Policy Optimization via Local Approximation

白刚

me@baigang.net

July 1, 2018

Overview

1. Backgrouds
2. A lower bound for policy improvement
3. Practical algorithms
   - Truncated natural policy gradient
   - Trust region policy optimization
   - Proximal policy optimization
   - Constraied policy optimization
   - Kronecker-factored approximation for scalability

Trust region strategy

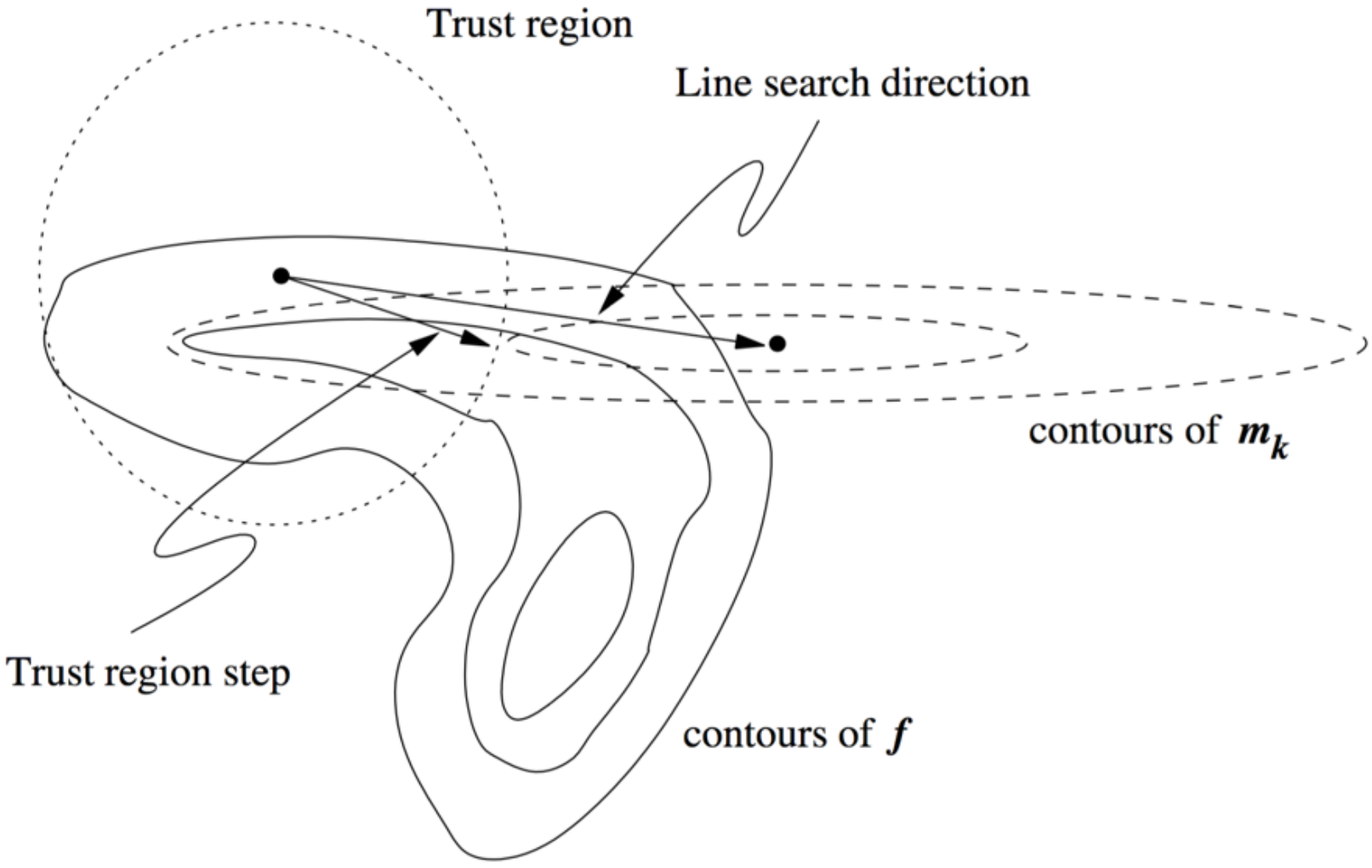
- Construct a model function $m_k(x)$
  - behavior at x_k is similar to $f(x_k)$
  - e.g. Taylor expansion:
    $m_k(x_k + p) = f(x_k) + p^T \nabla f(x_k) + \frac{1}{2} p^T H_k p$
- Find candidate step p by solving

$$\min_p \; m_k(x + p)$$

$$\text{subject to: } x_k + p \in \Delta$$

- $x_{k+1} = x_k + p$, if $x_k + p$ produces a sufficient decrease
  - Otherwise, shrink the trust region Δ and re-solve.

Trust-region step illustration



Monotonic improvement: Minorize-Maximization

- Goal: $\max_x f(x)$
- Find a surrogate function $g(x|x_k)$:
  - $g(x|x_k) \leq f(x), \forall x;\ g(x_k|x_k) = f(x_k)$
- Iteratively maximize $g(x|x_k)$ in lieu of $f(x)$
  - $f(x_{k+1}) \geq g(x_{k+1}|x_k) \geq g(x_k|x_k) = f(x_k)$
- i.e. maximizing the lower bound of the objective.

