = \max_s D_{KL}(\pi(.|s)||\tilde{\pi}(.|s))$$

Understanding TRPO

- Therefore our objective function is:

$$\tilde{\pi} = \max_{\tilde{\pi}} \; L_{\pi}(\tilde{\pi})$$
$$s.t. D_{KL}^{max}(\pi, \tilde{\pi}) \leq \delta$$

- Optimizing the above equation is impractical due to the large number of constraints introduced by

$$D_{KL}^{max}(\pi, \tilde{\pi}) = \max_{s} D_{KL}(\pi(.|s)||\tilde{\pi}(.|s))$$

Understanding TRPO

- We approximate our optimization function as:

$$\tilde{\pi} = \max_{\tilde{\pi}} \; L_{\pi}(\tilde{\pi})$$
$$s.t. \; \mathop{\mathbb{E}}_{s \sim \rho_{\pi}} [D_{KL}(\pi(.|s)||\tilde{\pi}(.|s))] \leq \delta$$

Understanding TRPO

- We approximate our optimization function as:

$$\tilde{\pi} = \max_{\tilde{\pi}} \; L_{\pi}(\tilde{\pi})$$

$$s.t. \; \mathop{\mathbb{E}}_{s \sim \rho_{\pi}} [D_{KL}(\pi(.|s)||\tilde{\pi}(.|s))] \leq \delta$$

- We further reduce our above optimization problem becomes:

$$L_{\pi}(\tilde{\pi}) = \eta(\pi) + \sum_{s} \rho_{\pi}(s) \sum_{a} \tilde{\pi}(a|s) A_{\pi}(s,a).$$

$$\tilde{\pi} = arg \max_{\tilde{\pi}} \mathop{\mathbb{E}}_{s,a \sim \rho_{\pi}} \left[\frac{\tilde{\pi}(a|s)}{\pi(a|s)} A_{\pi}(s,a) \right]$$

$$s.t. \; \mathop{\mathbb{E}}_{s \sim \rho_{\pi}} [D_{KL}(\pi(.|s)||\tilde{\pi}(.|s))] \leq \delta$$

Understanding TRPO

These nonlinear-constraints are difficult to optimize.

$$\tilde{\pi} = \max_{\tilde{\pi}} \; L_{\pi}(\tilde{\pi})$$
$$s.t. \; \mathop{\mathbb{E}}_{s \sim \rho_{\pi}} [D_{KL}(\pi(.|s)||\tilde{\pi}(.|s))] \leq \delta$$

Understanding TRPO

These nonlinear-constraints are difficult to optimize.

$$\tilde{\pi} = \max_{\tilde{\pi}} \; L_{\pi}(\tilde{\pi})$$
$$s.t. \; \mathop{\mathbb{E}}_{s \sim \rho_{\pi}} [D_{KL}(\pi(.|s)||\tilde{\pi}(.|s))] \leq \delta$$

TRPO makes some approximations with respect to theta parameters of the policy.

$$\mathcal{L}(\theta_k, \theta) \approx g^T(\theta - \theta_k)$$
$$\bar{D}_{KL}(\theta||\theta_k) \approx \frac{1}{2}(\theta - \theta_k)^T H(\theta - \theta_k)$$

Understanding TRPO

These nonlinear-constraints are difficult to optimize.

$$\tilde{\pi} = \max_{\tilde{\pi}} \; L_{\pi}(\tilde{\pi})$$
$$s.t. \; \mathop{\mathbb{E}}_{s \sim \rho_{\pi}} [D_{KL}(\pi(.|s)||\tilde{\pi}(.|s))] \leq \delta$$

TRPO makes some approximations with respect to theta parameters of the policy.

$$\mathcal{L}(\theta_k, \theta) \approx g^T(\theta - \theta_k)$$
$$\bar{D}_{KL}(\theta||\theta_k) \approx \frac{1}{2}(\theta - \theta_k)^T H(\theta - \theta_k)$$

H: second order approximation

Understanding TRPO

Therefore, at every iteration (say k+1), it solves the following constraint optimization problem:

$$\theta_{k+1} = \arg\max_{\theta} g^T(\theta - \theta_k)$$
$$\text{s.t. } \frac{1}{2}(\theta - \theta_k)^T H(\theta - \theta_k) \leq \delta.$$

We have a closed form solution to the above QP problem:

$$\theta_{k+1} = \theta_k + \sqrt{\frac{2\delta}{g^T H^{-1} g}} H^{-1} g.$$

Understanding TRPO

Directly updating theta with the QP solution may violate the KL constraint.

