Theory and Methods for Reinforcement Learning

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 5: Policy Gradient 2

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)

EE-618 (Spring 2023)

License Information for Theory and Methods for Reinforcement Learning (EE-618)

- This work is released under a Creative Commons License with the following terms:
- **Attribution**
  - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees must give the original authors credit.
- **Non-Commercial**
  - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees may not use the work for commercial purposes – unless they get the licensor's permission.
- **Share Alike**
  - The licensor permits others to distribute derivative works only under a license identical to the one that governs the licensor's work.
- Full Text of the License

Recap: Policy optimization

$$\max_{\theta} J(\pi_\theta) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 \sim \mu, \pi_\theta\right] = \mathbb{E}_{s\sim\mu}[V^{\pi_\theta}(s)]$$

Tabular parametrization

- Direct:

$$\pi_\theta(a|s) = \theta_{s,a}, \text{ with } \theta_{s,a} \geq 0, \sum\nolimits_a \theta_{s,a} = 1$$

- Softmax:

$$\pi_\theta(a|s) = \frac{\exp(\theta_{s,a})}{\sum_{a'\in\mathcal{A}} \exp(\theta_{s,a'})}$$

Non-tabular parametrization

- Softmax:

$$\pi_\theta(a|s) = \frac{\exp(f_\theta(s,a))}{\sum_{a'\in\mathcal{A}} \exp(f_\theta(s,a'))}$$

- Gaussian:

$$\pi_\theta(a|s) \sim \mathcal{N}\left(\mu_\theta(s), \sigma_\theta^2(s)\right)$$

Recap: Policy gradient methods

$$\max_{\theta} J(\pi_\theta) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 \sim \mu, \pi_\theta\right] = \mathbb{E}_{s\sim\mu}[V^{\pi_\theta}(s)]$$

Exact policy gradient method

$$\theta_{t+1} \longleftarrow \theta_t + \alpha_t \nabla_\theta J(\pi_{\theta_t}),$$

where $\nabla_\theta J(\pi_{\theta_t})$ is the full gradient of the performance objective.

Stochastic policy gradient method

$$\theta_{t+1} \longleftarrow \theta_t + \alpha_t \hat{\nabla}_\theta J(\pi_{\theta_t}),$$

where $\hat{\nabla}_\theta J(\pi_{\theta_t})$ is a stochastic estimate of the full gradient of the performance objective and is used in

- REINFORCE [18]
- REINFORCE with baseline
- Actor-Critic [11]
- ...

Previous lecture

Question 1 (Non-concavity)

When do policy gradient methods converge to an optimal solution? If so, how fast?

Question 2 (Vanishing gradient)

How to avoid vanishing gradients and further improve the convergence?

Previous lecture

Question 1 (Non-concavity)

When do policy gradient methods converge to an optimal solution? If so, how fast?

Remarks: ○ Optimization wisdom: GD/SGD can converge to the global optima for "convex-like" functions:

$$J(\pi^\star) - J(\pi) = O(\|\nabla J(\pi)\|) \text{ or } O(\|G(\pi)\|)$$

○ Take-away: Despite nonconcavity, PG converges to the optimal policy, in a sublinear or linear rate.

Question 2 (Vanishing gradient)

How to avoid vanishing gradients and further improve the convergence?

Previous lecture

Question 1 (Non-concavity)

When do policy gradient methods converge to an optimal solution? If so, how fast?

Remarks: ○ Optimization wisdom: GD/SGD can converge to the global optima for "convex-like" functions:

$$J(\pi^\star) - J(\pi) = O(\|\nabla J(\pi)\|) \text{ or } O(\|G(\pi)\|)$$

○ Take-away: Despite nonconcavity, PG converges to the optimal policy, in a sublinear or linear rate.

Question 2 (Vanishing gradient)

How to avoid vanishing gradients and further improve the convergence?

Remarks: ○ Optimization wisdom: Use divergence with good curvature information.

○ Take-away: Natural policy gradient achieves a faster convergence with better constants.

This lecture

Question 3 (theory)

- Why does NPG achieve a better convergence?
- How can we further improve the algorithm?

Question 4 (practice)

- How do we extend the algorithms to function approximation settings?
- How do we extend the algorithms to online settings without computing exact gradient?
- How do we extend the algorithms to off-policy settings?

Revisit gradient descent

- Consider the optimization problem $\min_{\mathbf{x}\in\mathbb{R}^d} \ f(\mathbf{x})$.
  - Gradient descent (GD):
$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta\nabla f(\mathbf{x}_t).$$
  - Equivalent regularized form:
$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}} \left\{\nabla f(\mathbf{x}_t)^\top(\mathbf{x}-\mathbf{x}_t) + \frac{1}{2\eta}\|\mathbf{x}-\mathbf{x}_t\|_2^2\right\}.$$
  - Equivalent trust region form:
$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}} \nabla f(\mathbf{x}_t)^\top(\mathbf{x}-\mathbf{x}_t), \ \text{s.t.} \ \|\mathbf{x}-\mathbf{x}_t\|_2^2 \le \delta.$$

Question: ○ Would GD give the same trajectory under invertible linear transformations ($x \to Ay$)?

Gradient descent revisited

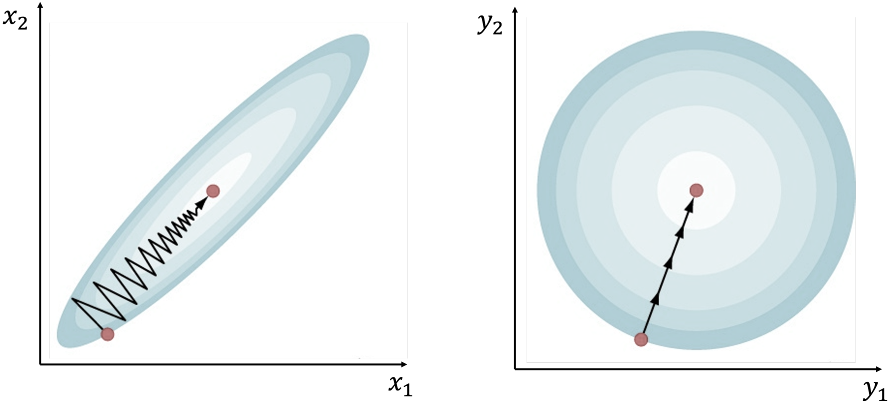


Figure: GD is not invariant w.r.t. linear transformation.

Recall Bregman divergences

Bregman divergence

Let $\omega : \mathcal{X} \to \mathbb{R}$ be continuously differentiable and 1-strongly convex w.r.t. some norm $\|\cdot\|$ on $\mathcal{X}$. The Bregman divergence D_ω associated to ω is defined as

$$D_\omega(\mathbf{x}, \mathbf{y}) = \omega(\mathbf{x}) - \omega(\mathbf{y}) - \nabla\omega(\mathbf{y})^T(\mathbf{x} - \mathbf{y}),$$

for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$.