Reinforcement learning: Notations and preliminaries

- MDP: $(\mathcal{S}, \mathcal{A}, P^{s'}_{sa}, R^{a}_{s}, \gamma, \rho_0)$
- The objective: expected return of the policy
  - expected discounted cumulative reward of policy π: $\eta(\pi) = \mathbb{E}_{s_0, a_0, \cdots}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t)\right]$
- the action value function: $Q_\pi(s_t, a_t) = \mathbb{E}_{s_{t+1}, a_{t+1}, \cdots}\left[\sum_{l=1}^{\infty} \gamma^l r(s_{t+l})\right]$
- the value function: $V_\pi(s_t) = \mathbb{E}_{a_t, s_{t+1}, \cdots}\left[\sum_{l=1}^{\infty} \gamma^l r(s_{t+l})\right]$
- the advantage function: $A_\pi(s, a) = Q_\pi(s, a) - V_\pi(s)$

Reinforcement learning: Policy gradient

$$\theta^* = \arg\max_\theta \ \eta(\pi_\theta) = \arg\max_\theta \ \mathbb{E}_{s_0, a_0, \cdots \sim \pi_\theta}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$$

Optimization via gradient ascent: $\theta^{k+1} = \theta^k + \alpha \nabla_\theta \eta(\pi_{\theta^k})$

Reinforcement learning: Policy gradient

$$\theta^* = \arg\max_\theta \ \eta(\pi_\theta) = \arg\max_\theta \ \mathbb{E}_{s_0, a_0, \cdots \sim \pi_\theta}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$$

Optimization via gradient ascent: $\theta^{k+1} = \theta^k + \alpha \nabla_\theta \eta(\pi_{\theta^k})$

$$\begin{aligned}
\nabla_\theta \eta(\pi_\theta) &= \sum_s \rho_{\pi_\theta}(s) \sum_a \nabla_\theta \pi_\theta(a|s) R(s, a) \\
&= \mathbb{E}_{s \sim \rho_\theta, a \sim \pi_\theta}[\frac{1}{\pi_\theta(a|s)} \nabla_\theta \pi_\theta(a|s) R(s, a)] \\
&= \mathbb{E}_{s \sim \rho_\theta, a \sim \pi_\theta}[\nabla_\theta log\, \pi_\theta(a|s) R(s, a)]
\end{aligned}$$

where $R(s, a)$ can be $Q(s, a)$, $A(s, a)$, TD-δ, $\sum R_t$, etc.

Reinforcement learning: Policy gradient

$$\theta^* = \arg\max_\theta \ \eta(\pi_\theta) = \arg\max_\theta \ \mathbb{E}_{s_0, a_0, \cdots \sim \pi_\theta}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$$

Optimization via gradient ascent: $\theta^{k+1} = \theta^k + \alpha \nabla_\theta \eta(\pi_{\theta^k})$

$$\begin{aligned}
\nabla_\theta \eta(\pi_\theta) &= \sum_s \rho_{\pi_\theta}(s) \sum_a \nabla_\theta \pi_\theta(a|s) R(s, a) \\
&= \mathbb{E}_{s \sim \rho_\theta, a \sim \pi_\theta}[\frac{1}{\pi_\theta(a|s)} \nabla_\theta \pi_\theta(a|s) R(s, a)] \\
&= \mathbb{E}_{s \sim \rho_\theta, a \sim \pi_\theta}[\nabla_\theta log\, \pi_\theta(a|s) R(s, a)]
\end{aligned}$$

where $R(s, a)$ can be $Q(s, a)$, $A(s, a)$, TD-δ, $\sum R_t$, etc.
The problem:

- Sample efficiency: on-policy expectation
- Intricate step size: parameter space gradient

Off-policy PG via importance sampling

PG is an **on-policy expectation** estimated by executing the policy.
Importance sampling: estimating expectations using samples from a different distribution:

$$\mathop{\mathbb{E}}_{x \sim P}[f(x)] = \mathop{\mathbb{E}}_{x \sim Q}[\frac{P(x)}{Q(x)} f(x)]$$

Though less stable: *variance* $= \left(\mathbb{E}_{x \sim P}[\frac{P(x)}{Q(x)} f(x)^2] - \mathbb{E}_{x \sim P}[f(x)]^2\right)$

Off-policy PG via importance sampling

PG is an **on-policy expectation** estimated by executing the policy.
Importance sampling: estimating expectations using samples from a different distribution:

$$\mathop{\mathbb{E}}_{x\sim P}[f(x)] = \mathop{\mathbb{E}}_{x\sim Q}[\frac{P(x)}{Q(x)} f(x)]$$

Though less stable: $variance = \left(\mathbb{E}_{x\sim P}[\frac{P(x)}{Q(x)} f(x)^2] - \mathbb{E}_{x\sim P}[f(x)]^2\right)$
Off-policy PG:

$$\begin{aligned}\nabla_\theta \eta(\pi_\theta) &= \mathbb{E}_{s,a\sim\pi_\theta}[\nabla_\theta log\, \pi_\theta(a|s) R(s,a)] \\ &= \mathbb{E}_{s,a\sim\pi_{\check{\theta}}}[\frac{\pi_\theta(a|s)}{\pi_{\check{\theta}}(a|s)} \nabla_\theta log\, \pi_\theta(a|s) R(s,a)]\end{aligned}$$

Overview

1. Backgrouds
2. A lower bound for policy improvement
3. Practical algorithms
   - Truncated natural policy gradient
   - Trust region policy optimization
   - Proximal policy optimization
   - Constraied policy optimization
   - Kronecker-factored approximation for scalability

Properties of the objective

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s_0, a_0, \cdots \sim \tilde{\pi}} \Big[\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t) \Big]$$
$$= \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a)$$

- proved in "Approximated Optimal Approximate Reinforcement Learning"
- guaranteed improvement:
  $\sum_a \tilde{\pi}(a|s) A_\pi(s, a) \geq 0 \implies \pi \to \tilde{\pi}$ increases performance η
- depends on the target policy $\tilde{\pi}$, difficult to evaluate.

