Reinforcement Learning

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 5: Policy Gradient II

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

EE-568 (Spring 2025)

License Information for Reinforcement Learning (EE-568)

Recap: Policy optimization

◦ The objective of reinforcement learning in terms of the policy parameters is given by the following:

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

Tabular parametrization

- Direct parameterization:

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

- Softmax parameterization:

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

Non-tabular parametrization

- Softmax parameterization:

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

- Gaussian parameterization:

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

Recap: Policy gradient methods

∘ The exact policy gradient method is a special case of the stochastic policy gradient method.

Stochastic policy gradient method

By stochastic policy gradient method, we mean the following update rule:

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

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 [18]
- Actor-critic [11]
- ...

Previous lecture

○ In the previous lecture, we answered the following two questions.

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

◦ In the previous lecture, we answered the following two questions.

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:

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

◦ 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

◦ In the previous lecture, we answered the following two questions.

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:

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

◦ 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

- In this lecture, we will answer the following questions.

Question 3 (theory)

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

- To answer Question 3, we first revisit some optimization background (next few slides).

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?

- To answer Question 4, we will have a look at recent papers (second part of this lecture).

The algorithmic path towards an understanding

- We will discover NPG and the two closely related algorithms: TRPO and OPPO.
- We will study the implications of advantage estimation and exploration in their convergence.
- We will further discuss the successful PPO algorithm.

Algorithm	Convergence rate	Unknown transitions	Hard environments
Vanilla PG [16]	$\mathcal{O}\left(\frac{16\lvert\mathcal{S}\rvert\kappa^2}{c^2(1-\gamma)^5 T}\right)$	✗	✗
Tabular NPG [2]	$\mathcal{O}\left(\frac{2}{(1-\gamma)^2 T}\right)$	✗	✓
Sample-based NPG	$\mathcal{O}\left(\frac{1}{1-\gamma}\sqrt{\frac{2\log\lvert\mathcal{A}\rvert}{T}} + \sqrt{\kappa\epsilon_{\text{stat}}}\right)$	✓	✗
OPPO [5]	$\mathcal{O}\left(\frac{\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{\sqrt{(1-\gamma)^3 T}}\right)$	✓	✓

Remarks:

- Here are the key quantities in the table:
  - $c = [\min_{s,t} \pi_{\theta_t}(a^\star(s)|s)]^{-1} > 0$
  - $\kappa = \left\| \frac{\lambda_\mu^{\pi^\star}}{\mu} \right\|_\infty$ is larger when it is harder to explore and is possibly ∞.
  - ϵ_{stat} is the statistical error incurred in estimating the advantage function A^π.

Revisiting gradient descent

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

Question: ◦ Would GD give the same trajectory under invertible linear transformations ($\mathbf{x} \to \mathbf{A}\mathbf{x}$)?

Revisiting gradient descent (cont'd)

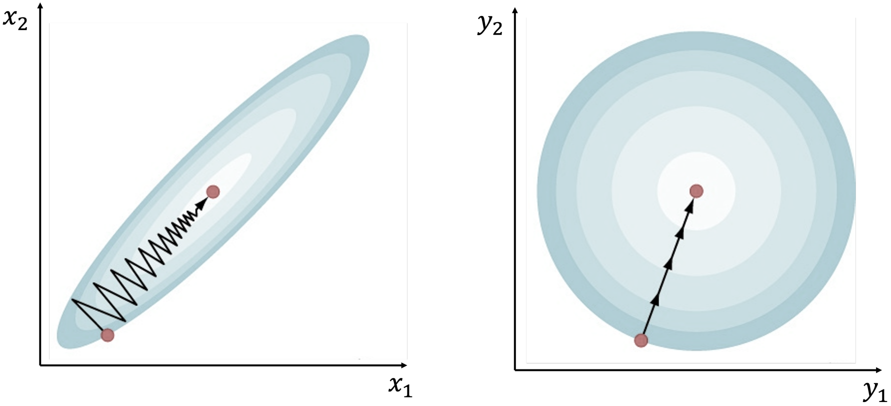


Figure: GD is not invariant w.r.t. linear transformations.

