Trust Region Policy Optimization (TRPO)

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, Pieter Abbeel

Presenter: Jingkang Wang
Date: January 21, 2020

A Taxonomy of RL Algorithms

We are here!

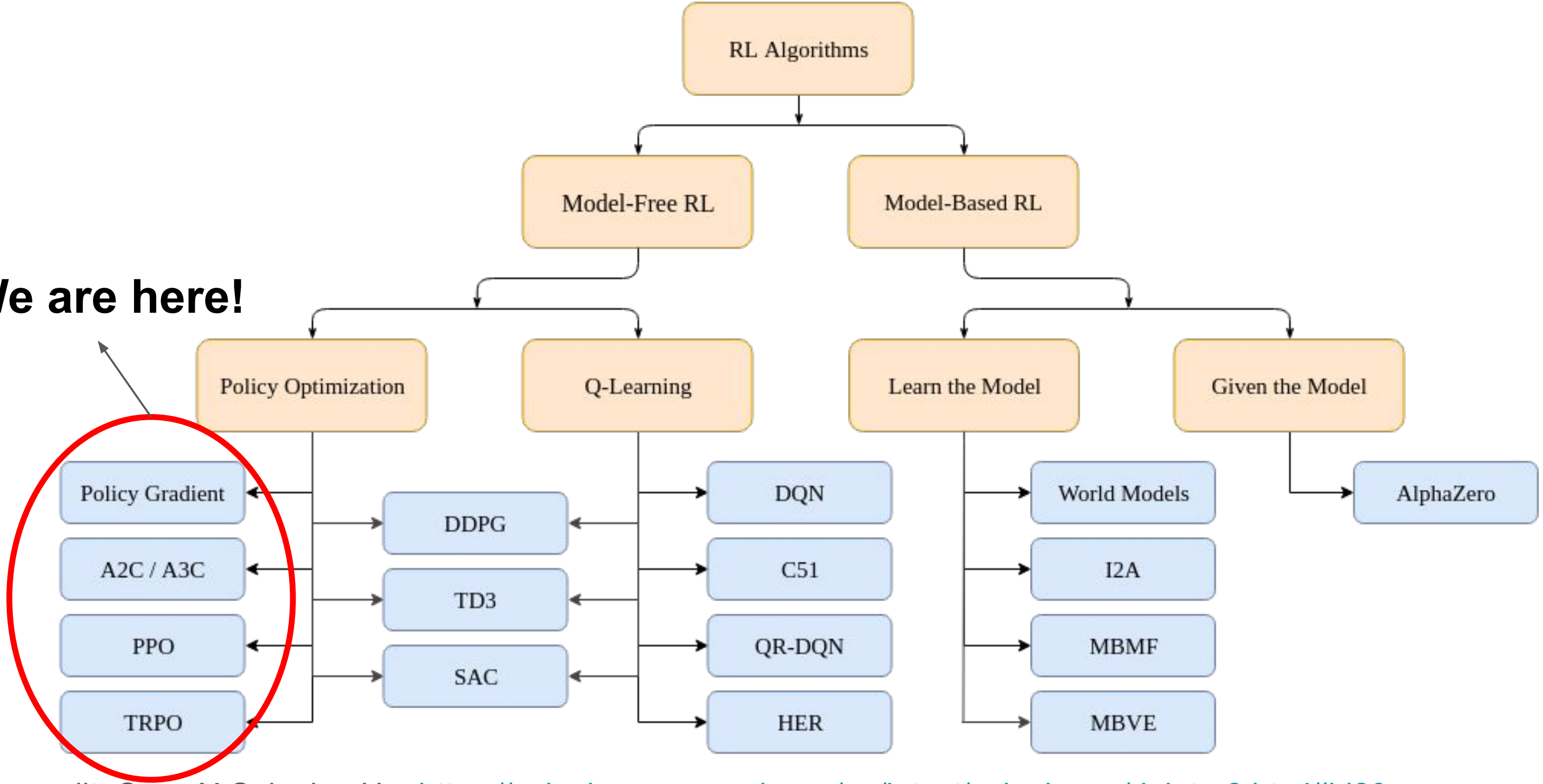


Image credit: OpenAI Spinning Up, https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html#id20

Policy Gradients (Preliminaries)

1) Score function estimator (SF, also referred to as REINFORCE):

$$\nabla_\theta \mathbb{E}_z \left[f(z) \right] = \mathbb{E}_z \left[f(z) \nabla_\theta \log p_\theta(z) \right]$$

Proof: $$\mathbb{E}_z[f(z)\nabla_\theta \log p_\theta(z)] = \mathbb{E}_z[\frac{f(z)}{p_\theta(z)}\nabla_\theta p_\theta(z)] = \int p_\theta(z)\frac{f(z)}{p_\theta(z)}\nabla_\theta p_\theta(z)dz$$
$$= \nabla_\theta \int f(z) p_\theta(z) dz = \nabla_\theta \mathbb{E}_z[f(z)]$$