Performance bounds proof page skippable

Relative policy performance:

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

$$= \mathbb{E}_{s_0, a_0, \cdots \sim \tilde{\pi}} \Big[\sum_{t=0}^{\infty} \gamma^t (R(s_t, a_t, s_{t+1}) + \gamma V^\pi(s_{t+1}) - V^\pi(s_t)) \Big]$$

$$= \eta(\tilde{\pi}) + \mathbb{E}_{s_0, a_0, \cdots \sim \tilde{\pi}} \Big[\sum_{t=0}^{\infty} \gamma^{t+1} V^\pi(s_{t+1}) - \sum_{t=0}^{\infty} \gamma^t V^\pi(s_t) \Big]$$

$$= \eta(\tilde{\pi}) + \mathbb{E}_{s_0, a_0, \cdots \sim \tilde{\pi}} \Big[\sum_{t=1}^{\infty} \gamma^t V^\pi(s_t) - \sum_{t=0}^{\infty} \gamma^t V^\pi(s_t) \Big]$$

$$= \eta(\tilde{\pi}) - \mathbb{E}_{s_0, a_0, \cdots \sim \tilde{\pi}} [V^\pi(s_0)]$$

$$= \eta(\tilde{\pi}) - \eta(\pi)$$

Local approximation

Use trajectories of the old policy instead of the new:

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

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

L_π matches η to first order for any parameter $\grave{\theta}$:

$$L_{\pi_{\grave{\theta}}}(\pi_{\grave{\theta}}) = \eta(\pi_{\grave{\theta}})$$

$$\nabla_\theta L_{\pi_{\grave{\theta}}}|_{\theta=\grave{\theta}} = \nabla_\theta \eta(\pi_\theta)|_{\theta=\grave{\theta}}$$

Implying a sufficiently small step $\pi_{\grave{\theta}} \to \tilde{\pi}$ that improves $L_{\pi_{\grave{\theta}}}$ will also improve η.
The problem: what step?

Conservative policy iteration and a lower bound

"Approximated Optimal Approximate Reinforcement Learning":
Let $\pi' = \arg\max_{\pi'} L_\pi(\pi')$, define the new policy as the mixture:

$$\tilde{\pi}(a|s) = (1-\alpha)\pi(a|s) + \alpha\pi'(a|s)$$

gives a lower bound:

$$\eta(\tilde{\pi}) \geq L_\pi(\tilde{\pi}) - \frac{2\epsilon\gamma}{(1-\gamma)^2}\alpha^2$$

where $\epsilon = \max_s \left|\mathbb{E}_{a\sim\pi'}\left[A_\pi(s,a)\right]\right|$

Lower bound in general stochastic policies

In conservative policy iteration: $\tilde{\pi}(a|s) = (1-\alpha)\pi(a|s) + \alpha\pi'(a|s)$, α denotes a distance between $\tilde{\pi}$ and π.

Lower bound in general stochastic policies

In conservative policy iteration: $\tilde{\pi}(a|s) = (1-\alpha)\pi(a|s) + \alpha\pi'(a|s)$,
α denotes a distance between $\tilde{\pi}$ and π.
For general cases, let $\alpha = D_{TV}^{max}(\pi, \tilde{\pi})$, the following bound holds:

$$\eta(\tilde{\pi}) \geq L_{\pi}(\tilde{\pi}) - \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha^2,$$

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

Lower bound in general stochastic policies

In conservative policy iteration: $\tilde{\pi}(a|s) = (1-\alpha)\pi(a|s) + \alpha\pi'(a|s)$,
α denotes a distance between $\tilde{\pi}$ and π.
For general cases, let $\alpha = D_{TV}^{max}(\pi, \tilde{\pi})$, the following bound holds:

$$\eta(\tilde{\pi}) \geq L_{\pi}(\tilde{\pi}) - \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha^2,$$

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

Since $D_{TV}(p \parallel q)^2 \leq D_{KL}(p \parallel q)$, we have:

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

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

Parametrized formulation

Notation:
$\eta(\theta) = \eta(\pi_\theta)$, $L_\theta(\tilde{\theta}) = L_{\pi_\theta}(\pi_{\tilde{\theta}})$ and $D_{KL}(\theta \parallel \tilde{\theta}) = D_{KL}(\pi \parallel \pi_{\tilde{\theta}})$.
We are guaranteed to improve the true objective η by performing:

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

Parametrized formulation

Notation:
$\eta(\theta) = \eta(\pi_\theta)$, $L_\theta(\tilde{\theta}) = L_{\pi_\theta}(\pi_{\tilde{\theta}})$ and $D_{KL}(\theta \parallel \tilde{\theta}) = D_{KL}(\pi \parallel \pi_{\tilde{\theta}})$.
We are guaranteed to improve the true objective η by performing:

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

In practice:
1) C makes step size too small. $\rightarrow$ Use a constraint instead of a penalty.
2) D_{KL}^{max} is impractical to evaluate. $\rightarrow$ Use avg instead of max.
Therefore, we generate a policy update by solving:

$$\underset{\theta}{\text{maximize}}\, L_{\theta_{old}}(\theta) = \sum_s \rho_{\theta_{old}}(s) \sum_a \frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\theta_{old}}(s, a)$$

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

Overview

1. Backgrouds
2. A lower bound for policy improvement
3. Practical algorithms
   - Truncated natural policy gradient
   - Trust region policy optimization
   - Proximal policy optimization
   - Constraied policy optimization
   - Kronecker-factored approximation for scalability

Practical approximations

In terms of expectations (sample average, empirical estimates):

$$\underset{\theta}{\text{maximize}}\, L_{\theta_{old}}(\theta) = \mathbb{E}_{s\sim\rho_{\theta_{old}},a\sim\pi_{\theta_{old}}}\Big[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\theta_{old}}(s,a)\Big]$$