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 and initialization $\mathbf{x}_0$, the iterates of mirror descent [3] are given by

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

Examples:

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

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

- Entropic mirror descent [3]: $\mathcal{X} = \Delta_d$, $\omega(\mathbf{x}) = \sum_{i=1}^d x_i \log x_i$, $D_\omega(\mathbf{x}, \mathbf{x}_t) = \mathrm{KL}(\mathbf{x}\|\mathbf{x}_t)$

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

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

- Entropic Mirror Descent attains nearly dimension-free convergence [3] (also see Chapter 4 [4]).
- 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):
$$\theta_{t+1} = \theta_t - \eta(F_{\theta_t})^\dagger \nabla_\theta f(\theta_t).$$
  - Equivalent regularized form:
$$\theta_{t+1} = \arg\min_\theta \left\{\nabla_\theta f(\theta_t)^\top(\theta-\theta_t) + \frac{1}{2\eta}(\theta-\theta_t)^\top F_{\theta_t}(\theta-\theta_t)\right\}.$$
  - Equivalent trust region form:
$$\theta_{t+1} = \arg\min_\theta \nabla_\theta f(\theta_t)^\top(\theta-\theta_t), \ \text{s.t. } \frac{1}{2}(\theta-\theta_t)^\top F_{\theta_t}(\theta-\theta_t) \le \frac{1}{2}\eta^2 \nabla_\theta f(\theta_t)^\top F_{\theta_t}^\dagger \nabla_\theta f(\theta_t).$$

Natural Policy Gradient (NPG)

Natural Policy Gradient (Kakade, 2002)[9]

Given the reinforcement learning objective $\max_\theta J(\pi_\theta) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 \sim \mu, \pi_\theta\right] = \mathbb{E}_{s\sim\mu}[V^{\pi_\theta}(s)]$, the iterates of NPG are given by

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

where $\eta > 0$ is the step-size of the algorithm.

Key elements:

- 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, which can be written as follows

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

- $A^{\pi_\theta}(s,a)$ is the advantage function:

$$A^{\pi_\theta}(s,a) = Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s).$$

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

Interpretation of NPG

∘ The update rule of NPG can be viewed as solving the quadratic approximation of the problem:

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

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

Explanation:

∘ Approximate the objective with the first-order Taylor expansion:

$$J(\pi_\theta) \approx J(\pi_{\theta_t}) + \nabla_\theta J(\pi_{\theta_t})^\top (\theta - \theta_t).$$

∘ Approximate the constraint with the second-order Taylor expansion (See Slide 11):

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

∘ Set $\delta = \frac{1}{2}\eta^2 \nabla_\theta f(\theta_t)^\top F_{\theta_t}^\dagger \nabla_\theta f(\theta_t)$ and see Slide 13

Question:

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

Computing natural policy gradient

∘ As opposed to naively computing $(F_\theta)^\dagger \nabla_\theta J(\pi_\theta)$ in NPG, we will use a key identity.

Equivalent form of NPG (Appendix C.3 [2])

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:

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

where $A^{\pi_\theta}(s,a)$ is the advantage function $A^{\pi_\theta}(s,a) = Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)$.

Proof:

$$\nabla_w \mathbb{E}_{s\sim\lambda_\mu^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)} \left[\left(w^\top \nabla_\theta \log \pi_\theta(a|s) - A^{\pi_\theta}(s,a) \right)^2 \right] \Bigg|_{w^\star(\theta)} = 0$$

$$2w^\star(\theta)^\top \underbrace{\mathbb{E}_{s\sim\lambda_\mu^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)} \left[\nabla_\theta \log \pi_\theta(a|s) \nabla_\theta \log \pi_\theta(a|s)^\top \right]}_{F_\theta} - 2 \underbrace{\mathbb{E}_{s\sim\lambda_\mu^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)} \left[A^{\pi_\theta}(s,a) \nabla_{\theta_t} \log \pi_\theta(a|s) \right]}_{(1-\gamma)\nabla_\theta J(\pi_\theta)} = 0$$

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

Computing natural policy gradient

◦ As opposed to naively computing $(F_\theta)^\dagger \nabla_\theta J(\pi_\theta)$ in NPG, we will use a key identity.

