Sequential Decision Making: Theory and Practice

Lecture 7: Policy Optimization

Yuejie Chi

Yale Statistics & Data Science



Fall 2026

Outline

Introduction to policy gradient methods

Global convergence of softmax policy gradient methods

Natural policy gradient methods

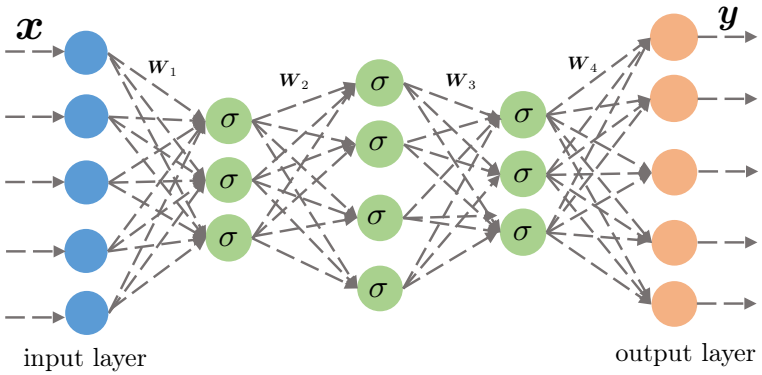
Entropy/KL-regularized policy optimization

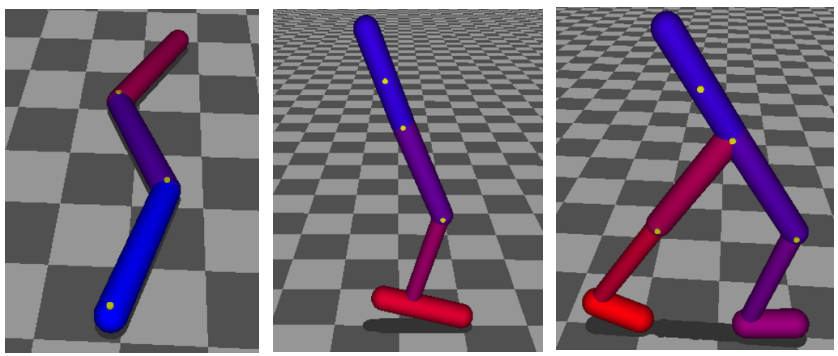
Actor-critic methods

Policy optimization

$$\text{maximize}_\theta \quad \text{value}(\text{policy}(\theta))$$

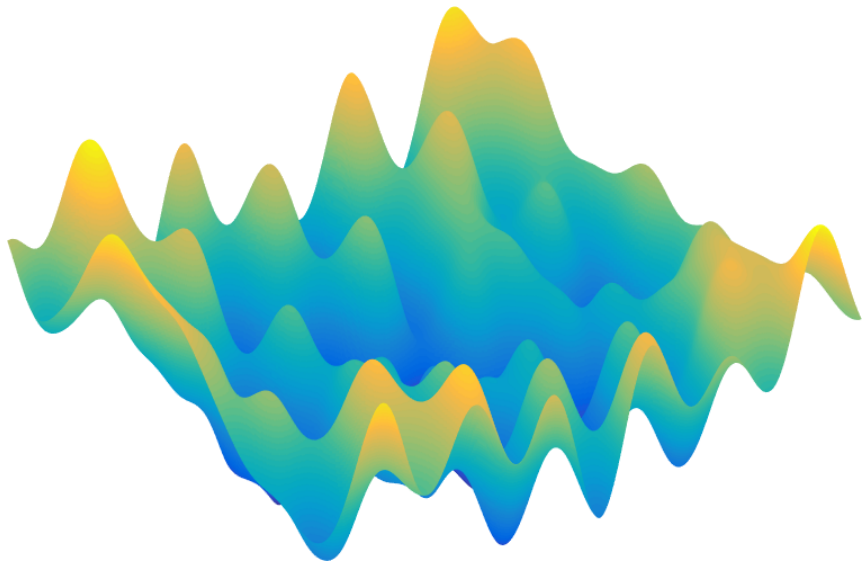
- directly optimize the policy, which is the quantity of interest;
- allow flexible differentiable parameterizations of the policy;
- work with both continuous and discrete problems.





