Proximal Policy Optimization (PPO)

Konpat Preechakul

Chulalongkorn University

November 2019

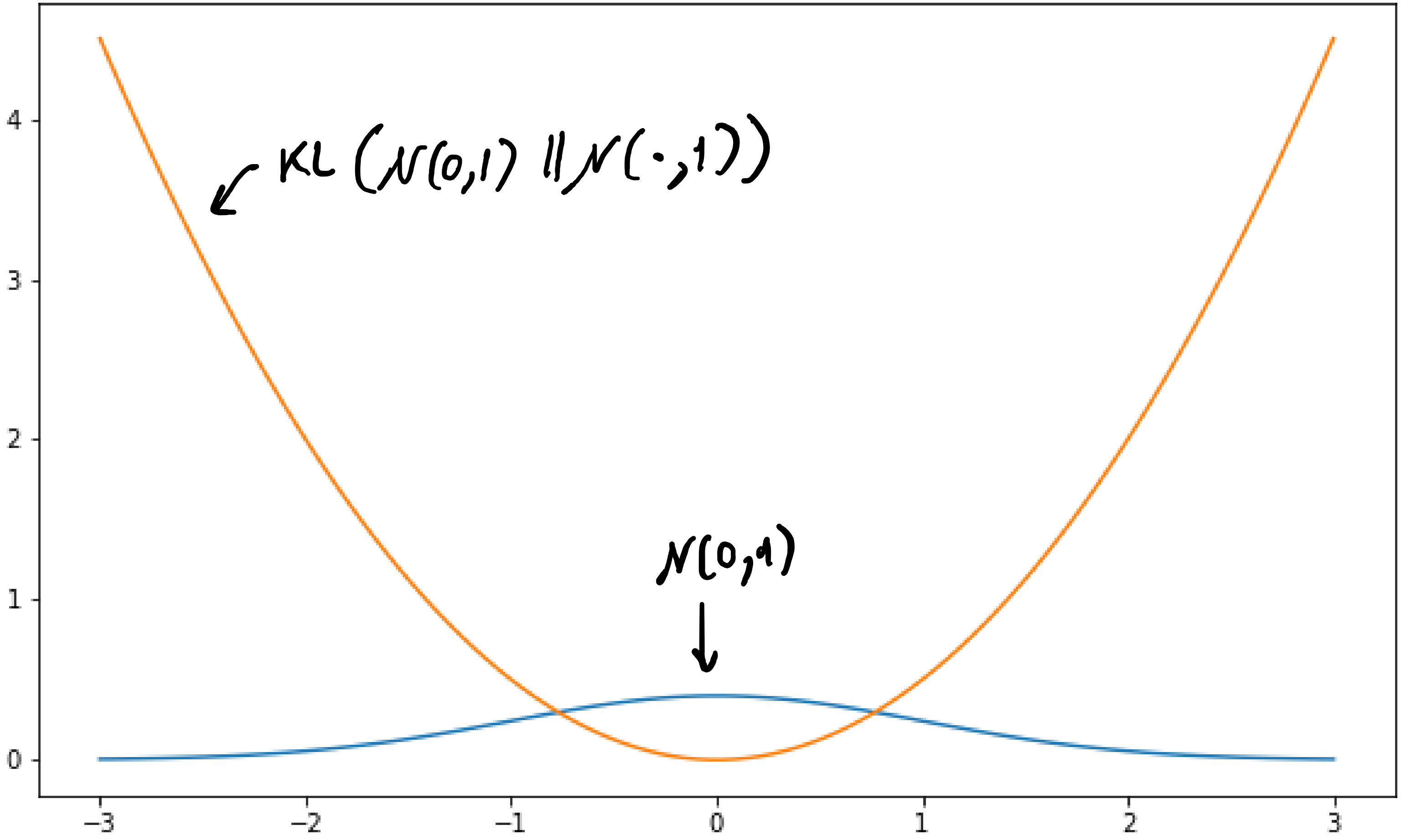
Policy gradient fails

- Critic is not oracle, it has its flaws
- Critic is prone to forgetting
- How to know when to trust?
- **How to limit the trust?**
- Not updating too much ...
- KL divergence:

$$\mathrm{KL}(P\|Q) = \sum_x P(x)[\log P(x) - \log Q(x)]$$

KL Divergence in picture

$$\mathrm{KL}(P\|Q) = \sum_x P(x)[\log P(x) - \log Q(x)]$$



Trust region policy optimization

$$d^* = \operatorname*{argmax}_d \; J(\theta + d) \quad \text{s.t. } \mathrm{KL}(\theta \| \theta + d) = c$$