Examples:

- ○ Euclidean distance: $\omega(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$, $D_\omega(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2$.
- ○ Mahalanobis distance: $\omega(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q \mathbf{x}$ (where $Q \succeq I$), $D_\omega(\mathbf{x}, \mathbf{y}) = \frac{1}{2}(\mathbf{x} - \mathbf{y})^T Q(\mathbf{x} - \mathbf{y})$.
- ○ Kullback-Leibler divergence: $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}_+^d : \sum_{i=1}^d x_i = 1\}$, $\omega(\mathbf{x}) = \sum_{i=1}^d x_i \log x_i$

$$D_\omega(\mathbf{x}, \mathbf{y}) = \mathrm{KL}(\mathbf{x}\|\mathbf{y}) := \sum_{i=1}^d x_i \log \frac{x_i}{y_i}.$$

Background: Mirror descent

Mirror descent (Nemirovski & Yudin, 1983)

For a given strongly convex function ω, the iterates of mirror descent [4] are given by

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \{D_\omega(\mathbf{x}, \mathbf{x}_t) + \eta_t \langle \nabla f(\mathbf{x}_t), \mathbf{x} - \mathbf{x}_t \rangle\}.$$

Examples:

- ○ Gradient descent: $\mathcal{X} \subseteq \mathbb{R}^d$, $\omega(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$, $D_\omega(\mathbf{x}, \mathbf{x}_t) = \frac{1}{2}\|\mathbf{x} - \mathbf{x}_t\|_2^2$.

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}(\mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t)).$$

- ○ Entropic mirror descent [4]: $\mathcal{X} = \Delta_d$, $\omega(\mathbf{x}) = \sum_{i=1}^d x_i \log x_i$, $D_\omega(\mathbf{x}, \mathbf{x}_t) = \mathrm{KL}(\mathbf{x}\|\mathbf{x}_t)$.

$$\mathbf{x}_{t+1} \propto \mathbf{x}_t \odot \exp(-\eta_t \nabla f(\mathbf{x}_t)),$$

where $\odot$ is element-wise multiplication and $\exp(\cdot)$ is applied element-wise.

- ○ Entropic Mirror Descent attains nearly dimension-free convergence (Chapter 4 of [5]).
- ○ See Lecture 3 Supplementary Material for more details and examples.

Background: Fisher information and KL divergence

Fisher Information Matrix

Consider a smooth parametrization of distributions $\theta \mapsto p_\theta(\cdot)$, the Fisher information matrix is defined as

$$F_\theta = \mathbb{E}_{z \sim p_\theta}[\nabla_\theta \log p_\theta(z) \nabla_\theta \log p_\theta(z)^\top].$$

Remarks:

- It is an invariant metric on the space of the parameters.
- Fisher information matrix is the Hessian of KL divergence.

$$F_{\theta_0} = \frac{\partial^2}{\partial \theta^2} \mathrm{KL}(p_{\theta_0} \| p_\theta)\Big|_{\theta=\theta_0}.$$

- The second-order Taylor expansion of KL divergence is given by

$$\mathrm{KL}(p_{\theta_0} \| p_\theta) \approx \frac{1}{2}(\theta - \theta_0)^\top F_{\theta_0} (\theta - \theta_0).$$

Background: Natural gradient descent

- Consider the optimization problem $\min_{\mathbf{x}\in\Delta} f(\mathbf{x})$ and represent $\mathbf{x}$ by $p_\theta(\cdot)$.
  - Natural gradient descent (Amari, 1998):

$$\theta_{t+1} = \theta_t - \eta(F_{\theta_t})^\dagger \nabla f(\theta_t).$$

  - Equivalent regularized form:

$$\theta_{t+1} = \arg\min_\theta \left\{ \nabla f(\theta_t)^\top (\theta - \theta_t) + \frac{1}{2\eta}(\theta - \theta_t)^\top F_{\theta_t}(\theta - \theta_t) \right\}.$$

  - Equivalent trust region form:

$$\theta_{t+1} = \arg\min_\theta \nabla f(\theta_t)^\top (\theta - \theta_t), \text{ s.t. } \frac{1}{2}(\theta - \theta_t)^\top F_{\theta_t}(\theta - \theta_t) \le \delta.$$

Natural policy gradient method for policy optimization

$$\max_{\theta} J(\pi_\theta) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 \sim \mu, \pi_\theta\right] = \mathbb{E}_{s\sim\mu}[V^{\pi_\theta}(s)]$$

Natural policy gradient (Kakade, 2002)[10]

For a stepsize $\eta > 0$, the iterates of natural policy gradient are given by

$$\theta_{t+1} = \theta_t + \eta (F_{\theta_t})^{\dagger} \nabla_\theta J(\pi_{\theta_t}).$$

Remarks:

- F_θ is the Fisher Information Matrix:

$$F_\theta = \mathbb{E}_{s\sim\lambda_\mu^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)}\left[\nabla_\theta \log \pi_\theta(a|s) \nabla_\theta \log \pi_\theta(a|s)^\top\right].$$

- $\nabla_\theta J(\pi_\theta)$ is the policy gradient:

$$\nabla_\theta J(\pi_\theta) = \frac{1}{1-\gamma} \mathbb{E}_{s\sim\lambda_\mu^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)}\left[\nabla_\theta \log \pi_\theta(a|s) A^{\pi_\theta}(s,a)\right].$$

- $C^\dagger$ is the Moore–Penrose inverse of the matrix C.

Interpretation of NPG

○ NPG can be viewed as repeatedly solving the quadratic approximation of the subproblem:

$$\theta_{t+1} \approx \arg\max_{\theta} \left\{ J(\pi_\theta), \text{ s.t. } \mathrm{KL}\left(p_{\theta_t}(\tau) \| p_\theta(\tau)\right) \leq \delta \right\},$$

where $p_\theta(\tau)$ is the probability of the random trajectory $\tau = (s_0, a_0, r_1, \ldots, \ldots)$.

Explanation:

○ Approximate the objective with the first-order Taylor expansion: $\nabla J(\pi_{\theta_t})^\top (\theta - \theta_t)$.

○ Approximate the constraint with the second-order Taylor expansion:

$$\mathrm{KL}\left(p_{\theta_t}(\tau) \| p_\theta(\tau)\right) \leq \delta \quad \rightarrow \quad \frac{1}{2}(\theta - \theta_t)^\top F_{\theta_t}(\theta - \theta_t) \leq \delta.$$

Question:

○ How can we compute the iterates of natural policy gradient efficiently?

Computing natural policy gradient

Equivalent form of NPG (Appendix C.3 [3])

Let $w^\star(\theta)$ be such that

$$(1-\gamma)(F_\theta)^\dagger \nabla_\theta J(\pi_\theta) = w^\star(\theta).$$

Then, $w^\star(\theta)$ is the solution to the following least squares minimization problem:

$$w^\star(\theta) \in \arg\min_{w} \mathbb{E}_{s\sim\lambda_\mu^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)} \left[\left(w^\top \nabla_\theta \log \pi_\theta(a|s) - A^{\pi_\theta}(s,a) \right)^2 \right]. \tag{1}$$

Remarks:

- ○ The proof follows immediately by first-order optimality condition.
- ○ Equivalently, we can rewrite NPG as:

$$\theta_{t+1} = \theta_t + \frac{\eta}{1-\gamma} w^\star(\theta_t).$$

- ○ $w^\star(\theta_t)$ can be obtained by solving (1) via conjugate gradients, SGD, and other solvers.