$$\theta_{k+1} = \theta_k + \sqrt{\frac{2\delta}{g^T H^{-1} g}} H^{-1} g. \xrightarrow{\text{may violate}} \mathop{\mathbb{E}}_{s\sim\rho_\pi}[D_{KL}(\pi(.|s)||\tilde{\pi}(.|s))] \leq \delta$$

Understanding TRPO

Directly updating theta with the QP solution may violate the KL constraint.

$$\theta_{k+1} = \theta_k + \sqrt{\frac{2\delta}{g^T H^{-1} g}} H^{-1} g. \xrightarrow{\text{may violate}} \mathop{\mathbb{E}}_{s\sim\rho_\pi}[D_{KL}(\pi(.|s)||\tilde{\pi}(.|s))] \leq \delta$$

Therefore, we choose alpha_j such that the constraints are satisfied.

$$\theta_{k+1} = \theta_k + \alpha^j \sqrt{\frac{2\delta}{g^T H^{-1} g}} H^{-1} g,$$

- TRPO can be computationally expensive as it uses second-order approximation of the KL divergence constraint.

Proximal Policy Optimization (PPO)

PPO: maximize the clipped objective which is defined as:

$$L(s, a, \theta_k, \theta) = \min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s, a),\ \ \text{clip}\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}, 1-\epsilon, 1+\epsilon\right) A^{\pi_{\theta_k}}(s, a)\right)$$

Understanding PPO

Clipped Objective:

$$L(s, a, \theta_k, \theta) = \min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s, a),\ \ \text{clip}\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}, 1-\epsilon, 1+\epsilon\right) A^{\pi_{\theta_k}}(s, a)\right)$$

Alternatively, we can rewrite the equation:

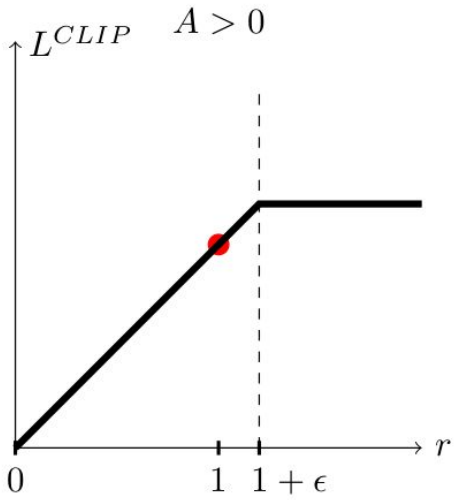
$$L(s, a, \theta_k, \theta) = \min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s, a),\ \ g(\epsilon, A^{\pi_{\theta_k}}(s, a))\right)$$

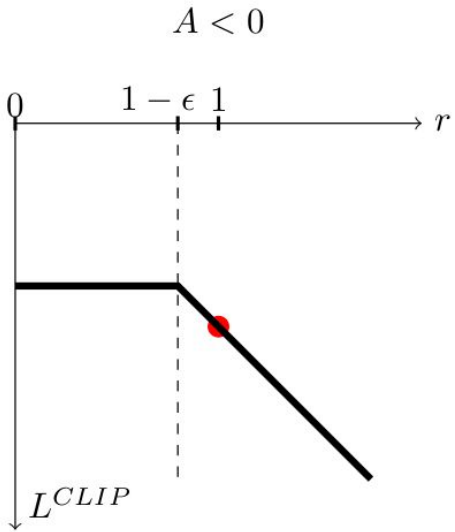
where,

$$g(\epsilon, A) = \begin{cases} (1+\epsilon)A & A \geq 0 \\ (1-\epsilon)A & A < 0. \end{cases}$$

PPO

$$\hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$





PPO Algorithm

Input: initial policy parameters θ_0, clipping threshold ϵ
for $k = 0, 1, 2, \ldots$ **do**
Collect set of partial trajectories $\mathcal{D}_k$ on policy $\pi_k = \pi(\theta_k)$
Estimate advantages $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm
Compute policy update

$$\theta_{k+1} = \arg\max_{\theta} \mathcal{L}_{\theta_k}^{CLIP}(\theta)$$

by taking K steps of minibatch SGD (via Adam), where

$$\mathcal{L}_{\theta_k}^{CLIP}(\theta) = \mathop{\mathrm{E}}_{\tau \sim \pi_k} \left[\sum_{t=0}^{T} \left[\min(r_t(\theta)\hat{A}_t^{\pi_k}, \mathrm{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t^{\pi_k}) \right] \right]$$