Remark: $f(z)$ can be either differentiable and non-differentiable functions

Policy Gradients (Preliminaries)

1) Score function estimator (SF, also referred to as REINFORCE):

$$\nabla_\theta \mathbb{E}_z \left[f(z) \right] = \mathbb{E}_z \left[f(z) \nabla_\theta \log p_\theta(z) \right]$$

2) Subtracting a control variate $b(z)$ $\quad \mu_b = \mathbb{E}_z \left[b(z) \nabla_\theta \log p_\theta(z) \right]$

$$\begin{aligned} \nabla_\theta \mathbb{E}_z \left[f(z) \right] &= \mathbb{E}_z \left[f(z) \nabla_\theta \log p_\theta(z) + \left(b(z) \nabla_\theta \log p_\theta(z) - b(z) \nabla_\theta \log p_\theta(z) \right) \right] \\ &= \mathbb{E}_z \left[\left(f(z) - b(z) \right) \nabla_\theta \log p_\theta(z) \right] + \mu_b \end{aligned}$$

Remark: if baseline is not a function of z $\quad \nabla_\theta \mathbb{E}[f(z)] = \mathbb{E}_z[(f(z) - b) \nabla_\theta \log p_\theta(z)]$

Policy Gradients (PG)

Policy Gradient Theorem [1]:

$$\nabla_\theta \eta(\pi_\theta) = \mathbb{E}_{\rho_\pi, \pi}\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{Q}(s_t, a_t)\right]$$

$\mathbb{E}_\tau[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$ — **Expected reward**

$\sum_{t=0}^{\infty} \gamma^t p(s_t = s)$ — **Visitation frequency**

$\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'}$ — **State-action function (Q-value)**

Subtract the Baseline - state-value function $\hat{A}(s_t, a_t) = \hat{Q}(s_t, a_t) - \hat{V}(s_t)$

$$\nabla_\theta \eta(\pi_\theta) = \mathbb{E}_{\rho_\pi, \pi}\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{A}(s_t, a_t)\right] \qquad \hat{V}(s_t) = \sum \pi(a_t|s_t)\hat{Q}(s_t, a_t)$$

Policy Gradients (PG)

Policy Gradient Theorem [1]:

$$\nabla_\theta \eta(\pi_\theta) = \mathbb{E}_{\rho_\pi,\pi}\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{Q}(s_t,a_t)\right]$$

$\mathbb{E}_\tau[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)]$ — **Expected reward**

$\sum_{t=0}^{\infty}\gamma^t p(s_t = s)$ — **Visitation frequency**

$\sum_{t'=t}^{\infty}\gamma^{t'-t}r_{t'}$ — **State-action function (Q-value)**

Subtract the Baseline - state-value function $\hat{A}(s_t,a_t) = \hat{Q}(s_t,a_t) - \hat{V}(s_t)$

$$\nabla_\theta \eta(\pi_\theta) = \mathbb{E}_{\rho_\pi,\pi}\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{A}(s_t,a_t)\right]$$

$$\hat{V}(s_t) = \sum \pi(a_t|s_t)\hat{Q}(s_t,a_t)$$

Motivation - Problem in PG

REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for π_*

Input: a differentiable policy parameterization $\pi(a|s, \boldsymbol{\theta})$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \boldsymbol{\theta})$
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ (G_t)
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta})$

How to choose the step size?

Motivation - Problem in PG

REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for π_*

Input: a differentiable policy parameterization $\pi(a|s, \boldsymbol{\theta})$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \boldsymbol{\theta})$
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ (G_t)
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta})$

How to choose the step size?

too large? 1) bad policy -> 2) collected data under bad policy

too small? cannot leverage data sufficiently

Motivation - Problem in PG

REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for π_*

Input: a differentiable policy parameterization $\pi(a|s, \boldsymbol{\theta})$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
    Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \boldsymbol{\theta})$
    Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
        $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ (G_t)
        $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta})$

How to choose the step size?

too large? 1) bad policy → 2) collected data under bad policy

too small? cannot leverage data sufficiently

Cannot recover!

Motivation: Why trust region optimization?

Line search
(like gradient ascent)

Trust region

Image credit: https://medium.com/@jonathan_hui/rl-trust-region-policy-optimization-trpo-explained-a6ee04eeeee9

TRPO - What Loss to optimize?

- Original objective

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

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

- Improvement of new policy over old policy [1]

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

- Local approximation (visitation frequency is unknown)

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

$$L_{\pi_{\theta_0}}(\pi_{\theta_0}) = \eta(\pi_{\theta_0}), \quad \nabla_\theta L_{\pi_{\theta_0}}(\pi_\theta)\big|_{\theta=\theta_0} = \nabla_\theta \eta(\pi_\theta)\big|_{\theta=\theta_0}$$

TRPO - What Loss to optimize?