Equivalent form of NPG (Appendix C.3 [2])

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:

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

where $A^{\pi_\theta}(s,a)$ is the advantage function $A^{\pi_\theta}(s,a) = Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)$.

Remarks: ◦ Note that since the update rule of NPG is $\theta_{t+1} = \theta_t + \eta(F_\theta)^\dagger \nabla_\theta J(\pi_\theta)$, 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.

Example 1: Tabular NPG under softmax parameterization

◦ With softmax parameterization, the NPG becomes the policy mirror descent algorithm (Slide 11)

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 NPG parameter update can be simplified to the following:

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

Proof available in the Supplementary material.

NPG policy update + softmax parametrization = policy mirror descent

In policy space, the induced update corresponds to the following:

$$\pi_{t+1}(a|s) = \pi_t(a|s) \frac{\exp(\eta/(1-\gamma) \cdot A^{\pi_t}(s,a))}{Z_t(s)}, \text{ where } Z_t(s) = \frac{\sum_{a'} \exp(\theta_{t,s,a'})}{\sum_{a'} \exp(\theta_{t,s,a'} + \eta/(1-\gamma) \cdot A^{\pi_t}(s,a'))}.$$

Example 2: NPG with linear function approximation

∘ In this case, we can also express the NPG update rule via a regression problem.

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 $\pi_t = \pi_{\theta_t}$. In this case we have that $\nabla_\theta \log(\pi_\theta(a|s)) = \phi(s,a) - \sum_{a'} \pi_\theta(a|s')\phi(s,a')$ and consequently:

$$w^\star(\theta) \in \arg\min_w \mathbb{E}_{s\sim\lambda_\mu^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)} \left[\left(w^\top \left(\phi(s,a) - \sum_{a'} \pi_\theta(a|s')\phi(s,a') \right) - A^{\pi_\theta}(s,a) \right)^2 \right].$$

Finally, the induced NPG parameter update becomes: $\theta_{t+1} = \theta_t + \frac{\eta}{1-\gamma} w^\star(\theta_t)$

NPG policy update + softmax parametrization = policy mirror descent

Similarly, we can obtain a mirror descent update rule in the policy space.

$$\pi_{t+1}(a|s) = \pi_t(a|s) \frac{\exp\left(\frac{\eta}{(1-\gamma)} w^\star(\theta_t)^\top \phi(s,a) \right)}{Z_t(s)}, \text{ where } Z_t(s) = \frac{\sum_{a'} \exp(\theta_{t,s,a'})}{\sum_{a'} \exp\left(\theta_{t,s,a'} + \frac{\eta}{(1-\gamma)} w^\star(\theta_t)^\top \phi(s,a') \right)}$$

Convergence of tabular NPG with softmax parametrization

◦ **Question:** In the case of NPG with softmax parametrization, how fast do we converge to the optimal solution?

NPG policy update

Remember that for the softmax parametrization we have:

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

Convergence of tabular NPG [2]

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.
- ◦ In the case of known environment, $\eta = \infty$ recovers Policy Iteration (Supplementary material)

Question:

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

Sample-based NPG

◦ **Questions:** What if we do not know the environment? Can we estimate $A^{\pi_t}(s,a)$?