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

Practical approximations

In terms of expectations (sample average, empirical estimates):

$$\underset{\theta}{\text{maximize}}\, L_{\theta_{old}}(\theta) = \mathbb{E}_{s\sim\rho_{\theta_{old}},a\sim\pi_{\theta_{old}}}\Big[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\theta_{old}}(s,a)\Big]$$

$$\text{subject to } \mathbb{E}_{s\sim\rho_{\theta_{old}}}\Big[D_{KL}(\pi_{\theta_{old}}(\cdot|s) \parallel \pi_\theta(\cdot|s))\Big] \le \delta$$

1st order approx to obj: $L_{\theta_{old}}(\theta) = L_{\theta_{old}}(\theta_{old}) + g^T(\theta - \theta_{old})$

- where $g = \nabla_\theta L_{\theta_{old}}(\theta)|_{\theta=\theta_{old}}$, i.e policy gradient

2nd order approx to KL term: $D_{KL}(\theta \parallel \theta_{old}) = \frac{1}{2}(\theta - \theta_{old})^T H(\theta - \theta_{old})$

- where $H = \nabla^2_\theta \mathbb{E}_{s\sim\rho_{\theta_{old}}}[D_{KL}(\pi_{\theta_{old}}(\cdot|s) \parallel \pi_\theta(\cdot|s))]|_{\theta=\theta_{old}}$
- 0th and 1st order terms of KL are 0 at θ_{old}
- 2nd order term of $L_{\theta_{old}}$ is negligible compared to KL term
- H is positive semidefinite; 2nd order term of $L_{\theta_{old}}$ is not

Overview

1. Backgrouds
2. A lower bound for policy improvement
3. Practical algorithms
   - Truncated natural policy gradient
   - Trust region policy optimization
   - Proximal policy optimization
   - Constraied policy optimization
   - Kronecker-factored approximation for scalability

Practical algorithm: Natural policy gradient

The approximated problem:

$$\underset{\theta}{\text{maximize}}\, g^T(\theta - \theta_{old})$$

$$\text{subject to } \frac{1}{2}(\theta - \theta_{old})^T H(\theta - \theta_{old}) \leq \delta$$

By converting the constraint into a penalty, we have the analytical solution:

$$\theta^* = \theta_{old} + \sqrt{\frac{2\delta}{g^T H^{-1} g}} H^{-1} g$$

Practical algorithm: Natural policy gradient

The approximated problem:

$$\underset{\theta}{\text{maximize}}\, g^T(\theta - \theta_{old})$$

$$\text{subject to } \frac{1}{2}(\theta - \theta_{old})^T H(\theta - \theta_{old}) \leq \delta$$

By converting the constraint into a penalty, we have the analytical solution:

$$\theta^* = \theta_{old} + \sqrt{\frac{2\delta}{g^T H^{-1} g}} H^{-1} g$$

$H^{-1}g$ can be effectively computed using **conjugate gradient** algorithm, a.k.a **Truncated Natural Policy Gradient** (TNPG).
Solve $Ax = b$ by finding projection to Krylov subspace spanning $\{b, Ab, A^2 b, \cdots, A^{j-1} b\}$

Natural policy gradient algorithm

Algorithm 1 Natural Policy Gradient

Input: initial policy parameters θ_0

for $k = 0, 1, 2, \ldots$ **do**

Collect set of trajectories $\mathcal{D}_k$ on policy $\pi_k = \pi(\theta_k)$

Estimate advantages $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm

Form sample estimates for

- policy gradient $\hat{g}_k$ (using advantage estimates)
- and KL-divergence Hessian / Fisher Information Matrix $\hat{H}_k$

Compute Natural Policy Gradient update:

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

end for

Overview

Trust region policy optimization

TNPG gives a good step direction:

$$\theta^* - \theta_{old} = \sqrt{\frac{2\delta}{g^T H^{-1} g}} H^{-1} g \approx \frac{1}{\lambda} H^{-1} g$$

TRPO does a line search along this direction to guarantee improvement and enforce KL-constraint:

Algorithm 2 Line Search for TRPO

Compute proposed policy step $\Delta_k = \sqrt{\frac{2\delta}{\hat{g}_k^T \hat{H}_k^{-1} \hat{g}_k}} \hat{H}_k^{-1} \hat{g}_k$

for $j = 0, 1, 2, ..., L$ **do**
  Compute proposed update $\theta = \theta_k + \alpha^j \Delta_k$
  if $\mathcal{L}_{\theta_k}(\theta) \geq 0$ and $\bar{D}_{KL}(\theta || \theta_k) \leq \delta$ **then**
    accept the update and set $\theta_{k+1} = \theta_k + \alpha^j \Delta_k$
    break
  end if
end for

Overview

1. Backgrouds
2. A lower bound for policy improvement
3. Practical algorithms
   - Truncated natural policy gradient
   - Trust region policy optimization
   - Proximal policy optimization
   - Constraied policy optimization
   - Kronecker-factored approximation for scalability

Surrogate objectives

NPG/TRPO maximizes the surrogate objective:

$$L_{\theta_{old}}(\theta) = \mathbb{E}_{s\sim\rho_{\theta_{old}}, a\sim\pi_{\theta_{old}}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\theta_{old}}(s,a)\right]$$

Surrogate objectives

NPG/TRPO maximizes the surrogate objective:

$$L_{\theta_{old}}(\theta) = \mathbb{E}_{s\sim\rho_{\theta_{old}}, a\sim\pi_{\theta_{old}}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\theta_{old}}(s,a)\right]$$

Instead of using a constraint to prevent excessively large updates:
1) use adaptive KL penalty: larger penalty coeff for large divergence
2) clipped objective: PPO penalizes policy prob ratios away from 1

Surrogate objectives

NPG/TRPO maximizes the surrogate objective:

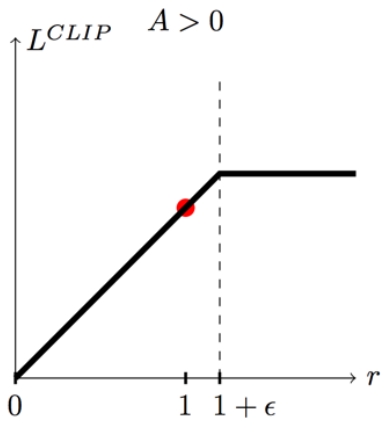
$$L_{\theta_{old}}(\theta) = \mathbb{E}_{s\sim\rho_{\theta_{old}},a\sim\pi_{\theta_{old}}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}A_{\theta_{old}}(s,a)\right]$$

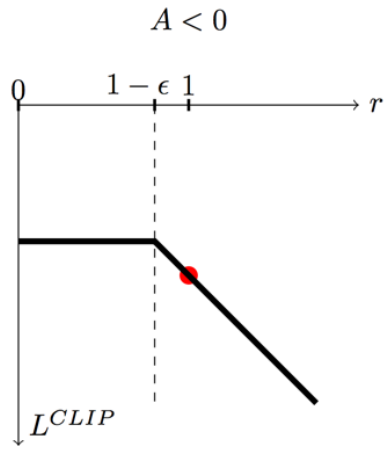
Instead of using a constraint to prevent excessively large updates:
1) use adaptive KL penalty: larger penalty coeff for large divergence
2) clipped objective: PPO penalizes policy prob ratios away from 1
In experiments, method (2) works better.

$$L^{CLIP}_{\theta_{old}}(\theta) = \mathbb{E}_{s\sim\rho_{\theta_{old}},a\sim\pi_{\theta_{old}}}\Big[$$
$$min\Big(\xi_{s,a}A_{\theta_{old}}(s,a), clip(\xi_{s,a},1-\varepsilon,1+\varepsilon)A_{\theta_{old}}(s,a)\Big)\Big]$$

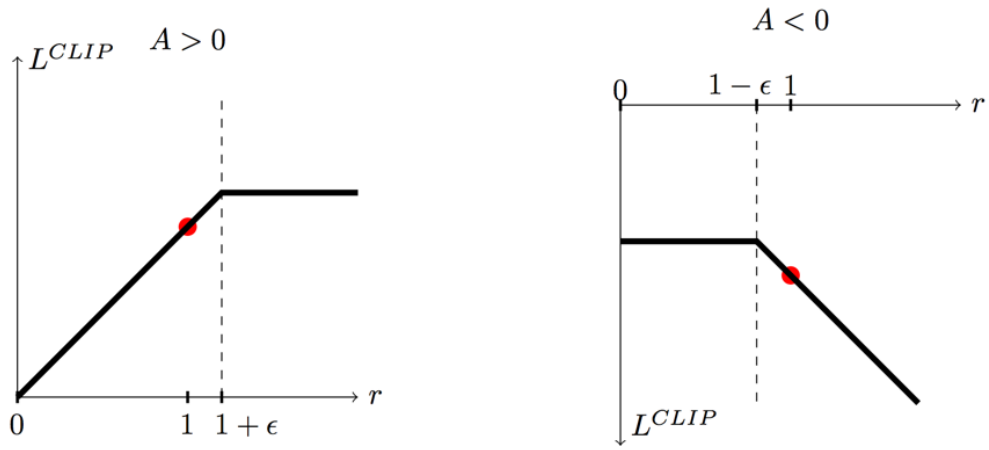
$$\text{where } \xi_{s,a} = \frac{\pi_\theta(s|a)}{\pi_{\theta_{old}}(s|a)}$$

Intepretation of CLIP





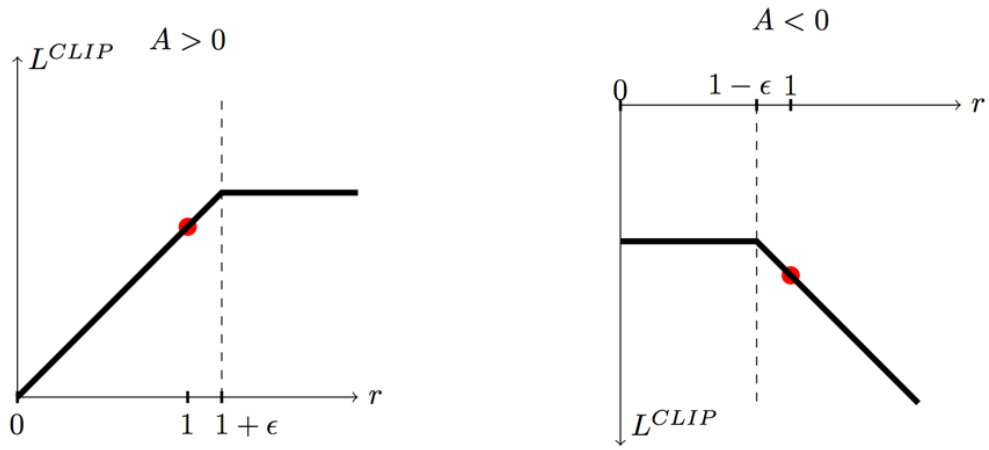
Intepretation of CLIP



Discouraging policy updates that

- over-exploiting a "good" action
- excessively avoiding a "bad" action

Intepretation of CLIP



Discouraging policy updates that

- over-exploiting a "good" action
- excessively avoiding a "bad" action

which provides stability, reliability and simplicity.

$$L^{CLIP}_{\theta_{old}}(\theta) \leq L_{\theta_{old}}(\theta) \leq \eta(\theta)$$