- Original objective

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

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

- Improvement of new policy over old policy [1]

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

- Local approximation (visitation frequency is unknown)

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

$$L_{\pi_{\theta_0}}(\pi_{\theta_0}) = \eta(\pi_{\theta_0}), \quad \nabla_\theta L_{\pi_{\theta_0}}(\pi_\theta)\big|_{\theta=\theta_0} = \nabla_\theta \eta(\pi_\theta)\big|_{\theta=\theta_0}$$

Proof: Relation between new and old policy:

$$A^{\pi_{\text{old}}}(s,a) = \mathbb{E}_{s' \sim P(s' \mid s,a)}\left[r(s) + \gamma V^{\pi_{\text{old}}}(s') - V^{\pi_{\text{old}}}(s)\right]$$

$$\begin{aligned}
&\mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t A_{\pi_{\text{old}}}(s_t,a_t)\right] \\
&= \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t\left(r(s_t) + \gamma V^{\pi_{\text{old}}}(s_{t+1}) - V^{\pi_{\text{old}}}(s_t)\right)\right] \\
&= \mathbb{E}_{\tau\sim\pi}\left[-V^{\pi_{\text{old}}}(s_0) + \sum_{t=0}^{\infty}\gamma^t r(s_t)\right] \\
&= -\mathbb{E}_{s_0}\left[V^{\pi_{\text{old}}}(s_0)\right] + \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t)\right] \\
&= -\eta(\pi_{\text{old}}) + \eta(\pi)
\end{aligned}$$

TRPO - What Loss to optimize?

- Original objective

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

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

- Improvement of new policy over old policy [1]

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

- Local approximation (visitation frequency is unknown)

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

$$L_{\pi_{\theta_0}}(\pi_{\theta_0}) = \eta(\pi_{\theta_0}), \quad \nabla_\theta L_{\pi_{\theta_0}}(\pi_\theta)\big|_{\theta=\theta_0} = \nabla_\theta \eta(\pi_\theta)\big|_{\theta=\theta_0}$$

Surrogate Loss: Important sampling Perspective

Important Sampling:

$$\boxed{\begin{aligned}\eta(\pi) &= const + \mathbb{E}_{s\sim\pi,a\sim\pi}\left[A^{\pi_{\text{old}}}(s,a)\right] \\ &= const + \mathbb{E}_{s\sim\pi,a\sim\pi_{\text{old}}}\left[\frac{\pi(a\mid s)}{\pi_{\text{old}}(a\mid s)}A^{\pi_{\text{old}}}(s,a)\right]\end{aligned}}$$

$$\begin{aligned}L(\pi) &= \mathbb{E}_{s\sim{\color{red}\pi_{\text{old}}},a\sim\pi}\left[A^{\pi_{\text{old}}}(s_t,a_t)\right] \\ &= \mathbb{E}_{s\sim{\color{red}\pi_{\text{old}}},a\sim\pi_{\text{old}}}\left[\frac{\pi(a\mid s)}{\pi_{\text{old}}(a\mid s)}A^{\pi_{\text{old}}}(s,a)\right]\end{aligned}$$

Matches to first order for parameterized policy:

$$\begin{aligned}\nabla_\theta L(\pi_\theta)\big|_{\theta_{\text{old}}} &= \mathbb{E}_{s,a\sim\pi_{\text{old}}}\left[\frac{\nabla_\theta\pi_\theta(a\mid s)}{\pi_{\text{old}}(a\mid s)}A^{\pi_{\text{old}}}(s,a)\right]\Big|_{\theta_{\text{old}}} \\ &= \mathbb{E}_{s,a\sim\pi_{\text{old}}}\left[\nabla_\theta\log\pi_\theta(a\mid s)A^{\pi_{\text{old}}}(s,a)\right]\Big|_{\theta_{\text{old}}} = \nabla_\theta\eta(\pi_\theta)\big|_{\theta=\theta_{\text{old}}}\end{aligned}$$

Surrogate Loss: Important sampling Perspective

Important Sampling:

$$\begin{aligned}\eta(\pi) &= const + \mathbb{E}_{s\sim\pi,a\sim\pi}\left[A^{\pi_{\text{old}}}(s,a)\right] \\ &= const + \mathbb{E}_{s\sim\pi,a\sim\pi_{\text{old}}}\left[\frac{\pi(a\mid s)}{\pi_{\text{old}}(a\mid s)}A^{\pi_{\text{old}}}(s,a)\right]\end{aligned}$$

$$\begin{aligned}L(\pi) &= \mathbb{E}_{s\sim\pi_{\text{old}},a\sim\pi}\left[A^{\pi_{\text{old}}}(s_t,a_t)\right] \\ &= \mathbb{E}_{s\sim\pi_{\text{old}},a\sim\pi_{\text{old}}}\left[\frac{\pi(a\mid s)}{\pi_{\text{old}}(a\mid s)}A^{\pi_{\text{old}}}(s,a)\right]\end{aligned}$$