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** {NPG steps}

  Initialize w_0, denote $\pi_t = \pi_{\theta_t}$

  for $n = 0, 1, \ldots, N-1$ **do** {Gradient Descent steps for the regression problem}

    Sample $s \sim \lambda_\mu^{\pi_t}$, $a \sim \pi_t(\cdot|s)$

    Estimate $\hat{A}(s,a)$ {Unbiased estimator of $A^{\pi_t}(s,a)$}

    Update $w_{n+1} \leftarrow w_n - \alpha(w^\top \nabla_\theta \log \pi_t(a|s) - \hat{A}(s,a)) \cdot \nabla_\theta \log \pi_t(a|s)$ {Gradient Descent step}

  end for

  Update $\theta_{t+1} = \theta_t + \frac{\eta}{1-\gamma} w_N$ {NPG step}

end for

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

Sampling routine for λ^π_μ

Input : a policy π.
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 π.
Sample $(s_T, a_T) \sim \lambda^\pi_\mu$, 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 probability $1-\gamma$ terminate.
end while
Output : $\hat{Q}$.

Remarks:

- See Algorithm 1 in [2].
- We sample from the occupancy measure by generating (s_T, a_T) with $T \sim \text{Geometric}(1-\gamma)$.
- $\hat{Q}$ is an unbiased estimate of $Q(s_T, a_T)$.
- Unbiased estimates of $V(s_T)$ and $A(s_T, a_T)$ can be obtained from $\hat{Q}(s,a)$.

Convergence of sample-based NPG with function approximation

◦ We provide convergence guarantees for sample-based NPG in the linear function approximation case.

Convergence of sampled-based NPG (informal)

Let $\pi_\theta(a|s) = \frac{\exp(\theta^\top \phi(s,a))}{\sum_{a'} \exp(\theta^\top \phi(s,a'))}$ and $\theta^\star$ be the parameters asociated to the optimal policy.

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

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:

◦ $\epsilon_{\text{bias}} = 0$ under the so called "realizability" assumption for the features i.e.,

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

◦ $\kappa = \left\|\frac{\lambda_\mu^{\pi^\star}}{\mu}\right\|_\infty$ quantifies how exploratory the initial distribution is and **might be unbounded**

Question:

◦ Can we obtain an algorithm that converges in hard to explore environments (unbounded κ)?

Markov Decision Processes - Experts (MDP-E) [7]

Markov Decision Processes - Experts (MDP-E)

Initialize policy π_0, learning rate η

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 $\pi_0, \ldots, \pi_{T-1}$.

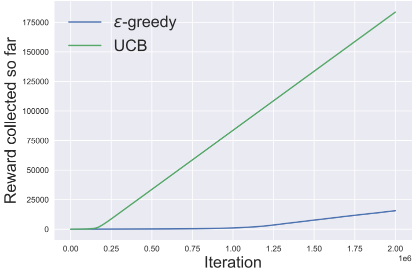
Remarks:

- 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.

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.
- Can we incorporate exploration techniques in policy gradient?

  e.g., ϵ-greedy [17] and UCB [8] (we studied in the first coding exercise.)



Recall: Finite Horizon RL

- The agent interacts with the environment for K rounds with horizon H.
- The objective is to find the policy that maximizes $\mathbb{E}_\pi\left[\sum_{h=1}^{H} r(s_h, a_h)\right]$.
- The optimal policy is non stationary.
- A non stationary policy is a collection of H policies $\pi_1, \ldots, \pi_H$.
- π_1 is used for the first decision, π_2 is used for the second decision and so on ....
- The value functions depend on the stage h, that is

$$Q_h^\pi(s,a) = \mathbb{E}_\pi\left[\sum_{h'=h}^{H} r(s_{h'}, a_{h'}) | s_h = s, a_h = a\right], \quad V_h^\pi(s) = \mathbb{E}_\pi\left[\sum_{h'=h}^{H} r(s_{h'}, a_{h'}) | s_h = s\right]$$

Optimistic variant of the Proximal Policy Optimization (OPPO)

- **Key idea:** Perform updates with *optimistic* estimates of the value function.
- OPPO resembles NPG/MDP-E but with an optimistic evaluation step.