- A constrained optimization

- We relax it using **Lagrangian:**

$$\mathcal{L}(d, \lambda) = J(\theta + d) + \lambda \left(\mathrm{KL}(\theta \| \theta + d) - c\right)$$

- Optimal d is at the critical point

$$\nabla_{d,\lambda} \mathcal{L} = 0$$

Policy improvement guarantee

$$\mathcal{A}_\pi(\pi') = \mathbb{E}_{a_t \sim \pi}\left[\frac{\pi'(a_t|s_t)}{\pi(a_t|s_t)}\gamma^t A^\pi(s_t, a_t)\right]$$

$$J(\theta') - J(\theta) = \sum_t \mathbb{E}_{s_t \sim P'(s_t)}\left[\mathcal{A}_\pi(\pi')\right]$$

Lower bound:

$$J(\theta') - J(\theta) = \sum_t \mathbb{E}_{s_t \sim P(s_t)}\left[\mathcal{A}_\pi(\pi')\right] - \sum_t \epsilon t \mathcal{O}\left(\frac{r_{\max}}{1-\gamma}\right)$$

$$\text{s.t. } \sqrt{\frac{1}{2}\mathrm{KL}^{\max}(\pi\|\pi')} = \epsilon$$

Constrained optimization is hard

- Using a fixed constant is not likely to work:

$$\overline{J}(\theta + d) = J(\theta + d) + \beta \mathrm{KL}(\theta \| \theta + d)$$

- Lagrangian involves exotic terms like “inverse”

$$\left(\nabla_\theta^2 KL \right)^{-1}$$

Is there an easy way to constrain KL?

Goal

- Design objective function J
- That has “zero” gradient
- When the constraint is breached

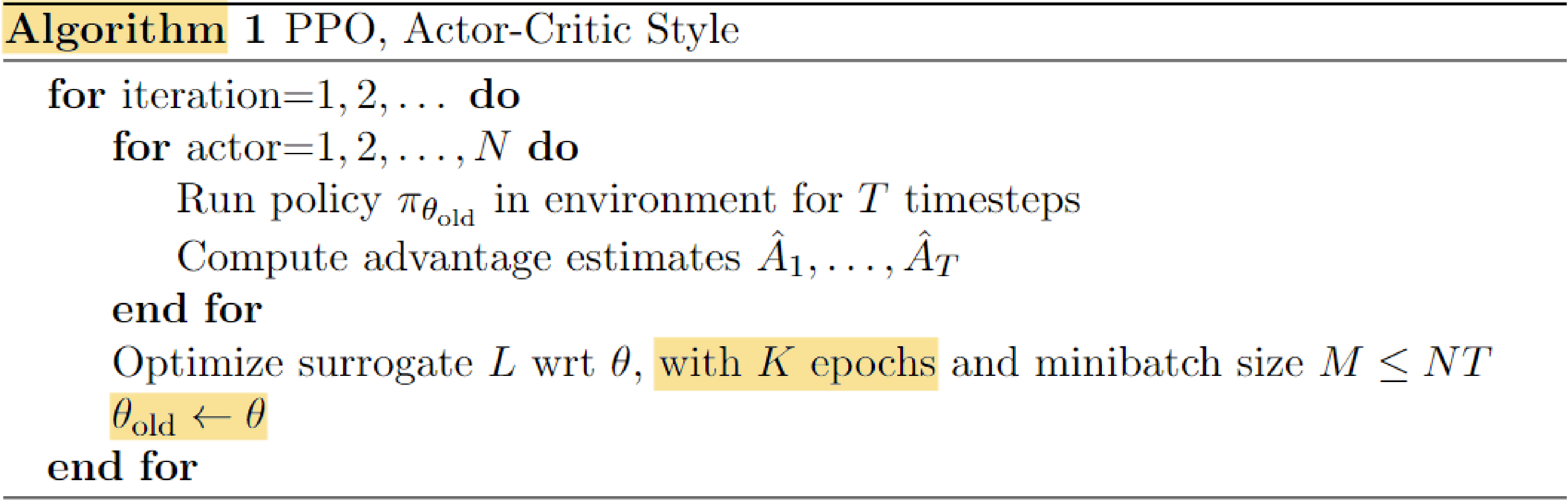
- Optimize normally …

First attempt

Possible pseudocode

Is there a one-liner?