Side story

Compatible function approximation (Sutton et al., 1999)[16]

Let $A_{w^\star}(s,a)$ be defined as w

$$A_{w^\star}(s,a) := w^\star \cdot \nabla_\theta \log \pi_\theta(a|s)$$

where $w^\star$ is as defined in (1). Then we have

$$\nabla_\theta J(\pi_\theta) = \frac{1}{1-\gamma} F_\theta \cdot w^\star = \frac{1}{1-\gamma} \mathbb{E}_{s\sim\lambda_\mu^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)} \left[\nabla_\theta \log \pi_\theta(a|s) A_{w^\star}(s,a)\right].$$

Remarks:

- ○ One can obtain unbiased policy gradient with $A_{w^\star}(s,a)$
  - ▶ This is the best linear approximation of $A^{\pi_\theta}(s,a)$ using feature maps $\nabla \log \pi_\theta(s,a)$.
- ○ Advantage value function approximation $A^{\pi_\theta}(s,a) \approx w^\top \phi(s,a)$ can introduce bias.

Example 1: Tabular NPG under softmax parameterization

NPG parameter update

Consider the softmax parameterization $\pi_\theta(a|s) = \frac{\exp(\theta_{s,a})}{\sum_{a'} \exp(\theta_{s,a'})}$ and denote $\pi_t = \pi_{\theta_t}$, the induced NPG parameter update corresponds to the following

$$\theta_{t+1} = \theta_t + \frac{\eta}{1-\gamma} A^{\pi_t}.$$

NPG policy update = policy mirror descent

In policy space, the induced update corresponds to the following

$$\pi_{t+1}(a|s) = \pi_t(a|s) \frac{\exp(\eta A^{\pi_t}(s,a)/(1-\gamma))}{Z_t(s)}.$$

Example 2: NPG with linear function approximation

NPG parameter update

Consider $\pi_\theta(a|s) = \frac{\exp(\theta^\top \phi(s,a))}{\sum_{a'} \exp(\theta^\top \phi(s,a'))}$ and denote $\pi_t = \pi_{\theta_t}$. The induced NPG parameter update corresponds to the following

$$\theta_{t+1} = \theta_t + \frac{\eta}{1-\gamma} w_t, \text{ where } w_t = \arg\min_w \mathbb{E}_{s\sim\lambda_\mu^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)} \left[\left(w^\top \bar{\phi}(s,a) - A^{\pi_\theta}(s,a) \right)^2 \right].$$

NPG policy update = policy mirror descent

Notice that the parameterizations can be chosen to result in the familiar mirror descent updates on policies:

$$\pi_{t+1}(a|s) = \pi_t(a|s) \frac{\exp(\eta w_t^\top \phi(s,a)/(1-\gamma))}{Z_t(s)}.$$

Convergence of tabular NPG with softmax parametrization

NPG policy update = policy mirror descent

$$\pi_{t+1}(a|s) = \pi_t(a|s)\frac{\exp(\eta A^{\pi_t}(s,a)/(1-\gamma))}{Z_t(s)}$$

Convergence of tabular NPG [3]

In the tabular setting, for any $\eta \geq (1-\gamma)^2 \log|\mathcal{A}|$ and $T > 0$, the tabular NPG satisfies

$$J(\pi^\star) - J(\pi_T) \leq \frac{2}{(1-\gamma)^2 T}.$$

Remarks:

- ∘ Nearly dimension-free convergence, no dependence on $|\mathcal{A}|, |\mathcal{S}|$.
- ∘ No dependence on distribution mismatch coefficient.

Question:

- ∘ What is the computational cost of this (nearly) dimension-free method?

Proof of tabular NPG convergence

Lemma (Policy Improvement)

$$J(\pi) - J(\pi_t) = \frac{1}{\eta}\mathbb{E}_{s\sim\lambda_\mu^\pi}\left[KL(\pi(\cdot|s)\|\pi_t(\cdot|s)) - KL(\pi(\cdot|s)\|\pi_{t+1}(\cdot|s)) + \log Z_t(s)\right].$$

Proof sketch:

- ○ Recall from Performance Difference Lemma:

$$J(\pi) - J(\pi_t) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim\lambda_\mu^\pi, a\sim\pi(a|s)}[A^{\pi_t}(s,a)].$$

- ○ From the update rule $\pi_{t+1}(a|s) = \pi_t(a|s)\frac{\exp(\eta A^{\pi_t}(s,a)/(1-\gamma))}{Z_s}$, we have

$$A^{\pi_t}(s,a) = \frac{1-\gamma}{\eta}\log\frac{\pi_{t+1}(a|s)Z_t(s)}{\pi_t(a|s)}.$$

- ○ Combing these two equations, we have the above lemma.

Proof of Tabular NPG convergence

Proof.

- Setting $\pi = \pi^\star$ in the previous lemma and telescoping from $t = 0, \ldots, T-1$

$$\frac{1}{T}\sum_{t=0}^{T-1} J(\pi^\star) - J(\pi_t) \leq \frac{1}{\eta T}\mathbb{E}_{s\sim\lambda_\mu^{\pi^\star}}\left[\mathrm{KL}(\pi^\star(\cdot|s)\|\pi_0(\cdot|s))\right] + \frac{1}{\eta T}\sum_{t=0}^{T}\mathbb{E}_{s\sim\lambda_\mu^{\pi^\star}}\left[\log Z_t(s)\right].$$

- Setting $\pi = \pi_{t+1}$ in the previous lemma, we have

$$J(\pi_{t+1}) - J(\pi_t) \geq \frac{1}{\eta}\mathbb{E}_{s\sim\lambda_\mu^{\pi_{t+1}}}\left[\log Z_t(s)\right] \geq \frac{1-\gamma}{\eta}\mathbb{E}_{s\sim\mu}\left[\log Z_t(s)\right] \geq 0, \forall\mu.$$

- Combining these two equations and the fact that $J(\pi) \geq \frac{1}{1-\gamma}$ implies that

$$\frac{1}{T}\sum_{t=0}^{T-1} J(\pi^\star) - J(\pi_t) \leq \frac{\log|\mathcal{A}|}{\eta T} + \frac{1}{(1-\gamma)^2 T}.$$

□

Sample-based NPG

Sample-based NPG (informal)

- ◦ Use N-step SGD to estimate $w_t \approx w^\star(\theta_t)$
- ◦ Update $\theta_{t+1} = \theta_t + \frac{\eta}{1-\gamma} w_t$

Sample-based NPG

```
Initialize policy parameter $\theta_0 \in \mathbb{R}^d$, step size $\eta > 0$, $\alpha > 0$
for $t = 0, 1, \ldots, T-1$ do
    Initialize $w_0$, denote $\pi_t = \pi_{\theta_t}$
    for $n = 0, 1, \ldots, N-1$ do
        Obtain sample $s \sim \lambda_\mu^{\pi_t}$, $a \sim \pi_t(\cdot|s)$
        Obtain an estimate $\hat{A}(s,a)$ for $A^{\pi_t}(s,a)$
        Update $w$: $w \leftarrow w - \alpha(w^\top \nabla_\theta \log \pi_t(a|s) - \hat{A}(s,a)) \cdot \nabla_\theta \log \pi_t(a|s)$
    end for
    Set $w_t = w$ (or the average)
    Update $\theta_{t+1} = \theta_t + \frac{\eta}{1-\gamma} w_t$
end for
```

How to sample from an occupancy measure and estimate $\hat{A}(s,a)$?