OPPO [5] (simplified version)

Initialize policy parameter $\theta_0 \in \mathbb{R}^d$, step size $\eta > 0$, $\alpha > 0$
for $t = 0, 1, \ldots, T-1$ **do**

Policy Evaluation
Estimate bonus and transitions $\mathrm{bonus}_h(s,a)$ and $\hat{P}_h(s'|s,a)$

Compute optimistic value functions Q_h^t

Policy Improvement
Update policies at every h, s, a with a NPG/MDP-E step

$$\pi_h^{t+1}(a|s) \propto \pi_h^t(a|s) \exp \eta Q_h^t(s,a)$$

end for

Estimate transition and bonuses

- Compute the empirical average of the transition dynamics.
- Set the function $\text{bonus}_h^t(s,a)$ 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 $t = 0, 1, \ldots, T-1$ **do**
  for $h = 0, 1, \ldots, H-1$ **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 $\text{bonus}_h(s,a) \approx \sqrt{\frac{1}{N(s_h^t, a_h^t)}}$.
  end for
end for

Estimate optimistic value function

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

Backward induction to estimate Q^t.

Initialize $Q_{H+1}^t(s,a) = 0$.

for $h = H, \ldots, 1$ **do**

Recurse backward to compute Q_h^t

$$Q_h^t(s,a) = r_h^t(s,a) + \text{bonus}_h^t(s,a) + \sum_{s',a'} \hat{P}_h(s'|s,a)\pi_{h+1}(a'|s')Q_{h+1}^t(s',a')$$

$$Q_h^t(s,a) = \text{clip}(Q_h^t(s,a); 0, H-h+1)$$

end for

Remark: ◦ If it holds that $\left|\sum_{s'}(\hat{P}_h(s'|s,a) - P_h(s'|s,a))V(s')\right| \leq \text{bonus}_h(s,a)$, then Optimism and Bounded Optimism hold.

Provable exploration in policy gradient

- 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.

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

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

Benefit of OPPO

- The regret bound of OPPO: $\sum_{t=1}^{T} V^\star(s_1) - V^{\pi_t}(s_1) \leq \mathcal{O}\left(\sum_{h=1}^{H} \sum_{t=1}^{T} \text{bonus}_h^t(s_h^t, a_h^t)\right)$.
- Next, one shows that $\sum_{h=1}^{H} \sum_{t=1}^{T} \text{bonus}_h^t(s_h^t, a_h^t) \leq \mathcal{O}(\sqrt{T})$.

Theorem

Let $\pi^1, \pi^2, \dots, \pi^T$ the sequence of non stationary 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.

Recall convergence of sampled-based NPG

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

where κ depends on the initial distribution and the environment.

Remarks: ◦ OPPO is much better because it removes the dependence on κ.

Revisiting baselines

- The baselines can be used as a variance reduction mechanism.
- Actually, one can prove which choice for the baseline guarantees minimum variance.

Theorem

Consider the gradient with baseline $\widehat{\nabla}_\theta J(\pi_\theta) = \sum_{t=1}^{\infty} (Q^{\pi_\theta}(s_t, a_t) - b(s_t)) \nabla \log \pi_\theta(a_t|s_t)$ *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*

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

Is it always good to minimize variance?

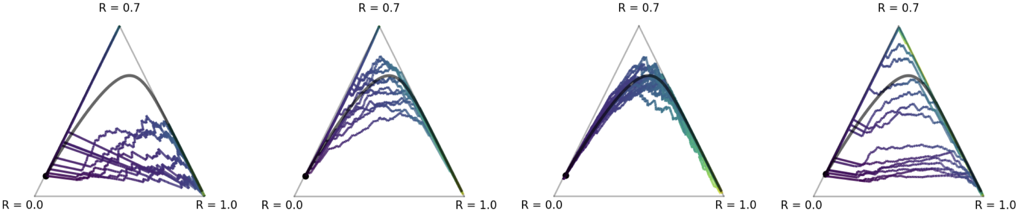
- The answer is no. Because, reducing the variance of the baseline can hinder exploration.
- As a result, the minimum variance baseline may lead to a suboptimal policy.
- Here we describe the result in [6].

Theorem

Theorem 1 in [6] 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

- Three-arm bandit enviroment example:



(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.