Theoretical challenges: non-concavity

Little understanding on the global convergence of policy gradient methods until very recently, e.g. (Fazel et al., 2018; Bhandari and Russo, 2019; Agarwal et al., 2019; Mei et al. 2020), and many more.



Our goal:

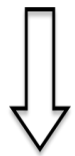
- introduce the algorithmic framework of popular policy gradient methods
- understand finite-time convergence rates of popular heuristics

Introduction to policy gradient methods

Policy gradient methods

Given an initial state distribution $s \sim \rho$, find policy π such that

$$\text{maximize}_\pi \quad V^\pi(\rho) := \mathbb{E}_{s\sim\rho}\left[V^\pi(s)\right]$$



Parameterization:

$$\pi := \pi_\theta$$

$$\text{maximize}_\theta \quad V^{\pi_\theta}(\rho) := \mathbb{E}_{s\sim\rho}\left[V^{\pi_\theta}(s)\right]$$

Policy gradient method

For $t = 0, 1, \cdots$

$$\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_\theta V^{\pi_\theta^{(t)}}(\rho)$$

where η *is the learning rate.*

How to calculate the gradient?

Policy gradient derivation

- Assume π_θ is differentiable when it is non-zero with gradient $\nabla_\theta \pi_\theta$.

The policy gradient can be decomposed as

$$
\begin{aligned}
&\nabla_\theta V^{\pi_\theta}(\rho) \\
&= \nabla_\theta \mathbb{E}_{s_0 \sim \rho}\left[V^{\pi_\theta}(s_0)\right] \\
&= \mathbb{E}_{s_0 \sim \rho}\left[\nabla_\theta \left(\sum_{a_0 \in \mathcal{A}} \pi_\theta(a_0|s_0) Q^{\pi_\theta}(s_0, a_0)\right)\right] \\
&= \mathbb{E}_{s_0 \sim \rho}\left[\sum_{a_0 \in \mathcal{A}} \nabla_\theta \pi_\theta(a_0|s_0) Q^{\pi_\theta}(s_0, a_0) + \sum_{a_0 \in \mathcal{A}} \pi_\theta(a_0|s_0) \nabla_\theta Q^{\pi_\theta}(s_0, a_0)\right]
\end{aligned}
$$

We discuss the two terms separately.

Policy gradient derivation - first term

Note that

$$\nabla_\theta \pi_\theta(a|s) = \pi_\theta(a|s) \frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)} = \pi_\theta(a|s) \underbrace{\nabla_\theta \log \pi_\theta(a|s)}_{\text{score function}}.$$

The first term in the policy gradient is expressed as

$$\begin{aligned}
&\mathbb{E}_{s_0 \sim \rho} \left[\sum_{a_0 \in \mathcal{A}} \nabla_\theta \pi_\theta(a_0|s_0) Q^{\pi_\theta}(s_0, a_0) \right] \\
&= \mathbb{E}_{s_0 \sim \rho} \left[\sum_{a_0 \in \mathcal{A}} \pi_\theta(a_0|s_0) \nabla_\theta \log \pi_\theta(a_0|s_0) Q^{\pi_\theta}(s_0, a_0) \right] \\
&= \mathbb{E}_{s_0 \sim \rho, a_0 \sim \pi_\theta(\cdot|s_0)} \left[\nabla_\theta \log \pi_\theta(a_0|s_0) Q^{\pi_\theta}(s_0, a_0) \right]
\end{aligned}$$