Sampling routine for λ_μ^π

```
Input : a policy $\pi$.
Sample $T \sim \mathrm{Geom}(1-\gamma)$ and $s_0 \sim \mu$.
for $t = 0, 1, \ldots, T-1$ do
    Sample $a_t \sim \pi(\cdot|s_t)$.
    Sample $s_{t+1} \sim \mathsf{P}(\cdot|s_t, a_t)$.
end for
Output : $(s_T, a_T)$.
```

An estimation routine for $\hat{Q}(s,a)$

```
Input: a policy $\pi$.
Sample $(s_T, a_T) \sim \lambda_\mu^\pi$, Initialize $\hat{Q} = 0$.
while True do
    Sample $s_{T+1} \sim \mathsf{P}(\cdot|s_T, a_T)$.
    Sample $a_{T+1} \sim \pi(\cdot|s_T)$.
    Set $\hat{Q} = \hat{Q} + r_{T+1}$.
    Set $T = T + 1$.
    With probility $1-\gamma$ terminate.
end while
Output : $\hat{Q}$.
```

Remarks:

- ◦ See Algorithm 1 in [3].
- ◦ We sample from the occupancy measure by generating (s_T, a_T) with $T \sim$ Geometric$(1-\gamma)$.
- ◦ $\hat{Q}$ is an unbiased estimate of $Q(s_T, a_T)$.
- ◦ An unbiased estimate of $V(s_T)$ can be obtained with the same algorithm.
- ◦ An unbiased estimate of A can be obtained as $\hat{Q} - \hat{V}$.

How good is our approximation of $w^\star$?

- Following [3], we introduce the loss function.

$$L(w;\pi,\lambda) = \mathbb{E}_{s,a\sim\lambda}\left[\left(A^\pi(s,a) - w^T \nabla \log \pi_\theta(s,a)\right)^2\right].$$

- Its gradient is $2\mathbb{E}_{s,a\sim\lambda}\left[(A^\pi(s,a) - w^T\nabla \log \pi_\theta(s,a))\nabla \log \pi_\theta(s,a)\right]$.
- Using the estimate $\hat{A}(s,a)$, we build the stochastic gradient:

$$2\left[(\hat{A}(\hat{s},\hat{a}) - w^T\nabla \log \pi_\theta(\hat{s},\hat{a}))\nabla \log \pi_\theta(\hat{s},\hat{a})\right]$$

- $\hat{s},\hat{a}$ are sampled from $\lambda_\mu^{\pi_t}$ as described in previous slide.
- We approximate $w^\star$ with N steps of SGD and we quantify the following errors:
  - $\epsilon_{\text{stat}} = L(w_t;\pi_t,\lambda_\mu^{\pi_t}) - \min_{w:||w||\leq B} L(w;\pi_t,\lambda_\mu^{\pi_t})$.
  - $\epsilon_{\text{bias}} = \min_{w:||w||\leq B} L(w;\pi_t,\lambda_\mu^{\pi_t})$.
  - The bound B is in the order of $\frac{1}{1-\gamma}$.

Convergence of sample-based NPG with function approximation

Convergence of sampled-based NPG (informal)

$$\mathbb{E}\left[\min_{t\leq T} J(\pi_{\theta_\star}) - J(\pi_{\theta_t})\right] \leq \mathcal{O}\left(\frac{1}{1-\gamma}\sqrt{\frac{2\log|A|}{T}} + \sqrt{\kappa\epsilon_{\mathsf{stat}}} + \sqrt{\epsilon_{\mathsf{bias}}}\right),$$

where ϵ_{stat} is how close w_t is to a $w^\star(\theta_t)$ (statistical error) and ϵ_{bias} is how good the best policy in the class is (function approximation error).

Remarks:

- ◦ Using N iterations of SGD as explained in the previous slide we can get $\epsilon_{\mathsf{stat}} = \mathcal{O}(N^{-1/2})$.
- ◦ $\epsilon_{\mathsf{bias}} = 0$ under the so called "realizability" assumption for the features i.e.,

$$\forall \pi \in \Pi, \quad \exists \theta \quad \text{s.t.} \quad Q^\pi(s,a) = \theta^T\phi(s,a) \quad \forall s,a \in \mathcal{S}\times\mathcal{A}.$$

- ◦ Hence, we get

$$\mathbb{E}\left[\min_{t<T} J(\pi_{\theta_\star}) - J(\pi_{\theta_t})\right] \leq \mathcal{O}\left(\frac{1}{T^{1/2}} + \frac{1}{N^{1/4}}\right).$$

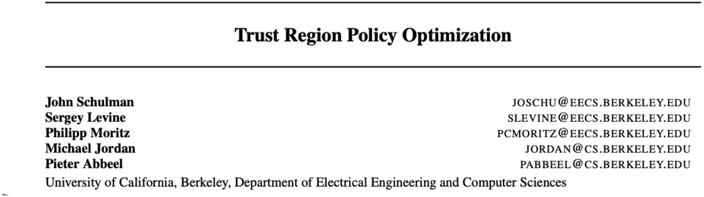
- ◦ Given that an ϵ-stationary point is achieved with $T = \mathcal{O}(\epsilon^{-2})$ and $N = \mathcal{O}(\epsilon^{-4})$.
- ◦ Total sample complexity is $\mathcal{O}(\epsilon^{-6})$.

Other remarks on the NPG bound

Remarks:

- κ quantifies how exploratory the initial distribution is.
- Requiring a bounded κ is a strong assumption.
- We discuss later how to incorporate exploration in policy gradient.
- Notice that if $\epsilon_{\text{stat}} = 0$ and $\epsilon_{\text{bias}} = 0$, we get a rate $\mathcal{O}(\sqrt{T})$.
- This is not optimal for tabular setting.
- Indeed, we have proven rate $\mathcal{O}(T^{-1})$ for Tabular NPG.
- In the tabular setting, the NPG updates attain monotonic progress.
- This is not necessarily true for general smooth policy parameterization.
- The proof in latter case is based on the no regret property of NPG.
- NPG update rule is equivalent to
  - MDP-Experts [8].
  - Mirror Descent Modified Policy Iteration [9].

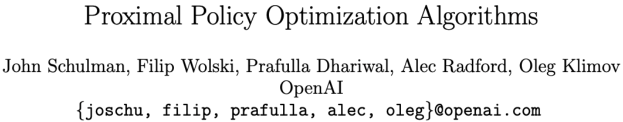
Popular Baselines



Trust Region Policy Optimization

John Schulman JOSCHU@EECS.BERKELEY.EDU
Sergey Levine SLEVINE@EECS.BERKELEY.EDU
Philipp Moritz PCMORITZ@EECS.BERKELEY.EDU
Michael Jordan JORDAN@CS.BERKELEY.EDU
Pieter Abbeel PABBEEL@CS.BERKELEY.EDU
University of California, Berkeley, Department of Electrical Engineering and Computer Sciences

TRPO (ICML, 2015)



Proximal Policy Optimization Algorithms

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov
OpenAI
`{joschu, filip, prafulla, alec, oleg}@openai.com`

PPO (arXiv, 2017)

OpenAI implementation: `https://github.com/openai/baselines`

Trust Region Policy Optimization (TRPO)

TRPO (key idea) [14]

$$\begin{aligned} \max_{\theta} \quad & \mathbb{E}_{s\sim\lambda_{\mu}^{\pi_{\theta_t}}, a\sim\pi_{\theta_t}(\cdot|s)}\left[\frac{\pi_\theta(a \mid s)}{\pi_{\theta_t}(a \mid s)} A^{\pi_{\theta_t}}(s,a)\right], \\ \text{s.t.} \quad & \mathbb{E}_{s\sim\lambda_{\mu}^{\pi_{\theta_t}}}[\mathrm{KL}(\pi_\theta(\cdot \mid s)\|\pi_{\theta_t}(\cdot \mid s))] \leq \delta. \end{aligned}$$

Remarks:

○ The surrogate objective can be viewed as linear approximation in π of $J(\pi_\theta)$:

$$J(\pi) = J(\pi_t) + \frac{1}{1-\gamma}\mathbb{E}_{s\sim\lambda_{\mu}^{\pi}, a\sim\pi(a|s)}[A^{\pi_t}(s,a)]. \qquad \text{(PDL)}$$

○ It can be approximated by a natural policy gradient step.