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)

◦ How to choose the step-size of the stochastic policy gradient method? Trust region.

TRPO (key idea) [14]

TRPO computes the marginal benefit of a new policy with respect to an old policy:

$$\theta_{t+1} = \arg\max_{\theta} \quad \mathbb{E}_{s \sim \lambda_\mu^{\pi_{\theta_t}}, a \sim \pi_{\theta_t}(\cdot|s)} \left[\frac{\pi_\theta(a \mid s)}{\pi_{\theta_t}(a \mid s)} A^{\pi_{\theta_t}}(s, a) \right],$$

$$\text{s.t.} \quad \mathbb{E}_{s \sim \lambda_\mu^{\pi_{\theta_t}}} [\mathrm{KL}(\pi_\theta(\cdot \mid s) \| \pi_{\theta_t}(\cdot \mid s))] \leq \delta.$$

where the constraint measures the distance between two policies.

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

- Compute a search direction, which (almost) boils down to natural policy gradient.
  - The first order approximation of the objective.

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

  - The second order expansion of the constraints

$$\mathbb{E}_{s\sim\lambda_\mu^{\pi_{\theta_t}}}[\mathrm{KL}(\pi_\theta(\cdot \mid s) \| \pi_{\theta_t}(\cdot \mid s))] \approx \frac{1}{2}(\theta - \theta_k)^T F(\theta_k)(\theta - \theta_k)$$

- Execute line seach along the direction $F(\theta_k)^\dagger \nabla_\theta J(\theta_k)$.
  - Approximations may result in a solution that does not satisfy the origin trust region.
  - Select the largest possible step size η that $x_{t+1} = x_t + \eta F(\theta_k)^\dagger \nabla_\theta J(\theta_k)$ satisfies the original constraints:

$$\eta = \sqrt{\frac{2\delta}{\nabla_\theta J(\theta_k)^\top F(\theta_k)^\dagger \nabla_\theta J(\theta_k)}}$$

Equivalence between TRPO and MDP-E [7]

- The previous result proves that TRPO produces a monotonically improving sequence of policies [14, Section 3].
- We can prove a stronger result noticing that TRPO is equivalent to MDP-E [13, Section B.3] and [7].

Proximal policy optimization (PPO2)

∘ **Intuition:** The main problem of TRPO lies in numerically computing the Quadratic Program.

∘ **Solution:** Theoretical update equation is optimizing in a local region.

PPO uses no formal constraints and instead clips the distance between policies in the loss function.

PPO (key idea) [15]

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

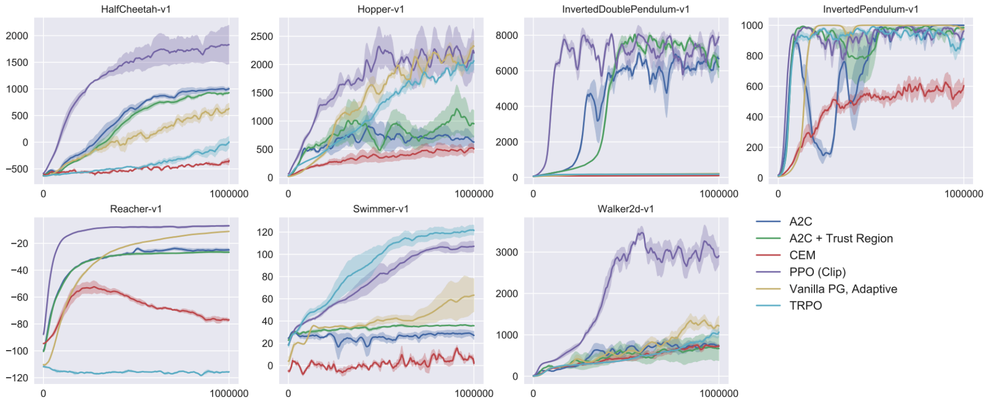
Remarks: ∘ PPO penalizes large deviations directly inside the objective function through clipping the ratio $\frac{\pi_\theta}{\pi_{\theta_t}}$:

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

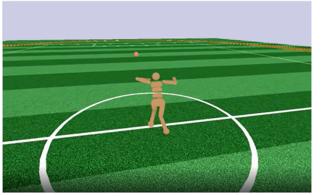
∘ Run SGD. No need to deal with the KL divergence or trust region constraints.

