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!

Policy Optimization
- Policy Gradient
  - A2C / A3C
  - PPO
  - TRPO
- DDPG
- TD3
- SAC

Q-Learning
- DQN
- C51
- QR-DQN
- HER

Learn the Model
- World Models
  - I2A
  - MBMF
  - MBVE

Given the Model
- AlphaZero

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:

\[
\mathbb{E}_z[f(z)\nabla_\theta \log p_\theta(z)] = \mathbb{E}_z\left[ \frac{f(z)}{p_\theta(z)} \nabla_\theta p_\theta(z) \right] = \int p_\theta(z) \frac{f(z)}{p_\theta(z)} \nabla_\theta p_\theta(z) \, dz \\
= \nabla_\theta \int f(z)p_\theta(z) \, dz = \nabla_\theta \mathbb{E}_z[f(z)]
\]

Remark: \( f(z) \) can be either differentiable and non-differentiable functions.
Policy Gradients (Preliminaries)

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

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

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

\[ \nabla_\theta \mathbb{E}_z[f(z)] = \mathbb{E}_z[f(z) \nabla_\theta \log p_\theta(z) + (b(z) \nabla_\theta \log p_\theta(z) - b(z) \nabla_\theta \log p_\theta(z))] = \mathbb{E}_z[(f(z) - b(z)) \nabla_\theta \log p_\theta(z)] + \mu_b \]

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

Policy Gradient Theorem [1]:

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

- **Expected reward**: \( \mathbb{E}_\tau [\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)] \)
- **Visitation frequency**: \( \sum_{t=0}^{\infty} \gamma^t p(s_t = s) \)
- **State-action function (Q-value)**: \( \sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'} \)

Subtract the Baseline - state-value function:

\[
\hat{A}(s_t, a_t) = \hat{Q}(s_t, a_t) - \hat{V}(s_t)
\]

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

Policy Gradient Theorem [1]:

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

- Expected reward: \( \mathbb{E}_{\tau}[\sum_{t=0}^{\infty} \gamma^{t}r(s_{t}, a_{t})] \)
- Visitation frequency: \( \sum_{t=0}^{\infty} \gamma^{t}p(s_{t} = s) \)
- State-action function (Q-value): \( \sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'} \)

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 $\pi_*$

Input: a differentiable policy parameterization $\pi(a|s, \theta)$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\theta \in \mathbb{R}^{d'}$ (e.g., to 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 $\pi(\cdot|\cdot, \theta)$
Loop for each step of the episode $t = 0, 1, \ldots, T - 1$:

$G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$
$\theta \leftarrow \theta + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t, \theta)$

$(G_t)$

How to choose the step size?
Motivation - Problem in PG

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

How to choose the step size?

too large? 1) bad policy  2) collected data under bad policy
too small? cannot leverage data sufficiently
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)**

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

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

\[
\eta(\hat{\pi}) = \eta(\pi) + \sum_s \rho_{\hat{\pi}}(s) \sum_a \hat{\pi}(a|s) A_\pi(s, a)
\]

- Local approximation (visitation frequency is unknown)

\[
L_\pi(\hat{\pi}) = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \hat{\pi}(a|s) A_\pi(s, a)
\]

\[
L_{\pi_\theta_0}(\pi_\theta_0) = \eta(\pi_\theta_0), \quad \nabla_\theta L_{\pi_\theta_0}(\pi_\theta)|_{\theta=\theta_0} = \nabla_\theta \eta(\pi_\theta)|_{\theta=\theta_0}
\]
Proof: Relation between new and old policy:

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

\[
= \mathbb{E}_{\tau \sim \pi} \left[ \sum_{t=0}^{\infty} \gamma^t A^{\pi_{\text{old}}}(s_t, a_t) \right]
\]

\[
= \mathbb{E}_{\tau \sim \pi} \left[ \sum_{t=0}^{\infty} \gamma^t (r(s_t) + \gamma V^{\pi_{\text{old}}}(s_{t+1}) - V^{\pi_{\text{old}}}(s_t)) \right]
\]

\[
= \mathbb{E}_{\tau \sim \pi} \left[ -V^{\pi_{\text{old}}}(s_0) + \sum_{t=0}^{\infty} \gamma^t r(s_t) \right]
\]

\[
= -\mathbb{E}_{s_0} [V^{\pi_{\text{old}}}(s_0)] + \mathbb{E}_{\tau \sim \pi} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t) \right]
\]

\[
= -\eta(\pi_{\text{old}}) + \eta(\pi)
\]
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)

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

\[
L_{\pi_{\theta_0}}(\pi_{\theta_0}) = \eta(\pi_{\theta_0}), \quad \nabla_\theta L_{\pi_{\theta_0}}(\pi_{\theta_0})_{|_{\theta = \theta_0}} = \nabla_\theta \eta(\pi_{\theta_0})_{|_{\theta = \theta_0}}
\]
Surrogate Loss: Important sampling Perspective

Important Sampling:

\[ \eta(\pi) = \text{const} + \mathbb{E}_{s \sim \pi, a \sim \pi} \left[ A^{\pi_{\text{old}}}(s, a) \right] \]
\[ = \text{const} + \mathbb{E}_{s \sim \pi, a \sim \pi_{\text{old}}} \left[ \frac{\pi(a | s)}{\pi_{\text{old}}(a | s)} A^{\pi_{\text{old}}}(s, a) \right] \]

\[ L(\pi) = \mathbb{E}_{s \sim \pi_{\text{old}}, a \sim \pi} \left[ A^{\pi_{\text{old}}}(s_t, a_t) \right] \]
\[ = \mathbb{E}_{s \sim \pi_{\text{old}}, a \sim \pi_{\text{old}}} \left[ \frac{\pi(a | s)}{\pi_{\text{old}}(a | s)} A^{\pi_{\text{old}}}(s, a) \right] \]

Matches to first order for parameterized policy:

\[ \nabla_\theta L(\pi_\theta) \bigg|_{\theta_{\text{old}}} = \mathbb{E}_{s, a \sim \pi_{\text{old}}} \left[ \nabla_\theta \pi_\theta(a | s) \frac{\pi_{\text{old}}(a | s)}{\pi_{\text{old}}(a | s)} A^{\pi_{\text{old}}}(s, a) \right] \bigg|_{\theta_{\text{old}}} \]
\[ = \mathbb{E}_{s, a \sim \pi_{\text{old}}} \left[ \nabla_\theta \log \pi_\theta(a | s) A^{\pi_{\text{old}}}(s, a) \right] \bigg|_{\theta_{\text{old}}} = \nabla_\theta \eta(\pi_\theta) \bigg|_{\theta=\theta_{\text{old}}} \]
Surrogate Loss: Important sampling Perspective

Important Sampling:

\[
\eta(\pi) = \text{const} + \mathbb{E}_{s \sim \pi, a \sim \pi}[A^{\pi_{\text{old}}}(s, a)]
\]

\[
= \text{const} + \mathbb{E}_{s \sim \pi, a \sim \pi_{\text{old}}}
\left[ \frac{\pi(a | s)}{\pi_{\text{old}}(a | s)} A^{\pi_{\text{old}}}(s, a) \right]
\]

\[
L(\pi) = \mathbb{E}_{s \sim \pi_{\text{old}}, a \sim \pi}[A^{\pi_{\text{old}}}(s_t, a_t)]
\]

\[
= \mathbb{E}_{s \sim \pi_{\text{old}}, a \sim \pi_{\text{old}}}
\left[ \frac{\pi(a | s)}{\pi_{\text{old}}(a | s)} A^{\pi_{\text{old}}}(s, a) \right]
\]