Matches to first order for parameterized policy:

$$\begin{aligned}\nabla_\theta L(\pi_\theta)\big|_{\theta_{\text{old}}} &= \mathbb{E}_{s,a\sim\pi_{\text{old}}}\left[\frac{\nabla_\theta\pi_\theta(a\mid s)}{\pi_{\text{old}}(a\mid s)}A^{\pi_{\text{old}}}(s,a)\right]\Big|_{\theta_{\text{old}}} \\ &= \mathbb{E}_{s,a\sim\pi_{\text{old}}}\left[\nabla_\theta\log\pi_\theta(a\mid s)A^{\pi_{\text{old}}}(s,a)\right]\Big|_{\theta_{\text{old}}} = \nabla_\theta\eta(\pi_\theta)\big|_{\theta=\theta_{\text{old}}}\end{aligned}$$

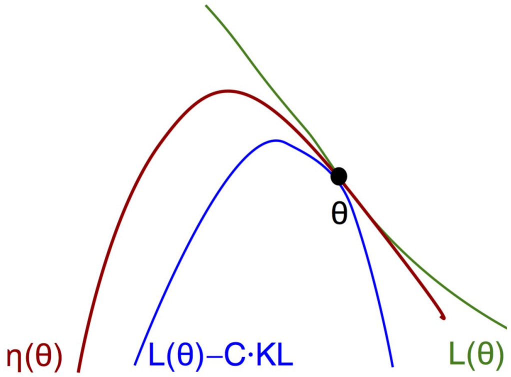
Monotonic Improvement Result

- Find the lower bound in general stochastic gradient policies

$$\eta(\tilde{\pi}) \geq L_\pi(\tilde{\pi}) - C D_{\mathrm{KL}}^{\max}(\pi, \tilde{\pi}),$$
$$\text{where } C = \frac{4\epsilon\gamma}{(1-\gamma)^2}. \quad D_{\mathrm{KL}}^{\max}(\pi, \tilde{\pi}) = \max_s D_{\mathrm{KL}}(\pi(\cdot|s) \parallel \tilde{\pi}(\cdot|s))$$

- Optimized objective: maximize $M_i(\pi)$ guarantees $\eta(\pi_i)$ non-decreasing

$$M_i(\pi) = L_{\pi_i}(\pi) - C D_{\mathrm{KL}}^{\max}(\pi_i, \pi)$$
$$\eta(\pi_{i+1}) \geq M_i(\pi_{i+1})$$
$$\eta(\pi_i) = M_i(\pi_i), \text{ therefore,}$$
$$\eta(\pi_{i+1}) - \eta(\pi_i) \geq M_i(\pi_{i+1}) - M(\pi_i).$$



Optimization of Parameterized Policies

- If we used the penalty coefficient C recommended by the theory above, the step sizes would be very small

$$\underset{\theta}{\text{maximize}}\left[L_{\theta_{\text{old}}}(\theta) - C D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta)\right]$$

Optimization of Parameterized Policies

- If we used the penalty coefficient C recommended by the theory above, the step sizes would be very small

$$\underset{\theta}{\text{maximize}}\left[L_{\theta_{\text{old}}}(\theta) - C D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta)\right]$$

- One way to take larger steps in a robust way is to use a constraint on the KL divergence between the new policy and the old policy, i.e., a trust region constraint:

$$\begin{aligned} &\underset{\theta}{\text{maximize}}\, L_{\theta_{\text{old}}}(\theta) \\ &\text{subject to } D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \le \delta. \end{aligned}$$

Optimization of Parameterized Policies

- If we used the penalty coefficient C recommended by the theory above, the step sizes would be very small

$$\underset{\theta}{\text{maximize}} \left[L_{\theta_{\text{old}}}(\theta) - C D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \right]$$

- One way to take larger steps in a robust way is to use a constraint on the KL divergence between the new policy and the old policy, i.e., a trust region constraint:

$$\begin{aligned} &\underset{\theta}{\text{maximize}}\, L_{\theta_{\text{old}}}(\theta) \\ &\text{subject to } D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \le \delta. \end{aligned} \quad \Rightarrow \quad \begin{aligned} &\underset{\theta}{\text{maximize}}\, L_{\theta_{\text{old}}}(\theta) \\ &\text{subject to } \overline{D}_{\text{KL}}^{\rho_{\theta_{\text{old}}}}(\theta_{\text{old}}, \theta) \le \delta. \end{aligned}$$

Solving the Trust-Region Constrained Optimization

1. Compute a search direction, using a linear approximation to objective and quadratic approximation to the constraint

$$Ax = g \longrightarrow \text{Conjugate gradient}$$

$$\overline{D}_{\mathrm{KL}}(\theta_{\mathrm{old}}, \theta) \approx \frac{1}{2}(\theta - \theta_{\mathrm{old}})^T A(\theta - \theta_{\mathrm{old}}) \qquad A_{ij} = \frac{\partial}{\partial \theta_i} \frac{\partial}{\partial \theta_j} \overline{D}_{\mathrm{KL}}(\theta_{\mathrm{old}}, \theta)$$