Pseudocode



Algorithm 1 PPO, Actor-Critic Style

```
for iteration=$1, 2, \ldots$ do
    for actor=$1, 2, \ldots, N$ do
        Run policy $\pi_{\theta_{\text{old}}}$ in environment for $T$ timesteps
        Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
    end for
    Optimize surrogate $L$ wrt $\theta$, with $K$ epochs and minibatch size $M \leq NT$
    $\theta_{\text{old}} \leftarrow \theta$
end for
```

Results

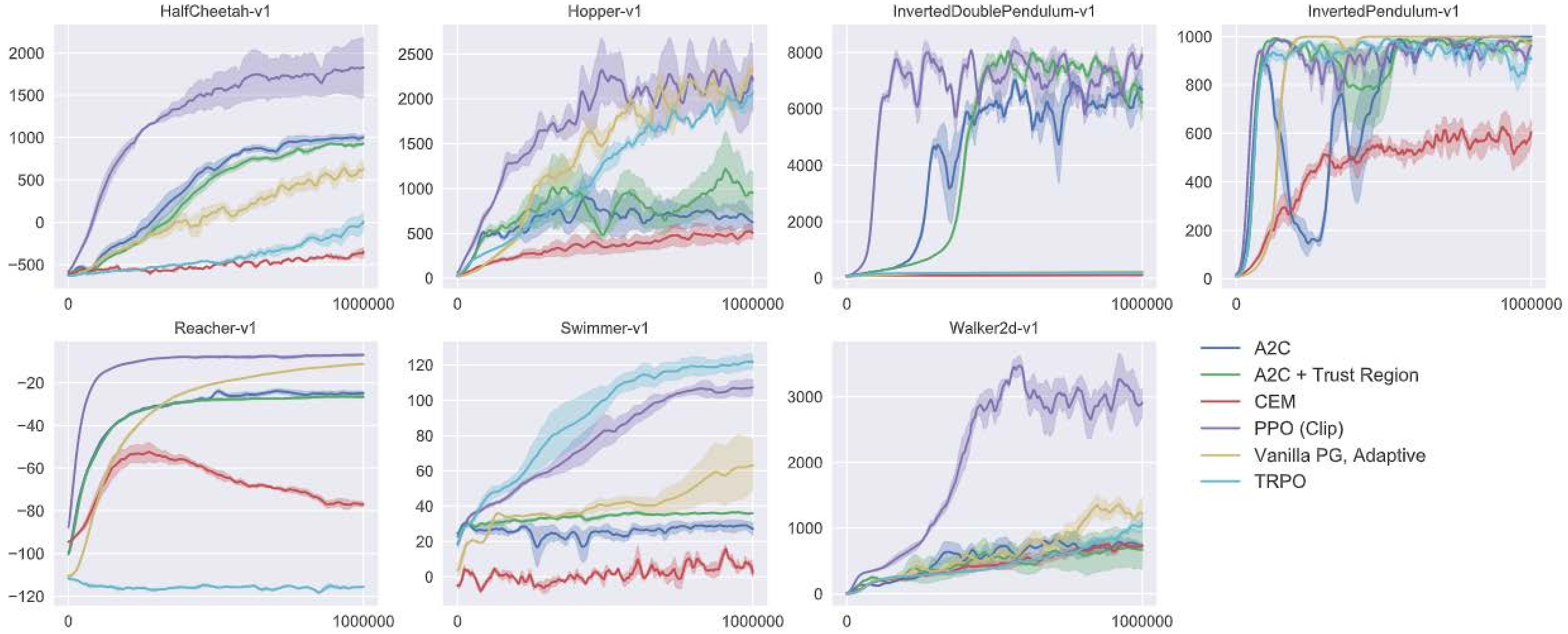


Figure 3: Comparison of several algorithms on several MuJoCo environments, training for one million timesteps.