Extra benefits of PPO over TRPO

- easy to implement: no 2nd order/KL terms
- compatible to noise/entropy architectures
  - optimizing with dropout
  - incorporate an entropy bonus to objective
- enables parameter sharing with a value function approximator

$$L = L^{CLIP}_{\theta_{old}}(\theta) - \lambda_1 \mathbb{E}_s(V_w(s) - V^{targ}_s)^2 + \lambda_2 \mathbb{E}_s \pi_\theta(\cdot|s)^T ln \pi_\theta(\cdot|s)$$

Overview

Constrained MDP

MDP augmented with constraints that restrict the set of allowable policies.
Auxiliary cost function $C_i : \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ (similar to the usual reward).
Expected discounted constraint return: C_i:
$J_{C_i}(\pi) = \mathbb{E}_{s\sim\rho^\pi, a\sim\pi}[\sum_{t=0}^{\infty} \gamma^t C_i(s_t, a_t, s_{t+1})]$
The set of feasible stationary policies:
$\Pi_{\mathcal{C}} \doteq \{\pi \in \Pi : J_{C_i}(\pi) \leq d_i, \forall i\}$
The RL problem in a CMDP: finding the feasible optimal policy:
$\pi^* = \arg\max_{\pi\in\Pi_{\mathcal{C}}} J(\pi)$ The policy update:

$$\pi_{new} = \arg\max J(\pi)$$
$$\text{subject to } J_{C_i} \leq d_i, i = 1, \cdots, m$$
$$D(\pi_{old}, \pi_{new}) \leq \delta$$

Performance bounds

Previously used bound :

$$J(\pi') - J(\pi) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim\rho_{\pi'},a\sim\pi'}[A^{\pi}(s,a)]$$
$$\geq \frac{1}{1-\gamma}\mathbb{E}_{s\sim\rho_{\pi},a\sim\pi'}\Big[\underbrace{A^{\pi}(s,a)}_{\text{1st order match}}\Big] - C\bar{D}_{KL}^{\rho_{\pi}}(\pi \,\|\, \pi')$$

Based on this, CPO derives tight double-sided bounds to approximately solve constrained MDP:

- $J(\pi_{k+1}) \geq J(\pi_k)$
  - requires a lower bound (surrogate) $J(\pi_{k+1}) \geq L_{\pi_k}(\pi_{k+1})$
- $J_{C_i}(\pi_{k+1}) \leq d_i$
  - requires an upper bound $J_{C_i}(\pi_{k+1}) \leq L_{\pi_k}^{C_i}(\pi_{k+1})$

Tight lower/upper bounds (page skippable)

Define:

$$\delta_{V^\pi}(s,a,s') \doteq R(s,a,s') + \gamma V^\pi(s') - V^\pi(s)$$

$$\mathfrak{E}^{\pi'}_{V^\pi} \doteq \max_s \left|\mathbb{E}_{a\sim\pi'}[\delta_{V^\pi}(s,a,s')]\right|$$

$$L_{\pi,V^\pi}(\pi') \doteq \mathbb{E}_{s\sim\rho_\pi,a\sim\pi}[(\frac{\pi'(a|s)}{\pi(a|s)} - 1)\delta_{V^\pi}(s,a,s')]$$

$$D^{\pm}_{\pi,V^\pi}(\pi') \doteq \frac{L_{\pi,V^\pi}(\pi')}{1-\gamma} \pm \frac{2\gamma\mathfrak{E}^{\pi'}_{V^\pi}}{(1-\gamma)^2}\mathbb{E}_{s\sim\rho_\pi}[D_{TV}(\pi(\cdot|s) \parallel \pi'(\cdot|s))]$$

then the following bounds hold:

$$D^{+}_{\pi,V^\pi}(\pi') \geq J(\pi') - J(\pi) \geq D^{-}_{\pi,V^\pi}(\pi')$$

Tight bounds

$\epsilon^{\pi'} = \max_s |\mathbb{E}_{a\sim\pi'}[A^{\pi}(s,a)]|, \epsilon^{\pi'}_{C_i} = \max_s \mathbb{E}_{a\sim\pi'}[A^{\pi}_{C_i}(s,a)]$ Update performance bound:

$$J(\pi') - J(\pi) \geq \frac{1}{1-\gamma}\mathbb{E}_{s\sim\rho_\pi, a\sim\pi'}\left[A^{\pi}(s,a) - \frac{2\gamma\epsilon^{pi'}}{1-\gamma}\sqrt{\frac{1}{2}\bar{D}^{\rho_\pi}_{KL}(\pi_{(\cdot|s)} \parallel \pi'_{(\cdot|s)})}\right]$$

Worst-case constraint violation:

$$J_{C_i}(\pi') - J_{C_i}(\pi) \leq \frac{1}{1-\gamma}\mathbb{E}_{s\sim\rho_\pi, a\sim\pi'}\left[A^{\pi}_{C_i}(s,a) + \frac{2\gamma\epsilon^{pi'}_{C_i}}{1-\gamma}\sqrt{\frac{1}{2}\bar{D}^{\rho_\pi}_{KL}(\pi_{(\cdot|s)} \parallel \pi'_{(\cdot|s)})}\right]$$

Trust region optimization of CMDP

$$\theta_{k+1} = \arg\max_{\theta} \mathbb{E}_{s\sim\rho_{\theta_k}, a\sim\pi_\theta}[A^{\pi_{\theta_k}}(s,a)]$$

$$\text{subject to } J_{C_i}(\pi_{\theta_k}) + \frac{1}{1-\gamma}\mathbb{E}_{s\sim\rho_{\theta_k}, a\sim\pi_\theta}[A^{\pi}_{C_i}(s,a)] \leq d_i, \forall i$$

$$\bar{D}^{\rho_{\theta_k}}_{KL}(\pi_{\theta_k} \parallel \pi_\theta \leq \delta)$$