Solving the Trust-Region Constrained Optimization

1. Compute a search direction, using a linear approximation to objective and quadratic approximation to the constraint

$$Ax = g \longrightarrow \text{Conjugate gradient}$$

$$\overline{D}_{\mathrm{KL}}(\theta_{\mathrm{old}}, \theta) \approx \frac{1}{2}(\theta - \theta_{\mathrm{old}})^T A(\theta - \theta_{\mathrm{old}}) \qquad A_{ij} = \frac{\partial}{\partial \theta_i} \frac{\partial}{\partial \theta_j} \overline{D}_{\mathrm{KL}}(\theta_{\mathrm{old}}, \theta)$$

2. Compute the maximal step length

$$\delta = \overline{D}_{\mathrm{KL}} \approx \frac{1}{2}(\beta s)^T A(\beta s) = \frac{1}{2}\beta^2 s^T$$

$$\beta = \sqrt{2\delta / s^T A s}$$

Solving the Trust-Region Constrained Optimization

1. Compute a search direction, using a linear approximation to objective and quadratic approximation to the constraint

$$Ax = g \longrightarrow \text{Conjugate gradient}$$

$$\overline{D}_{\mathrm{KL}}(\theta_{\mathrm{old}}, \theta) \approx \frac{1}{2}(\theta - \theta_{\mathrm{old}})^T A(\theta - \theta_{\mathrm{old}}) \qquad A_{ij} = \frac{\partial}{\partial \theta_i}\frac{\partial}{\partial \theta_j}\overline{D}_{\mathrm{KL}}(\theta_{\mathrm{old}}, \theta)$$

2. Compute the maximal step length: $\theta + \beta s$ satisfies the KL divergence

$$\delta = \overline{D}_{\mathrm{KL}} \approx \frac{1}{2}(\beta s)^T A(\beta s) = \frac{1}{2}\beta^2 s^T$$

$$\beta = \sqrt{2\delta / s^T A s}$$

3. Line search to ensure the constraints and monotonic improvement

$$L_{\theta_{\mathrm{old}}}(\theta) - \mathcal{X}[\overline{D}_{\mathrm{KL}}(\theta_{\mathrm{old}}, \theta) \leq \delta]$$

Summary - TRPO

1. Original objective:

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

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

Summary - TRPO

1. Original objective:

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

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

2. Policy improvement in terms of advantage function:

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

Summary - TRPO

1. Original objective:

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

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

2. Policy improvement in terms of advantage function:

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

3. Surrogate loss to remove the dependency on the trajectories of new policy

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

$$L_{\pi_{\theta_0}}(\pi_{\theta_0}) = \eta(\pi_{\theta_0}), \quad \nabla_\theta L_{\pi_{\theta_0}}(\pi_\theta)\big|_{\theta=\theta_0} = \nabla_\theta \eta(\pi_\theta)\big|_{\theta=\theta_0}$$

Summary - TRPO

4. Find the lower bound (monotonic improvement guarantee)

$$L_{\pi_i}(\pi) - CD_{\mathrm{KL}}^{\max}(\pi_i, \pi)$$

$$\eta(\pi_{i+1}) \geq M_i(\pi_{i+1})$$

$$\eta(\pi_i) = M_i(\pi_i), \text{ therefore,}$$

$$\eta(\pi_{i+1}) - \eta(\pi_i) \geq M_i(\pi_{i+1}) - M(\pi_i).$$

Summary - TRPO

4. Find the lower bound (monotonic improvement guarantee)

$$L_{\pi_i}(\pi) - CD_{\mathrm{KL}}^{\max}(\pi_i, \pi)$$

$$\eta(\pi_{i+1}) \geq M_i(\pi_{i+1})$$

$$\eta(\pi_i) = M_i(\pi_i), \text{ therefore,}$$

$$\eta(\pi_{i+1}) - \eta(\pi_i) \geq M_i(\pi_{i+1}) - M(\pi_i).$$

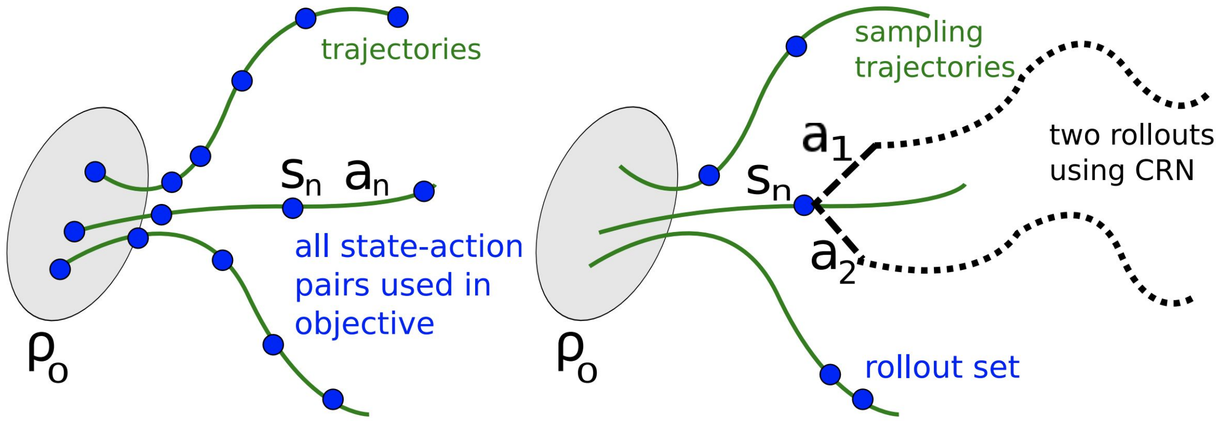
5. Solve the optimization problem using linear search (Fish matrix and conjugate gradients)