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

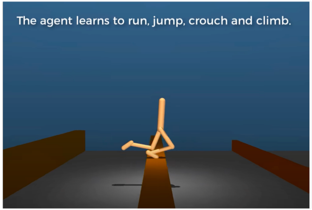
Numerical performance [15]



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

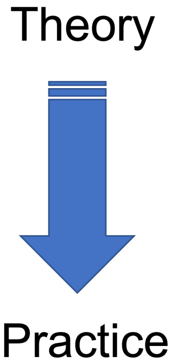
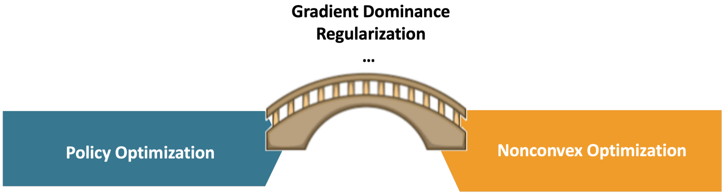


Figure from Schulman's slide on PPO in 2017.

Summary



Vanilla Policy Gradient [16]	Gradient Descent
REINFORCE [18]	Stochastic Gradient Descent
Natural Policy Gradient [9] TRPO [1] PPO [15]	Mirror Descent
Conservative Policy Iteration [10] . . .	Frank Wolfe . . .

References I

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel.
Constrained policy optimization.
In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 22–31. JMLR. org, 2017.
46

[2] 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.
9, 18, 19, 22, 24

[3] Amir Beck and Marc Teboulle.
Mirror descent and nonlinear projected subgradient methods for convex optimization.
Operations Research Letters, 31(3):167–175, 2003.
13

[4] Sébastien Bubeck.
Convex optimization: Algorithms and complexity.
Foundations and Trends in Machine Learning, 8(3-4):231–358, 2015.
13

[5] 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.
9, 29

References II

[6] 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.
35

[7] Eyal Even-Dar, Sham M Kakade, and Yishay Mansour.
Online markov decision processes.
Mathematics of Operations Research, 34(3):726–736, 2009.
26, 41

[8] Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan.
Is q-learning provably efficient?
Advances in neural information processing systems, 31, 2018.
27

[9] S. Kakade.
A natural policy gradient.
In *Advances in Neural Information Processing Systems (NeurIPS)*, 2001.
16, 46

[10] Sham Kakade and John Langford.
Approximately optimal approximate reinforcement learning.
In *In Proc. 19th International Conference on Machine Learning*. Citeseer, 2002.
46

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.
37

[13] G. Neu, A. Jonsson, and V. Gómez.
A unified view of entropy-regularized Markov decision processes.
arXiv:1705.07798, 2017.
41

[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.
39, 41

[15] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov.
Proximal policy optimization algorithms.
arXiv preprint arXiv:1707.06347, 2017.
42, 43, 46

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.
9, 46

[17] Christopher John Cornish Hellaby Watkins.
Learning from Delayed Rewards.
PhD thesis, King's College, Cambridge, UK, May 1989.
27

[18] Ronald J Williams.
Simple statistical gradient-following algorithms for connectionist reinforcement learning.
Machine learning, 8(3-4):229–256, 1992.
4, 46

Supplementary Material

Tabular NPG under softmax parametrization.

Proof.