Trust region optimization of CMDP

$$\theta_{k+1} = \arg\max_\theta \mathbb{E}_{s\sim\rho_{\theta_k}, a\sim\pi_\theta}[A^{\pi_{\theta_k}}(s,a)]$$
$$\text{subject to } J_{C_i}(\pi_{\theta_k}) + \frac{1}{1-\gamma}\mathbb{E}_{s\sim\rho_{\theta_k}, a\sim\pi_\theta}[A^{\pi}_{C_i}(s,a)] \le d_i, \forall i$$
$$\bar{D}^{\rho_{\theta_k}}_{KL}(\pi_{\theta_k} \,\|\, \pi_\theta \le \delta)$$

Approximated problem formulation:

$$\theta_{k+1} = \arg\max_\theta \quad g^T(\theta - \theta_k)$$
$$\text{subject to } c_i + b_i^T(\theta - \theta_k) \le 0 \quad \forall i$$
$$\frac{1}{2}(\theta - \theta_k)^T H(\theta - \theta_k) \le \delta$$

where g is the policy gradient, b_i is pg of constraint return, $c_i \doteq J_{C_i}(\pi_k) - d_i$, H is the 2nd order term of KL – all can be evaluated via trajectory sampling.

Practical algorithm

Solvig a convex optimizaton problem. For each iteration, first solving the dual

$$\lambda^*, \nu^* = \underset{\lambda \geq 0, \nu \leq 0}{\arg\max} -\frac{1}{2\lambda}(g^T H^{-1} g - 2r^T \nu + \nu^T S \nu) + \nu^T c - \frac{\lambda\delta}{2},$$

where $r \doteq g^T H W^{-1} B$, $S \doteq B^T H^{-1} B$, $B \doteq [b1, \cdots, b_m]$.

Practical algorithm

Solvig a convex optimizaton problem. For each iteration, first solving the dual

$$\lambda^*, \nu^* = \underset{\lambda \geq 0, \nu \leq 0}{\arg\max} -\frac{1}{2\lambda}(g^T H^{-1} g - 2r^T \nu + \nu^T S \nu) + \nu^T c - \frac{\lambda\delta}{2},$$

where $r \doteq g^T H W^{-1} B$, $S \doteq B^T H^{-1} B$, $B \doteq [b1, \cdots, b_m]$.
Then the solution to the primal:

$$\theta^* = \theta_k + \frac{1}{\lambda^*} H^{-1}(g - B\nu^*)$$

Practical algorithm

Solvig a convex optimizaton problem. For each iteration, first solving the dual

$$\lambda^*, \nu^* = \underset{\lambda \geq 0, \nu \leq 0}{\arg\max} -\frac{1}{2\lambda}(g^T H^{-1} g - 2r^T \nu + \nu^T S \nu) + \nu^T c - \frac{\lambda\delta}{2},$$

where $r \doteq g^T H W^{-1} B$, $S \doteq B^T H^{-1} B$, $B \doteq [b1, \cdots, b_m]$.
Then the solution to the primal:

$$\theta^* = \theta_k + \frac{1}{\lambda^*} H^{-1}(g - B\nu^*)$$

Plus a line search to enforce constraints (violated due to approximation error, etc).
Also the problem may not be feasible due to approximation error. Do a limiting direction line search to constraint-feasible region.

Another primal-dual solution

Use iteratively updated dual variables:

$$\theta_{k+1} = \theta_k + \sigma^j \sqrt{\frac{2\delta}{(g - \nu_k b)^T H^{-1} (g - \nu_k b)}} H^{-1}(g - \nu_k b)$$

$$\nu_{k+1} = (\nu_k + \alpha(J_{C_i}(\pi_{\theta_k}) - d))_+$$

where σ^j is from the line search, α is the learning rate parameter for the dual problem.
Inferior to CPO due to:

- correctly selecting α can be chanllenging
- only guarantees constraint at convergence, may violate constraints during iterations

Overview

1. Backgrouds
2. A lower bound for policy improvement
3. Practical algorithms
   - Truncated natural policy gradient
   - Trust region policy optimization
   - Proximal policy optimization
   - Constraied policy optimization
   - Kronecker-factored approximation for scalability

Shortcomings of TNPG/TRPO's approximation

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

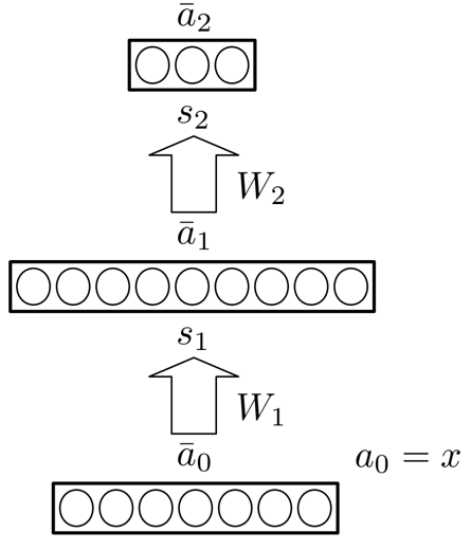
Shortcomings of TNPG/TRPO's approximation

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

- Evaluating $H^{-1}g$ using CG still requires repeated matrix-vector products.
  - Impractical for large neural networks.
- Accurately estimating g and H requires large batches of roll-outs.
  - Not sample efficient.