$$\underset{\theta}{\text{maximize}}\, L_{\theta_{\text{old}}}(\theta)$$

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

Experiments (TRPO)

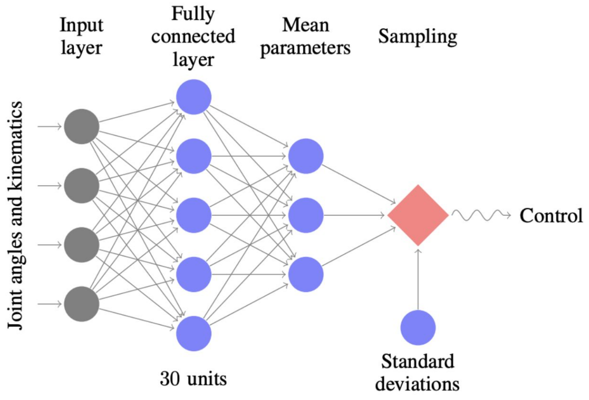
- Sample-based estimation of advantage functions
    - Single path: sample initial state $s_0 \sim \rho_0$ and generate trajectories following $\pi_{\theta_{\text{old}}}$
    - Vine: pick a “roll-out” subset and sample multiple actions and trajectories (**lower variance**)

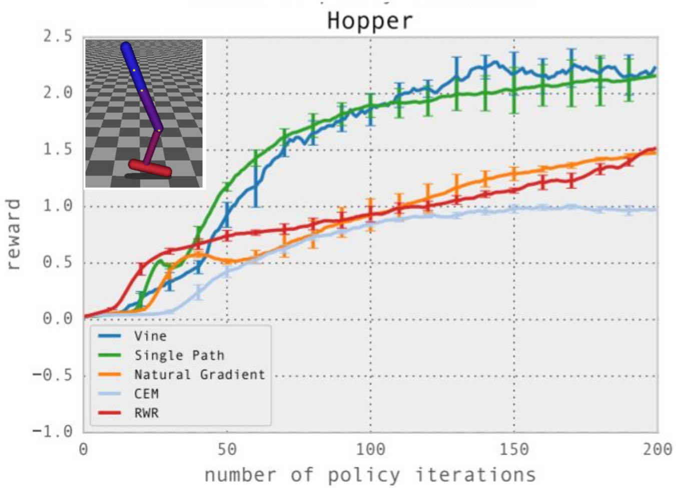


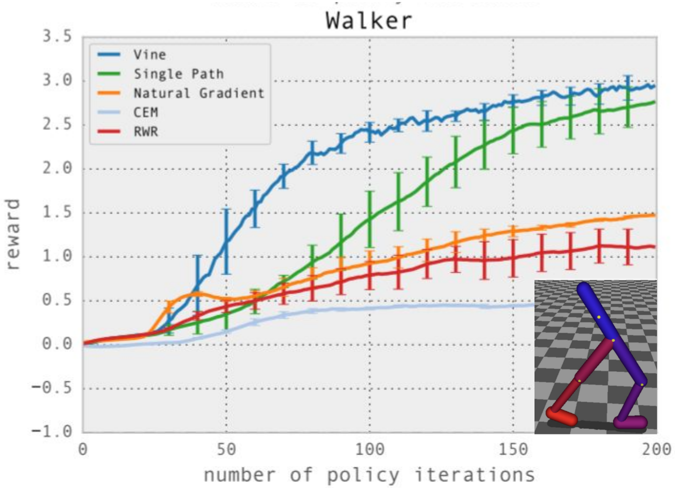
(a) Single Path (b) Vine

Experiments (TRPO)

- Simulated Robotic Locomotion tasks
  - Hopper: 12-dim state space
  - Walker: 18-dim state space
  - rewards: encourage fast and stable running (hopper); encourage smooth walke (walker)