○ Line-search can ensure performance improvement and no constraint violation.

TRPO: A detailed look at the implementation.

- We explain how TRPO is implemented in the widely used OpenAI library [1].
- We consider a first order approximation of the objective.

$$\mathbb{E}_{s\sim\lambda_\mu^{\pi_{\theta_t}}, a\sim\pi_{\theta_t}(\cdot|s)}\left[\frac{\pi_\theta(a \mid s)}{\pi_{\theta_t}(a \mid s)} A^{\pi_{\theta_t}}(s,a)\right] \approx \langle \nabla_\theta J(\theta_k), \theta - \theta_k \rangle$$

- And a second order expansion of the constraints

$$\mathbb{E}_{s\sim\lambda_\mu^{\pi_{\theta_t}}}[\mathrm{KL}(\pi_\theta(\cdot \mid s) \| \pi_{\theta_t}(\cdot \mid s))] \approx (\theta - \theta_k)^T F(\theta_k)(\theta - \theta_k)$$

- Therefore, the practical implementation (almost) boils down to natural policy gradient.
- The subtle difference is that TRPO executes line seach along the direction $F(\theta_k)^\dagger \nabla_\theta J(\theta_k)$.
- The goal is the select the largest possible step size η such that

$$x_{t+1} = x_t + \eta F(\theta_k)^\dagger \nabla_\theta J(\theta_k)$$

satisfies the original constraints.

Equivalence between MDP-E [8] and TRPO

- The previous result proves that TRPO produces a monotonically improving sequence of policies.
- We can prove a stronger result noticing that TRPO is equivalent to MDP-E [8, 13].

MDP-Experts (MDP-E)

```
Initialize policy $\pi_0$, learning rate $\eta$
for $t = 0, 1, \ldots, T-1$ do
    Evaluate $Q^{\pi_t}(s,a)$ for every state action pair.
    $\pi_{t+1}(a|s) \propto \pi_t(a|s) \exp \eta Q^{\pi_t}(s,a)$.
end for
Output : A policy sampled uniformly at random from the sequence $\pi_0, \ldots, \pi_{T-1}$.
```

Remarks:

- Check out the course Online Learning in Games!
- MDP-E is a no-regret algorithm for adversarially changing rewards.
- Therefore, it converges to the optimal policy for a fixed reward.

Equivalence between MD-MPI [9] and TRPO

- TRPO has been interpreted also as a regularized dynamic programming scheme.
- Indeed, it falls under the general template of Mirror descent modified policy iteration (MD-MPI)

MD-MPI

$$\begin{cases} \pi_{t+1}(\cdot|s) = \arg\max_{\pi} \left[r(s,a) + \gamma \sum_{s'} P(s'|s,a) V_t(s') \right] + KL(\pi(\cdot|s)||\pi_t(\cdot|s)) \\ V_{t+1}(s) = \left(\mathcal{T}^{\pi_t}\right)^m V_t(s) \end{cases}$$

- Recall that $\mathcal{T}^{\pi_t}$ is the Bellman expectation operator associated to the policy π_t.
- TRPO is recovered with $m = \infty$.
- Many other algorithms can be captured under this framework [9].

Generalizing TRPO via MD-MPI

- Generalizations can be achieved using any other Bregman divergence.
- When the dynamics are not known we consider errors ϵ_t and ϵ'_t in both steps.
- ϵ_t and ϵ'_t are vectors of dimension $\mathcal{S}$ quantifying the error at each state.

$$\begin{cases} \pi_{t+1}(\cdot|s) \quad s.t. \quad J_t(\pi_t^\star(\cdot|s)) - J_t(\pi_{t+1}(\cdot|s)) \leq \epsilon'_t \\ V_{t+1}(s) = (\mathcal{T}^{\pi_t})^m V_t(s) + \epsilon_t \end{cases}$$

with

$$J_t(\pi(\cdot|s)) = \left[r(s,a) + \gamma \sum_{s'} P(s'|s,a) V_t(s') \right] - D(\pi(\cdot|s)||\pi_t(\cdot|s))$$

$$\pi_t^\star(\cdot|s) = \arg\max_\pi J_t(\pi(\cdot|s))$$

- Generalization of TRPO can be achieved changing m.
- A further generalization is to consider regularization also in the evaluation step.

TRPO error propagation via MD-MPI

- It is possible to prove that the errors accumulate in additive way.

Theorem

After T iterate of MD-MPI, the suboptimality of the value function is bounded as

$$\max_s V^{\pi_T}(s) - V^{\pi^\star}(s) \le \mathcal{O}\left(\frac{1}{(1-\gamma)^2}\left(\sum_{t=1}^{T}\|\epsilon_t\|_\infty + \sum_{t=1}^{T}\left\|\epsilon_t'\right\|_\infty\right) + \frac{1}{(1-\gamma)^2 T}\right)$$

- The error propagation can be improved adding the effect of Bregman divergence also in the evaluation step.
- To this end, define the Bregman regularized Bellman operator associated to the divergence D_w.

$$(T_w^\pi V)(s) = \underbrace{\mathbb{E}_{a\sim\pi(\cdot|s)}\left[r(s,a) + \gamma\sum_{s'}\mathsf{P}(s'|s,a)V(s')\right]}_{T^\pi} - w(\pi(\cdot|s))$$

MD-MPI with Bregman regularized evaluation

- Using the Bregman regularized Bellman operator, we can define the following new scheme [9, 17]

$$\begin{cases} \pi_{t+1}(\cdot|s) \quad s.t. \quad J_t(\pi_t^\star(\cdot|s)) - J_t(\pi_{t+1}(\cdot|s)) \le \epsilon_t' \\ V_{t+1}(s) = (\mathcal{T}_w^{\pi_t})^m\, V_t(s) + \epsilon_t \end{cases}$$

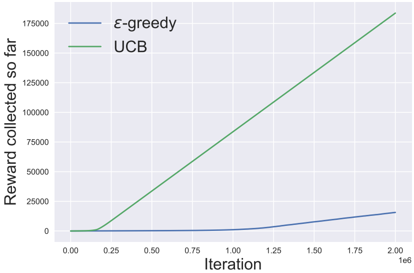
Theorem

Take $m = 1$ and $w(\pi(\cdot|s)) =< \pi(\cdot|s), \log \pi(\cdot|s) >$ and assume $\epsilon_t' = 0$ for all t, then after T iterate of the scheme above we obtain

$$\max_s V^\star(s) - V^{\pi_T}(s) \le \mathcal{O}\left(\frac{1}{1-\gamma} \left\| \frac{1}{T} \sum_{t=1}^{T} \epsilon_t \right\|_\infty + \frac{1}{(1-\gamma)T} \right)$$

- Better dependence on the effective horizon $(1-\gamma)^{-1}$.
- Better error propagation, notice that by Jensen $\left\| \frac{1}{T} \sum_{t=1}^{T} \epsilon_t \right\| \le \frac{1}{T} \sum_{t=1}^{T} \|\epsilon_t\|$.