Fisher matrix of NN

$\theta^l \in \mathbf{R}^{C^l_{out} \times C^l_{in}}$: the i-th layer's params,
$s^l = \theta^i a^{l-1}$,
$a^l = \varphi^l(s^l)$,
$\theta = [vec(\theta^1, \cdots, \theta^L]^T$,
NN learning is essentially an MLE:



$$\theta_{ML} = \arg\max_\theta \mathbb{E}_{x,y\sim\mathbf{D}}[logPr(y|f_\theta(x))]$$

NN defines a distribution $Pr(y|x,\theta)$, and the associated Fisher information matrix:

$$\mathbf{F} = \mathbb{E}_{x,y\sim\mathbf{D}}[\nabla_\theta logPr(y|x,\theta)\, \nabla_\theta log(y|x,\theta)^T]$$

The natural gradient: $\mathbf{F}^{-1}\nabla_\theta Loss(\theta)$

Fisher matrix of NN: layer-wise

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{1,1} & \cdots & \mathbf{F}_{1,L} \\ \vdots & \ddots & \vdots \\ \mathbf{F}_{L,1} & \cdots & \mathbf{F}_{L,L} \end{bmatrix}$$

其中 $\mathbf{F}_{i,j} = \mathbb{E}[\nabla_{\theta_i} log Pr(y|x,\theta^i)\, \nabla_{\theta_j} log(y|x,\theta^j)^T]$

$$\begin{bmatrix} \mathbb{E}[\nabla_\theta log Pr(y|x,\theta^1)\, \nabla_\theta log(y|x,\theta^1)^T] & \cdots & \mathbb{E}[\nabla_\theta log Pr(y|x,\theta^1)\, \nabla_\theta log(y|x,\theta^L)^T] \\ \vdots & \ddots & \vdots \\ \mathbb{E}[\nabla_\theta log Pr(y|x,\theta^L)\, \nabla_\theta log(y|x,\theta^1)^T] & \cdots & \mathbb{E}[\nabla_\theta log Pr(y|x,\theta^1)\, \nabla_\theta log(y|x,\theta^L)^T] \end{bmatrix}$$

Practical Fisher matrix approximation

Since $\nabla_{\theta^l} log Pr(y|x,\theta^l) = \nabla_{s^l} log Pr(y|x,\theta) a^{l-1^T}$,

$$
\begin{aligned}
\mathbf{F}_{i,j} &= \mathbb{E}[vec(\nabla_\theta log Pr(y|x,\theta^i))\, vec(\nabla_\theta log Pr(y|x,\theta^j))^T] \\
&= \mathbb{E}[a^{i-1} a^{j-1^T} \otimes (\nabla_{s^i} log Pr(y|x,\theta))(\nabla_{s^j} log Pr(y|x,\theta))^T] \\
&\approx \mathbb{E}[a^{i-1} a^{j-1^T}] \otimes \mathbb{E}[(\nabla_{s^i} log Pr(y|x,\theta))(\nabla_{s^j} log Pr(y|x,\theta))^T] \\
&\doteq \mathbf{A} \otimes \mathbf{S} := \hat{\mathbf{F}}_{i,j}
\end{aligned}
$$

The 2nd order stats of the **activations** and the bp-ed **derivatives** are uncorrelated.

Practical Fisher matrix approximation

Since $\nabla_{\theta^l} log Pr(y|x,\theta^l) = \nabla_{s^l} log Pr(y|x,\theta) a^{l-1^T}$,

$$
\begin{aligned}
\mathbf{F}_{i,j} &= \mathbb{E}[vec(\nabla_\theta log Pr(y|x,\theta^i))\, vec(\nabla_\theta log Pr(y|x,\theta^j))^T] \\
&= \mathbb{E}[a^{i-1} a^{j-1^T} \otimes (\nabla_{s^i} log Pr(y|x,\theta))(\nabla_{s^j} log Pr(y|x,\theta))^T] \\
&\approx \mathbb{E}[a^{i-1} a^{j-1^T}] \otimes \mathbb{E}[(\nabla_{s^i} log Pr(y|x,\theta))(\nabla_{s^j} log Pr(y|x,\theta))^T] \\
&\doteq \mathbf{A} \otimes \mathbf{S} := \hat{\mathbf{F}}_{i,j}
\end{aligned}
$$

The 2nd order stats of the **activations** and the bp-ed **derivatives** are uncorrelated.

Since $(\mathbf{P} \otimes \mathbf{Q})^{-1} = \mathbf{P}^{-1} \otimes \mathbf{Q}^{-1}$ and $(\mathbf{P} \otimes \mathbf{Q}) vec(\mathbf{B}) = \mathbf{P}\mathbf{B}\mathbf{Q}^T$

Efficient approximation of the natual gradient:

$$
vec(\Delta\theta^l) = \underbrace{\hat{\mathbf{F}}^{l^{-1}}}_{(C^l_{out} \times C^l_{in})^2} vec(\nabla_{\theta^l} \mathcal{J}) = vec(\underbrace{\mathbf{A}^{-1}}_{C^l_{in} \times C^l_{in}} \nabla_{\theta^l} \mathcal{J} \underbrace{\mathbf{S}^{-1}}_{C^l_{out} \times C^l_{out}})
$$

Actor-Critic using Kronecker-factored trust-region

- A-C improves sample efficiency.
- Small scaled matrix with Kronecer-factored approximation of Fisher information matrix
- Critic learning with KFA
  - Jacobian matrix: $J_{i,j} = \nabla_{\theta_j} f_i(\theta)$; Gauss-Newton matrix: $G \doteq \mathbb{E}[J^T J]$
  - Gaussian matrix is equivalent to the Fisher matrix for a Gaussian observation model. $F = G$
  - Gradient is the standard TD-δ grad.
- Backgrounds and implementation
  - https://arxiv.org/abs/1412.1193
  - https://arxiv.org/abs/1503.05671
  - https://arxiv.org/abs/1708.05144 ACKTR
  - https://github.com/openai/baselines/tree/master/baselines/acktr