end for

Results

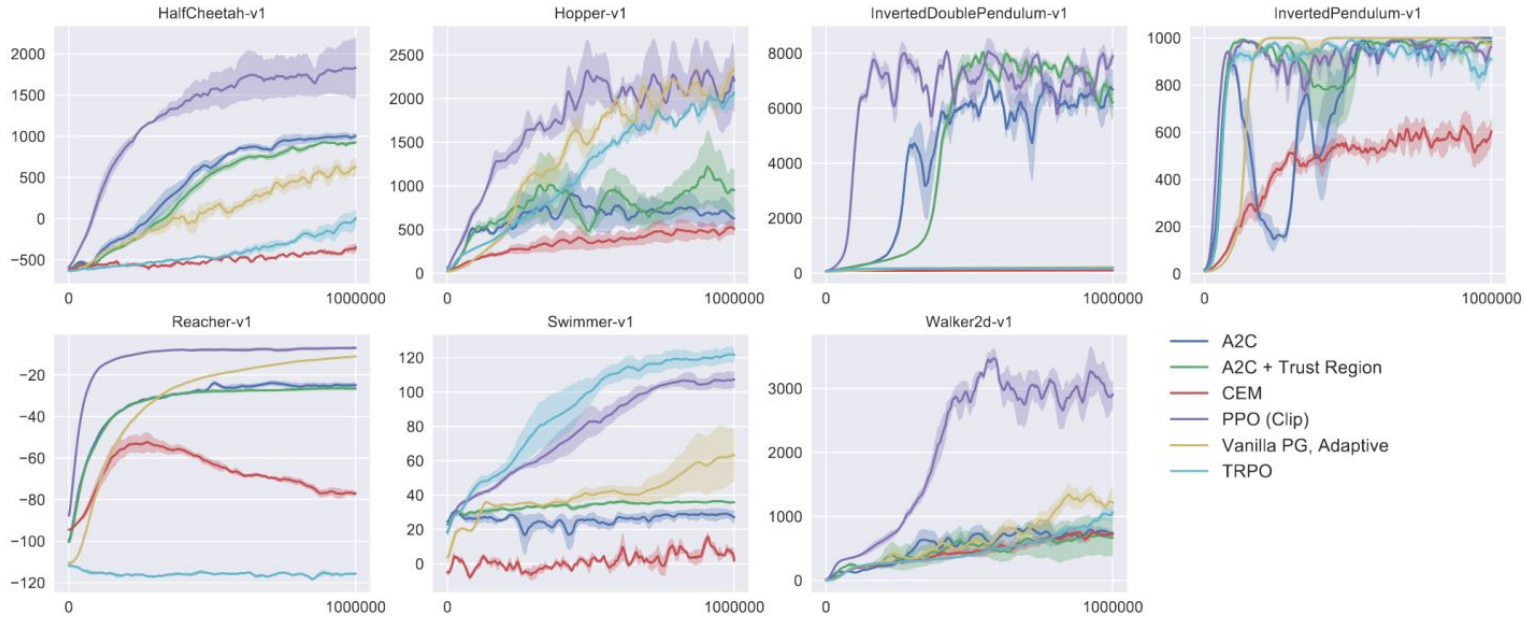


Figure: Performance comparison between PPO with clipped objective and various other deep RL methods on a slate of MuJoCo tasks. [10]

Implementation details for PPO

- Typically, clipping threshold ϵ = 0.2.
- Advantage is estimated using generalized advantage estimation

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + \cdots + (\gamma\lambda)^{T-t+1}\delta_{T-1},$$
$$\text{where} \quad \delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$$

- Separate actor and critic networks.
- tanh activation functions.
- Orthogonal initialization of actor, critic networks with appropriate scaling.
- Gradient clipping - ensure that the norm of the concatenated gradients of all network parameters does not exceed 0.5.

- TRPO can be computationally expensive as it uses second-order approximation of the KL divergence constraint.

- Whereas, PPO is simpler than TRPO, as it does not require second-order approximations of the KL divergence constraint. Instead, it uses a clipped surrogate objective that constrains the policy update to be within a specified range.

References

- Schulman, John, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. "Trust region policy optimization." In *International Conference on Machine Learning*, pp. 1889-1897. PMLR, 2015.
- Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms." *arXiv preprint arXiv:1707.06347* (2017).
- Schulman, John, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. "High-dimensional continuous control using generalized advantage estimation." *arXiv preprint arXiv:1506.02438* (2015).