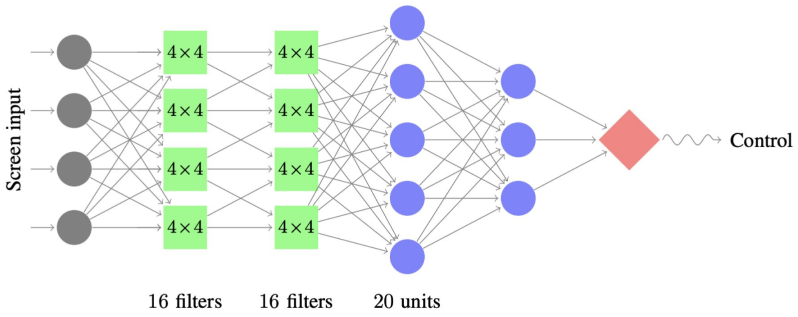


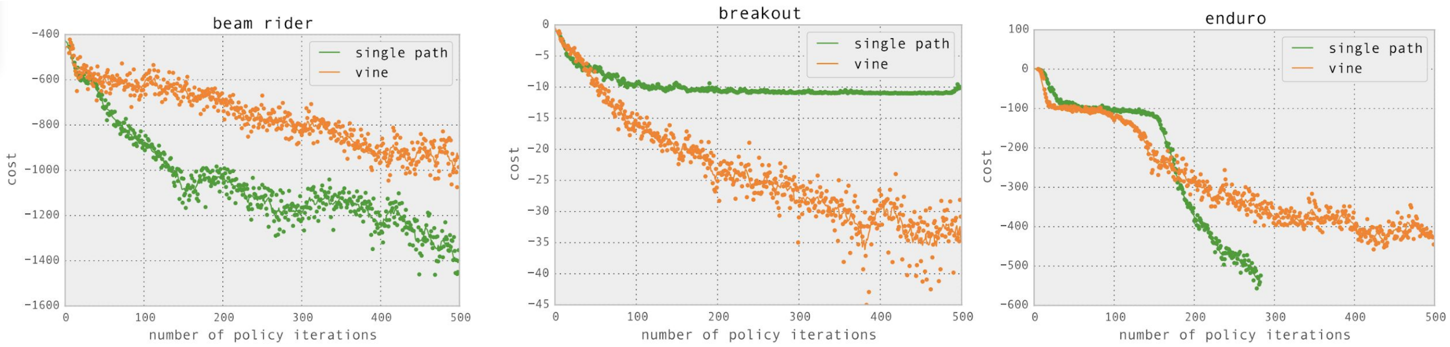




Experiments (TRPO)

- Atari games (discrete action space) - 0 / 1





	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

Limitations of TRPO

- Hard to use with architectures with multiple outputs, e.g., policy and value function (need to weight different terms in distance metric)

- Empirically performs poorly on tasks requiring deep CNNs and RNNs, e.g., Atari benchmark (more suitable for locomotion)

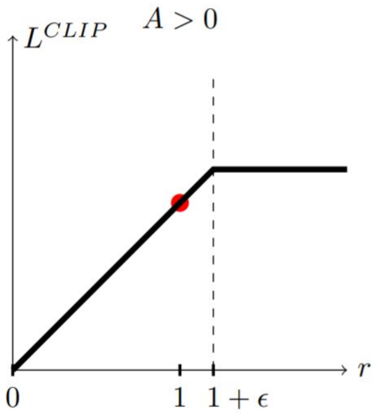
- Conjugate gradients makes implementation more complicated than SGD

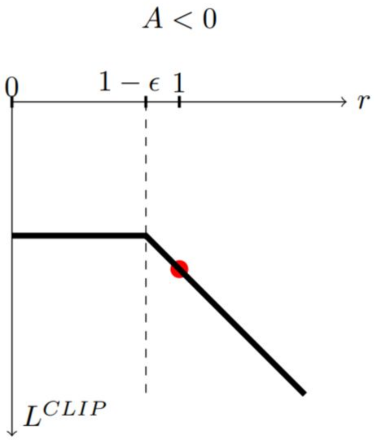
Proximal Policy Optimization (PPO)

- Clipped surrogate objective

TRPO: $$L^{CPI}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right] = \hat{\mathbb{E}}_t\left[r_t(\theta)\hat{A}_t\right]$$

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





Proximal Policy Optimization (PPO)