Policy gradient derivation - second term

The second term in the policy gradient is expressed as

$$
\begin{aligned}
&\mathbb{E}_{s_0\sim\rho}\left[\sum_{a_0\in\mathcal{A}}\pi_\theta(a_0|s_0)\nabla_\theta Q^{\pi_\theta}(s_0,a_0)\right]\\
&=\mathbb{E}_{s_0\sim\rho,a_0\sim\pi_\theta(\cdot|s_0)}\left[\nabla_\theta Q^{\pi_\theta}(s_0,a_0)\right]\\
&=\mathbb{E}_{s_0\sim\rho,a_0\sim\pi_\theta(\cdot|s_0)}\left[\nabla_\theta\big(r(s_0,a_0)+\gamma\mathbb{E}_{s_1\sim P(\cdot|s_0,a_0)}V^{\pi_\theta}(s_1)\big)\right]\\
&=\gamma\mathbb{E}_{s_0\sim\rho,a_0\sim\pi_\theta(\cdot|s_0),s_1\sim P(\cdot|s_0,a_0)}\left[\nabla_\theta V^{\pi_\theta}(s_1)\right],
\end{aligned}
$$

which is similar to what we want to bound, but a **discounted one-step look-ahead**.

Policy gradient derivation - recursion

Letting τ denote the trajectory following policy π_θ, by recursion,

$$\begin{aligned}
\nabla_\theta V^{\pi_\theta}(\rho) &= \mathbb{E}_{s_0\sim\rho, a_0\sim\pi_\theta(\cdot|s_0)}\left[\nabla_\theta \log \pi_\theta(a_0|s_0) Q^{\pi_\theta}(s_0,a_0)\right] \\
&\quad + \gamma \mathbb{E}_{s_0\sim\rho, a_0\sim\pi_\theta(\cdot|s_0), s_1\sim P(\cdot|s_0,a_0)}\left[\nabla_\theta V^{\pi_\theta}(s_1)\right] \\
&= \mathbb{E}_{(s_0,a_0)\sim\tau}\left[\nabla_\theta \log \pi_\theta(a_0|s_0) Q^{\pi_\theta}(s_0,a_0)\right] \\
&\quad + \gamma \mathbb{E}_{(s_0,a_0,s_1,a_1)\sim\tau}\left[\nabla_\theta \log \pi_\theta(a_1|s_1) Q^{\pi_\theta}(s_1,a_1)\right] + \ldots \\
&= \mathbb{E}_{(s_t,a_t)_{t\geq 0}\sim\tau}\left[\sum_{t=0}^{\infty} \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) Q^{\pi_\theta}(s_t,a_t)\right] \\
&= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\rho^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)}\Big[Q^{\pi_\theta}(s,a)\nabla \log \pi_\theta(a|s)\Big],
\end{aligned}$$

where $d_\rho^{\pi_\theta}$ is the **state visitation distribution:**

$$d_{s_0}^{\pi}(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mathbb{P}^{\pi}(s_t = s \,|\, s_0), \qquad d_\rho^{\pi}(s) = \mathbb{E}_{s_0\sim\rho}[d_{s_0}^{\pi}(s)].$$

The policy gradient theorem

Theorem 1 (Policy gradient theorem)

The policy gradient can be evaluated via

$$\nabla_\theta V^{\pi_\theta}(\rho) = \mathbb{E}_{(s_t,a_t)_{t\geq 0}\sim\tau}\left[\sum_{t=0}^{\infty}\gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) Q^{\pi_\theta}(s_t,a_t)\right]$$
$$= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\rho^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)}\left[Q^{\pi_\theta}(s,a)\nabla \log \pi_\theta(a|s)\right],$$

where

- $d_\rho^{\pi_\theta}$ *is the state visitation distribution,*
- $\nabla \log \pi_\theta(a|s)$ *is the score function.*

REINFORCE

The policy gradient theorem provides an effective scheme for policy gradient evaluation.