Exploration in Policy Gradient methods



- When the transition dynamics of the agent are unknown the agent needs to explore the state space.
- Unless the initial state distribution is exploratory enough to guarantee κ small.
- Recall that κ is a constant appearing in the bound for sample based NPG.
- Recall, in the first coding exercise that we studied two different exploration techniques (ϵ-greedy and UCB).
- Can we incorporate these techniques in policy gradient ?

Provable exploration in policy gradient

- We present OPPO [6] that incorporates exploration in policy gradient for finite horizon MDP.
- The idea is to perform TRPO updates with *optimistic* estimates of the value function.
- We look at the finite horizon case because in the infinite horizon case it is more difficult to perform exploration.
- The main difficulty is the *bonus* design for the infinite horizon.

Theorem

Let $\pi_1, \pi_2, \ldots, \pi_T$ *the sequence of policies generated by OPPO. Then it holds that*

$$\sum_{t=1}^{T} V^{\star}(s_1) - V^{\pi_t}(s_1) \leq \mathcal{O}\left(\sqrt{T}\right)$$

- *This holds also when the reward function can change adversarially from episode to episode.*

Key idea : Optimism

- Optimism means to overestimate the value of $Q^{\pi_t}(s,a)$ at every state action pairs.
- Formally, it means that $Q_h(s,a)$ satisfies

$$V_h^t(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}[Q_h^t(s,a)]$$

$$Q_h^t(s,a) \geq r_h^t(s,a) + \sum_{s'} \mathsf{P}(s'|s,a)V_h^t(s') \qquad \text{(Optimism)}$$

- Notice that $Q^{\pi_t}(s,a)$ would be the fixed point of the second expression.
- At the same time we need an estimate that is not too optimistic.

$$r_h^t(s,a) + \sum_{s'} \mathsf{P}(s'|s,a)V_h^t(s') + 2b_h^t(s,a) \geq Q_h^t(s,a) \qquad \text{(Bounded Optimism)}$$

- $b_h^t(s,a)$ needs to be decreasing with the number of visits for (s,a).
- This ensures that $Q_h^t(s,a) \to Q_h^{\pi_t}(s,a)$

Estimate transition and bonuses

- In order to do this, we compute the empirical average of the transition dynamics.
- We set the function $b_h^t(s,a)$ proportional to the square root of the inverse number of visits for s,a.
- **Intuition** The more often we visit a state, the more we expect the uncertainty to reduce.

Estimating transitions and bonuses

```
for $t = 0, 1, \dots, T-1$ do
  for $h = 0, 1, \dots, H-1$ do
    Visit the state action pair $(s_h^t, a_h^t)$ and next state $s_{h+1}^t$.
    Update counts $N_h(s_h^t, a_h^t, s_{h+1}^t) \leftarrow N_h(s_h^t, a_h^t, s_{h+1}^t) + 1$, $N(s_h^t, a_h^t) \leftarrow N(s_h^t, a_h^t) + 1$.
    Estimate transtion $\hat{P}_h(s'|s,a) = \frac{N_h(s,a,s')}{N_h(s,a)+1}$ for all $s, a, s'$.
    Compute exploration bonuses $b_h(s,a) \approx \sqrt{\frac{1}{N(s_h^t,a_h^t)}}$.
  end for
end for
```

Estimate optimistic value function

- Having estimated $\hat{P}_h(s'|s,a)$ and the bonus $b_h^t(s,a)$, we can compute $Q_h^t(s,a)$ as follows.

Backward induction to estimate Q^t.

```
Initialize $Q^t_{H+1}(s,a) = 0$.
for $h = H, \ldots, 1$ do
    Recurse backward to compute $Q^t_h$
        $$Q_h^t(s,a) = r_h^t(s,a) + b_h^t(s,a) + \sum_{s',a'} \hat{P}_h(s'|s,a)\pi_{h+1}(a'|s')Q^t_{h+1}(s',a')$$
        $$Q_h^t(s,a) = \mathrm{clip}(Q_h^t(s,a); 0, H-h+1)$$
end for
```

- If it holds that $\left|\sum_{s'}(\hat{P}_h(s'|s,a) - P_h(s'|s,a))V(s')\right| \leq b_h(s,a)$.
- This construction ensures that Optimism and Bounded Optimism hold.

The complete algorithm.

○ The overall algorithm resambles MDP-E but with an optimistic evaluation step.

OPPO [6] (simplified version)

```
Initialize policy parameter $\theta_0 \in \mathbb{R}^d$, step size $\eta > 0$, $\alpha > 0$
for $t = 0, 1, \dots, T-1$ do
    Policy Evaluation
    Estimate bonus and transitions $b_h(s,a)$ and $\hat{P}_h(s'|s,a)$
    Compute optimistic value functions $Q^t_h$
    Policy Improvement
    Update policies at every $h, s, a$ with a TRPO step
        $\pi_{t+1,h}(a|s) \propto \pi_{t,h}(a|s) \exp \eta Q_h(s,a)$
end for
```

○ To prove the regret bound one proves that $\mathrm{Regret(T)} \leq \mathcal{O}\left(\sum_{h=1}^H \sum_{t=1}^T b^t_h(s^t_h, a^t_h)\right)$.

○ Next, one shows that $\sum_{h=1}^H \sum_{t=1}^T b^t_h(s^t_h, a^t_h) \leq \mathcal{O}(\sqrt{T})$.

PC-PG: Policy Cover Guided Policy Gradient [2]

- It is a more involved exploration technique suitable for infinite horizon problem.
- The intuition is to generate a favourable exploratory distribution to make the mismatch coefficients small.
- Indeed, recall that from the gradient dominance theorem, we have

$$
\begin{aligned}
J(\pi^\star) - J(\pi_\theta) &\leq \left\| \frac{\lambda_\mu^{\pi^\star}}{\lambda_\mu^{\pi}} \right\|_\infty \times \max_{\bar{\pi} \in \Delta} \langle \bar{\pi} - \pi, \nabla J(\pi) \rangle \\
&\leq \frac{1}{1-\gamma} \left\| \frac{\lambda_\mu^{\pi^\star}}{\mu} \right\|_\infty \times \max_{\bar{\pi} \in \Delta} \langle \bar{\pi} - \pi, \nabla J(\pi) \rangle
\end{aligned}
$$

- Notice that if the initial distribution was exploratory, the mismatch coefficients would be small.
- For example, if μ is uniform then $\left\| \frac{\lambda_\mu^{\pi^\star}}{\mu} \right\|_\infty \leq |\mathcal{S}|$.
- The idea behind PC-PG is to construct a "virtual" initial distribution that ensures exploration.

PC-PG's main idea

- The idea is to learn a collection of policies $(\pi_i)_{i=1}^N$ such that.

$$\rho(s,a) = \frac{1}{N}\sum_{i=1}^{N} \lambda_N^{\pi_i}(s,a) \geq \beta \quad \forall s,a.$$

- To this end, at every iteration, define the bonuses

$$b(s,a) = \frac{1}{1-\gamma} \cdot \mathbf{1}\left\{\hat{\rho}(s,a) \leq \beta\right\}$$

- $\hat{\rho}$ is an empirical estimate of the policy cover.
- At this point, we can apply Sample Based NPG with ρ as initial distribution.
- We modify the reward to encourage the extension of the cover support.