We need to show that $w^*(\theta_t) = A^{\pi_t}$ in the case of softmax parametrization. To do so, we will first compute:

$$\nabla_\theta \log(\pi_\theta(a|s)) = \nabla_\theta \left(\theta_{s,a} - \log\left(\sum_{a'} \exp(\theta_{s,a'})\right)\right) = e_{s,a} - \pi_\theta(\cdot|s)\,.$$

In this case, we can check that $A^{\pi_\theta} \in \arg\min_w \mathbb{E}_{s\sim\lambda_\mu^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)}\left[\left(w^\top \nabla_\theta \log \pi_\theta(a|s) - A^{\pi_\theta}(s,a)\right)^2\right]$ because:

$$\begin{aligned}
\left(A^{\pi_\theta\top}\nabla_\theta \log \pi_\theta(a|s) - A^{\pi_\theta}(s,a)\right) &= \left(A^{\pi_\theta\top}(e_{s,a} - \pi_\theta(\cdot|s)) - A^{\pi_\theta}(s,a)\right)\\
&= A^{\pi_\theta}(s,a) - A^{\pi_\theta}(s,a) + \sum_{a'} \pi_\theta(a'|s))A^{\pi_\theta}(s,a')\\
\text{[Def. of } A^{\pi_\theta}(s,a)] &= \sum_{a'} \pi_\theta(a'|s))(Q^{\pi_\theta}(s,a') - V^{\pi_\theta}(s))\\
\text{[Def. of } V^{\pi_\theta}(s)] &= V^{\pi_\theta}(s)) - V^{\pi_\theta}(s))\\
&= 0
\end{aligned}$$

□

Proof of tabular NPG convergence

Lemma (Policy Improvement)

For any policy π and π_{t+1} being obtained with NPG in the softmax parametrization setup, we can express the performance difference as:

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

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 $\pi_{t+1}(a|s) = \pi_t(a|s)\frac{\exp(\eta A^{\pi_t}(s,a)/(1-\gamma))}{Z_t(s)}$, 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 (cont'd)

Proof (NPG):

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

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

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

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

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

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

NPG in the $\eta = \infty$ setup.

In the case of being able to compute A^{π_θ}, and setting $\eta = \infty$, we can see that NPG is equivalent to Policy Iteration (Lecture 2). Taking the NPG update rule for the softmax parametrization to the limit:

$$\begin{aligned}
\pi_{t+1}(a|s) &= \lim_{\eta\to\infty} \pi_t(a|s) \cdot \frac{\exp(\eta/(1-\gamma)A^{\pi_t}(s,a)) \cdot \sum_{a'} \exp(\theta_{t,s,a'})}{\sum_{a'} \exp(\theta_{t,s,a'} + \eta/(1-\gamma)A^{\pi_t}(s,a'))} \\
&= \lim_{\eta\to\infty} \frac{\pi_t(a|s)}{e^{\theta_{t,s,a}}} \cdot \frac{\exp(\theta_{t,s,a} + \eta/(1-\gamma)A^{\pi_t}(s,a)) \cdot \sum_{a'} \exp(\theta_{t,s,a'})}{\sum_{a'} \exp(\theta_{t,s,a'} + \eta/(1-\gamma)A^{\pi_t}(s,a'))} \\
&= \lim_{\eta\to\infty} \frac{\exp(\theta_{t,s,a} + \eta/(1-\gamma)A^{\pi_t}(s,a))}{\sum_{a'} \exp(\theta_{t,s,a'} + \eta/(1-\gamma)A^{\pi_t}(s,a'))}
\end{aligned}$$

$$[\lim_{\eta\to\infty} \text{softmax}(\eta \cdot x)_i = \mathbb{1}\{x_i = \max x\}] = \mathbb{1}\left\{a = \max_{a'} A^{\pi_t}(s,a')\right\}.$$

This means under $\eta = \infty$, we have that NPG gives us a greedy policy, where the action taken is given by:

$$\arg\max_{a'} A^{\pi_t}(s,a') = \arg\max_{a'} Q^{\pi_t}(s,a') - V^{\pi_t}(s) = \arg\max_{a'} Q^{\pi_t}(s,a'),$$

which is precisely the update formula for Policy Iteration.

Proof for the analytical expression with lowest variance.

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. □