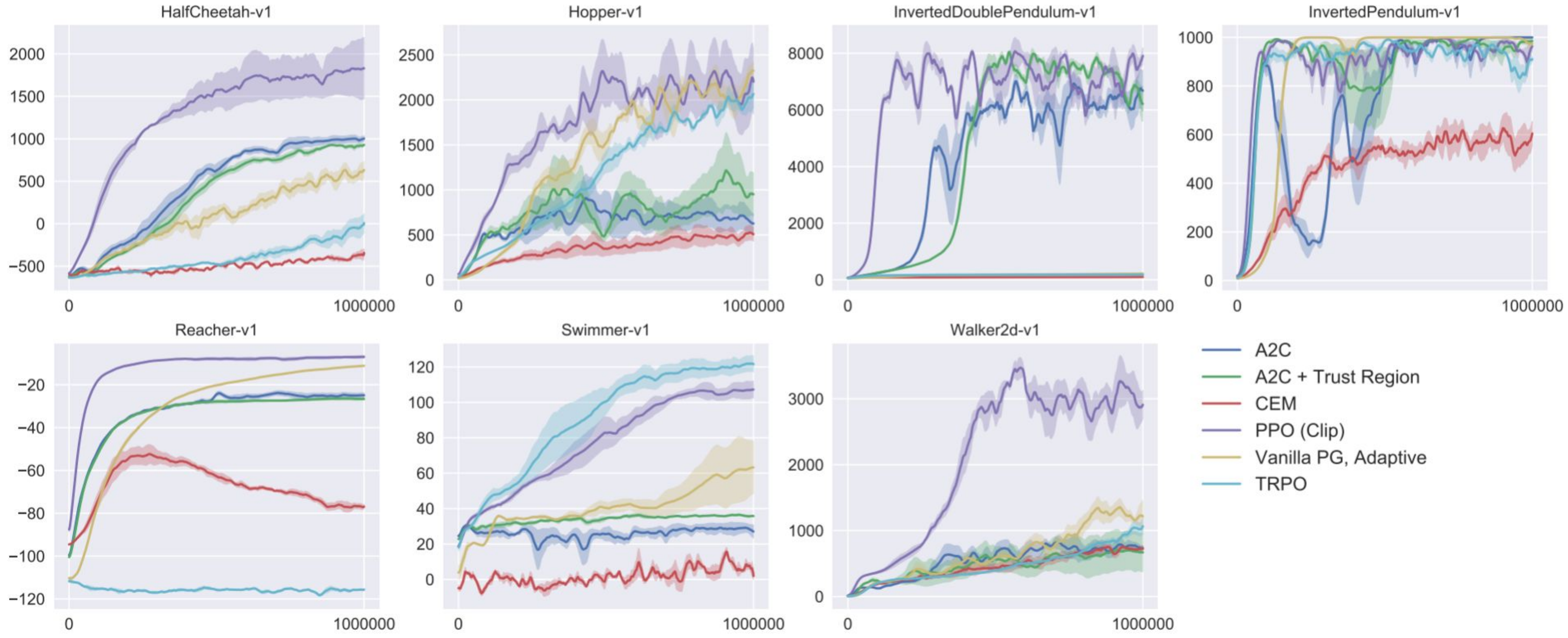
- Adaptive KL Penalty Coefficient

- Using several epochs of minibatch SGD, optimize the KL-penalized objective

$$L^{KLPEN}(\theta) = \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \hat{A}_t - \beta \, \mathrm{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)] \right]$$

- Compute $d = \hat{\mathbb{E}}_t[\mathrm{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]]$
  - If $d < d_{\text{targ}}/1.5$, $\beta \leftarrow \beta/2$
  - If $d > d_{\text{targ}} \times 1.5$, $\beta \leftarrow \beta \times 2$

Experiments (PPO)



Takeaways

- Trust region optimization guarantees the monotonic policy improvement.

- PPO is a first-order approximation of TRPO that is simpler to implement and achieves better empirical performance (both locomotion and Atari games).

Related Work

[1] S. Kakade. “A Natural Policy Gradient.” NIPS, 2001.

[2] S. Kakade and J. Langford. “Approximately optimal approximate reinforcement learning”. ICML, 2002.

[3] J. Peters and S. Schaal. “Natural actor-critic”. Neurocomputing, 2008.

[4] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel. “Trust Region Policy Optimization”. ICML, 2015.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. “Proximal Policy Optimization Algorithms”. 2017.

Questions

1. What is purpose of trust region? How we construct the trust region in TRPO

    (Hint: average KL divergence)

2. Why trust region optimization is not widely used in supervised learning?

    (Hint: i.i.d. assumption)

3. What are the differences between PPO and TRPO? Why PPO is preferred?

    (Hint: adaptive coefficient, surrogate loss function)

Reference

1. http://www.cs.toronto.edu/~tingwuwang/trpo.pdf
2. http://rll.berkeley.edu/deeprlcoursesp17/docs/lec5.pdf
3. https://medium.com/@jonathan_hui/rl-trust-region-policy-optimization-trpo-explained-a6ee04eeeee9
4. https://people.eecs.berkeley.edu/~pabbeel/nips-tutorial-policy-optimization-Schulman-Abbeel.pdf
5. https://www.depthfirstlearning.com/2018/TRPO#1-policy-gradient
6. https://cs.uwaterloo.ca/~ppoupart/teaching/cs885-spring18/slides/cs885-lecture15a.pdf
7. http://www.andrew.cmu.edu/course/10-703/slides/Lecture_NaturalPolicyGradientsTRPOPPO.pdf
8. https://towardsdatascience.com/policy-gradients-in-a-nutshell-8b72f9743c5d
9. Discretizing Continuous Action Space for On-Policy Optimization. Tang et al, ICLR 2018.
10. Trust Region Policy Optimization. Schulman et al., ICML 2015.
11. A Natural Policy Gradient. Sham Kakade., NIPS 2001.
12. Proximal Policy Optimization Algorithms. Schulman et al., 2017.
13. Variance Reduction for Policy Gradient with Action-Dependent Factorized Baselines. Wu et al., ICLR 2018.