The PC-PG algorithm

PC-PG [2] (simplified version)

```
for $t = 1, \dots, T$ do
    Policy Evaluation
    Sample $M$ tuples $\mathcal{D} = \{s_i, a_i\}_{i=1}^{M}$.
    Estimate $\hat{Q}(s_i, a_i)$ for any $(s_i, a_i) \in \mathcal{D}$.
    Update counts $N(s_i, a_i) = N(s_i, a_i) + 1$ for all $(s_i, a_i) \in \mathcal{D}$.
    Estimate $\hat{\rho}(s, a) \propto N(s, a)$.
    Compute bonus.
        $$b(s,a) = \frac{1}{1-\gamma} \cdot \mathbf{1}\left\{\hat{\rho}(s,a) \leq \beta\right\}$$
    Policy Improvement
        $$\pi_{t+1}(a|s) \propto \pi_t(a|s) \exp \eta \left(\hat{Q}(s,a) + b(s,a)\right)$$
end for
```

- The sample complexity no longer scales with the concentrability coefficients.
- However, the dependence on ϵ deteriorates to $\mathcal{O}(\epsilon^{-11})$.

A closer look to baselines

- We mentioned that the baselines are used as a variance reduction mechanism.
- Actually, one can prove which choice for the baseline guarantees minimum variance.

Theorem

Consider the gradient with baseline $\widehat{\nabla}_\theta J(\pi_\theta) = \sum_{t=1}^{\infty} \left(Q^{\pi_\theta}(s_t, a_t) - b(s_t)\right) \nabla \log \pi_\theta(a_t|s_t)$ *for a trajectory* $\tau \sim p_\theta$.

Then, $b^\star(s) = \arg\min_{b:\mathcal{S}\to\mathbb{R}} \left[\mathrm{Var}\left[\widehat{\nabla}_\theta J(\pi_\theta)|s\right]\right]$ *satisfies*

$$b^\star(s) = \frac{\|Q^{\pi_\theta}(s,a)\log \pi_\theta(a|s)\|}{\|\nabla \log \pi_\theta(a|s)\|}.$$

Proof.

Start noticing that

$$\begin{aligned}\mathrm{Var}\left[\widehat{\nabla}_\theta J(\pi_\theta)|s\right] &= \mathbb{E}\left[\left\|\widehat{\nabla}_\theta J(\pi_\theta)\right\|^2 |s\right] - \left\|\mathbb{E}\left[\widehat{\nabla}_\theta J(\pi_\theta)|s\right]\right\|^2 \\ &= \mathbb{E}\left[\left\|\widehat{\nabla}_\theta J(\pi_\theta)\right\|^2 |s\right] - \left\|\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[Q^{\pi_\theta}(s,a)\nabla\log\pi_\theta(a|s)\right]\right\|^2\end{aligned}$$

Therefore $\nabla_b \mathrm{Var}\left[\widehat{\nabla}_\theta J(\pi_\theta)|s\right] = \nabla_b \mathbb{E}\left[\left\|\widehat{\nabla}_\theta J(\pi_\theta)\right\|^2 |s\right]$. Developing the norm squared and differentianting, we get

$$\nabla_b \mathbb{E}\left[\left\|\widehat{\nabla}_\theta J(\pi_\theta)\right\|^2 |s\right] = 2\left(b(s)\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\|\nabla\log\pi_\theta(a|s)\|^2\right] - \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[Q^{\pi_\theta}(s,a)\|\nabla\log\pi_\theta(a|s)\|^2\right]\right)$$

Therefore, the proof is concluded setting $b^\star$ to minimize the latter expression. □

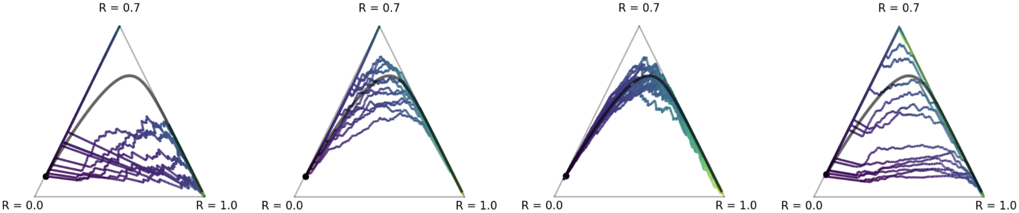
Is it always good to minimize variance ?

- The answer is no. Because, reducing the variance of the baseline can hinder exploration.
- As a result, the minimum variance baseline may lead to a suboptimal policy.
- Here we describe the result in [7].

Theorem

Theorem 1 in [7] There exists a three-arm bandit where using the stochastic natural gradient on a softmax parameterized policy with the minimum-variance baseline can lead to convergence to a suboptimal policy with positive probability, and there is a different baseline (with larger variance) which results in convergence to the optimal policy with probability 1.

Explore the baseline effect



(a) $b_\theta^- = b_\theta^* - {}^1\!/_2$

(b) $b_\theta = b_\theta^*$

(c) $b_\theta^+ = b_\theta^* + {}^1\!/_2$

(d) $b_\theta = V^{\pi_\theta}$

- The optimal policy plays the action in right corner.
- That is where the trajectories with baselines b_θ^+ and V^{π_θ} converge to .
- In the other cases, there are some trajectories converging to the top corner.
- These results confirm the issue with the minimum variance baseline.

Unbounded variance case. [12]

- Consider a bandit experiment with stochastic rewards with an action dependent distribution $R(a)$.
- A common unbiased estimator is constructed using importance sampling.
- Using an action $\hat{a} \sim \pi$ and observe $r \sim R(\hat{a})$.

$$\hat{r}(a) = \frac{r}{\pi(a)} \mathbf{1}(a = \hat{a})$$

- If we consider an additional baselines, we get the estimator

$$\hat{r}(a) = \frac{r - b}{\pi(a)} \mathbf{1}(a = \hat{a})$$

- The variance is unbounded no matter how b is chosen.

Proximal Policy Optimization (PPO2)

PPO (key idea) [15]

$$\max_{\theta} \quad \mathbb{E}_{s' \sim \lambda_\mu^{\pi_{\theta_t}}, a \sim \pi_{\theta_t}(\cdot|s)} \min\left\{ \frac{\pi_\theta(a|s)}{\pi_{\theta_t}(a|s)} A^{\pi_{\theta_t}}(s,a), \text{clip}\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_t}(a|s)}; 1-\epsilon; 1+\epsilon \right) A^{\pi_{\theta_t}}(s,a) \right\}$$

Remarks:

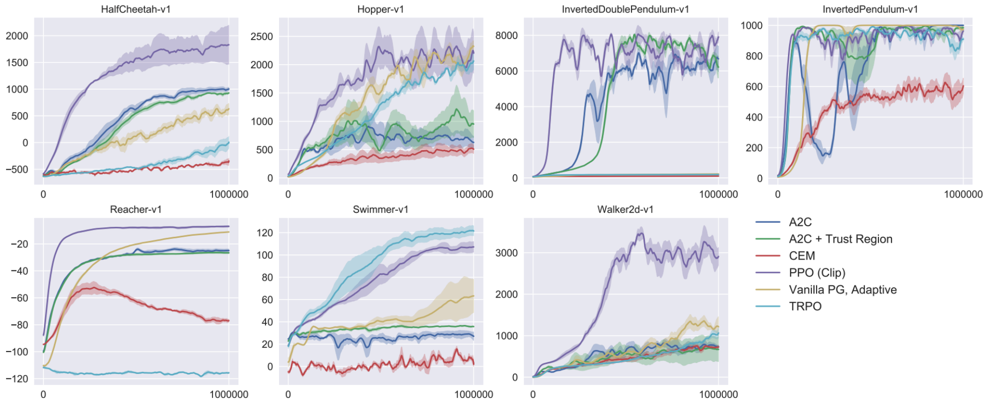
○ PPO penalizes large deviation from the current policy directly inside the objective function through clipping the ratio $\frac{\pi_\theta}{\pi_{\theta_t}}$.

$$\text{clip}(x; 1-\epsilon; 1+\epsilon) = \begin{cases} 1-\epsilon, & \text{if } x < 1-\epsilon \\ 1+\epsilon, & \text{if } x > 1+\epsilon \\ x, & \text{otherwise} \end{cases}$$

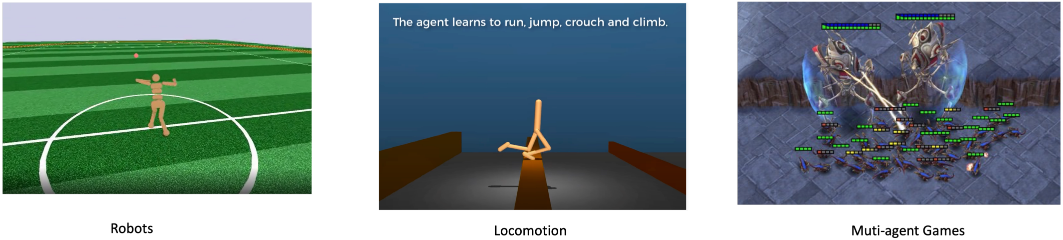
○ Run SGD. No need to deal with the KL divergence or trust region constraints.

○ Vastly adopted in practice but little is known about its theoretical properties.

Numerical Performance [15]



More Applications



Robots

Locomotion

Muti-agent Games

Figure: PPO performs well in many locomotion task and games.

- Some links:
  - `https://www.youtube.com/watch?v=hx_bgoTF7bs`
  - `https://openai.com/blog/openai-baselines-ppo/`

Summary

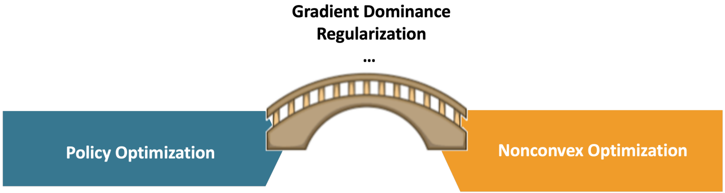
Vanilla Policy Gradient
Natural Policy Gradient
TRPO
ACKTR & PPO

Theory

Practice

Figure from Schulman's slide on PPO in 2017.

Summary



Gradient Dominance
Regularization
...

Policy Optimization

Nonconvex Optimization

Vanilla Policy Gradient	Gradient Descent
REINFORCE	Stochastic Gradient Descent
Natural Policy Gradient TRPO PPO	Mirror Descent
Conservative Policy Iteration ...	Frank Wolfe ...

Algorithms comparison.

- The main algorithms are summarized in the next table.

Algorithm	NMC[1] $\epsilon_{\text{stat}} = 0$	NMC $\epsilon_{\text{stat}} > 0$	EFH[2]	EIH[3]
Vanilla PG	✗	✗	✗	✗
NPG	✓	✗	✗	✗
TRPO	✓	✗	✗	✗
MD-MPI	✓	✗	✗	✗
OPPO	✓	✓	✓	✗
PC-PG	✓	✓	✓	✓

Remarks:

- Recall that a drawback of PC-PG is the extremely high sample complexity.
- The use of baselines can be beneficial but the right baseline is difficult to pick.
- PPO is a very practical algorithm.

[1] NMC: No Mismatch Coefficients.

[2] Exploration in Finite Horizon

[3] Exploration in Infinite Horizon

References I

[1] Joshua Achiam.
Spinning Up in Deep Reinforcement Learning.
2018.
31

[2] Alekh Agarwal, Mikael Henaff, Sham Kakade, and Wen Sun.
Pc-pg: Policy cover directed exploration for provable policy gradient learning.
Advances in neural information processing systems, 33:13399–13412, 2020.
43, 45

[3] Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan.
Optimality and approximation with policy gradient methods in markov decision processes.
In *Conference on Learning Theory*, pages 64–66. PMLR, 2020.
17, 21, 25, 26

[4] Amir Beck and Marc Teboulle.
Mirror descent and nonlinear projected subgradient methods for convex optimization.
Operations Research Letters, 31(3):167–175, 2003.
12

[5] Sébastien Bubeck.
Convex optimization: Algorithms and complexity.
Foundations and Trends in Machine Learning, 8(3-4):231–358, 2015.
12

References II

[6] Qi Cai, Zhuoran Yang, Chi Jin, and Zhaoran Wang.
Provably efficient exploration in policy optimization.
In *International Conference on Machine Learning*, pages 1283–1294. PMLR, 2020.
38, 42

[7] Wesley Chung, Valentin Thomas, Marlos C Machado, and Nicolas Le Roux.
Beyond variance reduction: Understanding the true impact of baselines on policy optimization.
In *International Conference on Machine Learning*, pages 1999–2009. PMLR, 2021.
48

[8] Eyal Even-Dar, Sham M Kakade, and Yishay Mansour.
Online markov decision processes.
Mathematics of Operations Research, 34(3):726–736, 2009.
28, 32

[9] Matthieu Geist, Bruno Scherrer, and Olivier Pietquin.
A Theory of Regularized Markov Decision Processes.
In *International Conference on Machine Learning (ICML)*, 2019.
28, 33, 36

[10] S. Kakade.
A natural policy gradient.
In *Advances in Neural Information Processing Systems (NeurIPS)*, 2001.
15

References III

[11] Vijay R Konda and John N Tsitsiklis.
On actor-critic algorithms.
SIAM journal on Control and Optimization, 42(4):1143–1166, 2003.
4

[12] Jincheng Mei, Wesley Chung, Valentin Thomas, Bo Dai, Csaba Szepesvari, and Dale Schuurmans.
The role of baselines in policy gradient optimization.
arXiv preprint arXiv:2301.06276, 2023.
50

[13] G. Neu, A. Jonsson, and V. Gómez.
A unified view of entropy-regularized Markov decision processes.
arXiv:1705.07798, 2017.
32

[14] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz.
Trust region policy optimization.
In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.
30

[15] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov.
Proximal policy optimization algorithms.
arXiv preprint arXiv:1707.06347, 2017.
51, 52

References IV

[16] Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al.
Policy gradient methods for reinforcement learning with function approximation.
In *Conference on Neural Information Processing Systems*, pages 1057–1063, 1999.
18

[17] Nino Vieillard, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Rémi Munos, and Matthieu Geist.
Leverage the average: an analysis of kl regularization in rl.
arXiv preprint arXiv:2003.14089, 2020.
36

[18] Ronald J Williams.
Simple statistical gradient-following algorithms for connectionist reinforcement learning.
Machine learning, 8(3-4):229–256, 1992.
4