Matches to first order for parameterized policy:

\[
\nabla_\theta L(\pi_\theta)|_{\theta_{\text{old}}} = \mathbb{E}_{s, a \sim \pi_{\text{old}}}
\left[ \frac{\nabla_\theta \pi_\theta(a | s)}{\pi_{\text{old}}(a | s)} A^{\pi_{\text{old}}}(s, a) \right]|_{\theta_{\text{old}}}
\]

\[
= \mathbb{E}_{s, a \sim \pi_{\text{old}}}[\nabla_\theta \log \pi_\theta(a | s) A^{\pi_{\text{old}}}(s, a)]|_{\theta_{\text{old}}} = \nabla_\theta \eta(\pi_\theta)|_{\theta=\theta_{\text{old}}}
\]
Monotonic Improvement Result

- Find the lower bound in general stochastic gradient policies

\[
\eta(\tilde{\pi}) \geq L_{\pi}(\tilde{\pi}) - CD_{KL}^{\text{max}}(\pi, \tilde{\pi}),
\]

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

\[
D_{KL}^{\text{max}}(\pi, \tilde{\pi}) = \max_s D_{KL}(\pi(\cdot|s) || \tilde{\pi}(\cdot|s))
\]

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

\[
M_i(\pi) = L_{\pi_i}(\pi) - CD_{KL}^{\text{max}}(\pi_i, \pi)
\]

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

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

\[
\eta(\pi_{i+1}) - \eta(\pi_i) \geq M_i(\pi_{i+1}) - M(\pi_i).
\]
Optimization of Parameterized Policies

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

$$\max_{\theta} [L_{\theta_{\text{old}}} (\theta) - CD_{KL}^{\max} (\theta_{\text{old}}, \theta)]$$
Optimization of Parameterized Policies

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

$$\max_{\theta} \left[ L_{\theta_{old}}(\theta) - CD_{KL}^{\max}(\theta_{old}, \theta) \right]$$

- 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:

$$\max_{\theta} L_{\theta_{old}}(\theta)$$

subject to $D_{KL}^{\max}(\theta_{old}, \theta) \leq \delta$. 
Optimization of Parameterized Policies

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

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

- 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:

$$\max_{\theta} L_{\theta_{\text{old}}} (\theta) \quad \text{subject to} \quad D_{\text{KL}}^{\max} (\theta_{\text{old}}, \theta) \leq \delta.$$
Solving the Trust-Region Constrained Optimization

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

\[ Ax = g \quad \text{Conjugate gradient} \]

\[ \overline{D}_{KL}(\theta_{\text{old}}, \theta) \approx \frac{1}{2}(\theta - \theta_{\text{old}})^T A(\theta - \theta_{\text{old}}) \]

\[ A_{ij} = \frac{\partial}{\partial \theta_i} \frac{\partial}{\partial \theta_j} \overline{D}_{KL}(\theta_{\text{old}}, \theta) \]
Solving the Trust-Region Constrained Optimization

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

\[ Ax = g \quad \text{Conjugate gradient} \]

\[ D_{KL}(\theta_{old}, \theta) \approx \frac{1}{2} (\theta - \theta_{old})^T A (\theta - \theta_{old}) \quad A_{ij} = \frac{\partial}{\partial \theta_i} \frac{\partial}{\partial \theta_j} D_{KL}(\theta_{old}, \theta) \]

2. Compute the maximal step length

\[ \delta = D_{KL} \approx \frac{1}{2} (\beta s)^T A (\beta s) = \frac{1}{2} \beta^2 s^T \]

\[ \beta = \sqrt{2\delta / s^T As} \]
Solving the Trust-Region Constrained Optimization

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

\[ Ax = g \quad \text{Conjugate gradient} \]

\[ D_{KL}(\theta_{old}, \theta) \approx \frac{1}{2} (\theta - \theta_{old})^T A (\theta - \theta_{old}) \quad A_{ij} = \frac{\partial}{\partial \theta_i} \frac{\partial}{\partial \theta_j} D_{KL}(\theta_{old}, \theta) \]

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

\[ \delta = D_{KL} \approx \frac{1}{2} (\beta s)^T A (\beta s) = \frac{1}{2} \beta^2 s^T \]

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

3. Line search to ensure the constraints and monotonic improvement

\[ L_{\theta_{old}}(\theta) - \mathcal{X}[D_{KL}(\theta_{old}, \theta) \leq \delta] \]
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:

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

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

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

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

4. Find the lower bound (monotonic improvement guarantee)

\[ L_{\pi_i}(\pi) - CD_{\text{KL}}^{\max}(\pi_i, \pi) \]
\[ \eta(\pi_{i+1}) \geq M_i(\pi_{i+1}) \]
\[ \eta(\pi_i) = M_i(\pi_i), \text{ therefore,} \]
\[ \eta(\pi_{i+1}) - \eta(\pi_i) \geq M_i(\pi_{i+1}) - M(\pi_i). \]
Summary - TRPO

4. Find the lower bound (monotonic improvement guarantee)

\[ L_{\pi_i}(\pi) - CD^{\text{max}}_{KL}(\pi_i, \pi) \]
\[ \eta(\pi_{i+1}) \geq M_i(\pi_{i+1}) \]
\[ \eta(\pi_i) = M_i(\pi_i), \text{ therefore,} \]
\[ \eta(\pi_{i+1}) - \eta(\pi_i) \geq M_i(\pi_{i+1}) - M(\pi_i). \]

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

\[
\underset{\theta}{\text{maximize}} \quad L_{\theta_{\text{old}}}(\theta) \\
\text{subject to} \quad D_{KL}^{\rho_{\theta_{\text{old}}}}(\theta_{\text{old}}, \theta) \leq \delta.
\]
Experiments (TRPO)

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

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

<table>
<thead>
<tr>
<th></th>
<th>B. Rider</th>
<th>Breakout</th>
<th>Enduro</th>
<th>Pong</th>
<th>Q*bert</th>
<th>Seaquest</th>
<th>S. Invaders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>354</td>
<td>1.2</td>
<td>0</td>
<td>-20.4</td>
<td>157</td>
<td>110</td>
<td>179</td>
</tr>
<tr>
<td>Human (Mnih et al., 2013)</td>
<td>7456</td>
<td>31.0</td>
<td>368</td>
<td>-3.0</td>
<td>18900</td>
<td>28010</td>
<td>3690</td>
</tr>
<tr>
<td>Deep Q Learning (Mnih et al., 2013)</td>
<td>4092</td>
<td>168.0</td>
<td>470</td>
<td>20.0</td>
<td>1952</td>
<td>1705</td>
<td>581</td>
</tr>
<tr>
<td>UCC-I (Guo et al., 2014)</td>
<td>5702</td>
<td>380</td>
<td>741</td>
<td>21</td>
<td>20025</td>
<td>2995</td>
<td>692</td>
</tr>
<tr>
<td>TRPO - single path</td>
<td>1425.2</td>
<td>10.8</td>
<td>534.6</td>
<td>20.9</td>
<td>1973.5</td>
<td>1908.6</td>
<td>568.4</td>
</tr>
<tr>
<td>TRPO - vine</td>
<td>859.5</td>
<td>34.2</td>
<td>430.8</td>
<td>20.9</td>
<td>7732.5</td>
<td>788.4</td>
<td>450.2</td>
</tr>
</tbody>
</table>
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: 
\[
L^{CPI}(\theta) = \mathbb{E}_t \left[ \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \hat{A}_t \right] = \mathbb{E}_t \left[ r_t(\theta) \hat{A}_t \right]
\]

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

- Adaptive KL Penalty Coefficient

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

\[ L_{KL\text{PEN}}(\theta) = \hat{E}_t \left[ \frac{\pi_\theta(a_t | s_t)}{\pi_{\theta_{old}}(a_t | s_t)} \hat{A}_t - \beta \text{KL}[\pi_{\theta_{old}}(\cdot | s_t), \pi_\theta(\cdot | s_t)] \right] \]

- Compute \( d = \hat{E}_t[\text{KL}[\pi_{\theta_{old}}(\cdot | s_t), \pi_\theta(\cdot | s_t)]] \)
  - 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


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

7. http://www.andrew.cmu.edu/course/10-703/slides/Lecture_NaturalPolicyGradientsTRPO.pdf