$$\nabla_\theta V^{\pi_\theta}(\rho) = \mathbb{E}_{(s_t,a_t)_{t\geq 0}\sim\tau}\left[\sum_{t=0}^{\infty}\gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) Q^{\pi_\theta}(s_t,a_t)\right]$$

REINFORCE [Williams, 1992]: a simple stochastic gradient algorithm for policy optimization. Given the current policy π_θ,

- roll out (multiple) trajectories $\{s_t, a_t\}_{t=0}^{\infty}$ following π_θ
- estimate the value function $Q^{\pi_\theta}(s_t, a_t)$ via return-to-go $G_t = \sum_{k=t}^{\infty}\gamma^{k-t} r_k$
- update the policy

$$\theta \leftarrow \theta + \eta \sum_{t\geq 0}\gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) G_t$$

- repeat

Note: this recipe uses *on-policy* policy evaluation.

Examples of policy parameterization

Discrete action space: softmax parameterization with function approximation

$$\pi_\theta(a|s) \propto \exp(\phi(s,a)^\top \theta)$$

- $\phi(s,a)$ is the feature vector of each state-action pair;
- the score function $\nabla \log \pi_\theta(a|s) = \phi(s,a) - \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[\phi(s,\cdot)]$.

Continuous action space: Gaussian policy

$$a \sim \mathcal{N}(\mu(s), \sigma^2), \quad \mu(s) = \phi(s)^\top \theta$$

- $\phi(s)$ is the feature of each state;
- σ^2 is the variance (kept constant for simplicity);
- the score function $\nabla \log \pi_\theta(a|s) = \frac{(a-\mu(s))\phi(s)}{\sigma^2}$.

Baseline

The policy gradient can have high variance with limited samples.

Variance reduction: introducing a baseline

$$\nabla_\theta V^{\pi_\theta}(\rho) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\rho^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)}\Big[\left(Q^{\pi_\theta}(s,a) - b(s)\right)\nabla \log \pi_\theta(a|s)\Big],$$

to help minimize the variance:

$$\begin{aligned}
\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\Big[\nabla \log \pi_\theta(a|s)\Big] &= \sum_a \pi_\theta(a|s)\frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)} \\
&= \sum_a \nabla_\theta \pi_\theta(a|s) \\
&= \nabla_\theta \sum_a \pi_\theta(a|s) = 0
\end{aligned}$$

Baseline

Variance reduction: introducing a baseline

$$\nabla_\theta V^{\pi_\theta}(\rho) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\rho^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)}\Big[\left(Q^{\pi_\theta}(s,a) - b(s)\right)\nabla \log \pi_\theta(a|s)\Big],$$

to help minimize the variance.

- In practice, choose $b(s) = V^{\pi_\theta}(s)$, leading to

$$\nabla_\theta V^{\pi_\theta}(\rho) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\rho^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)}\Big[A^{\pi_\theta}(s,a)\nabla \log \pi_\theta(a|s)\Big]$$

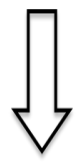
- $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$ is the advantage function.
- Instead of estimating $Q^\pi(s,a)$, directly estimate $A^\pi(s,a)$.

Global convergence of softmax policy gradient methods

Softmax policy gradient methods

Given an initial state distribution $s \sim \rho$, find policy π such that

$$\text{maximize}_\pi \quad V^\pi(\rho) := \mathbb{E}_{s\sim\rho}\left[V^\pi(s)\right]$$



softmax parameterization:

$$\pi_\theta(a|s) \propto \exp(\theta(s,a))$$

$$\text{maximize}_\theta \quad V^{\pi_\theta}(\rho) := \mathbb{E}_{s\sim\rho}\left[V^{\pi_\theta}(s)\right]$$

Policy gradient method

For $t = 0, 1, \cdots$

$$\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_\theta V^{\pi_\theta^{(t)}}(\rho)$$

where η *is the learning rate.*

Global convergence of the PG method

Exact gradient evaluation: suppose we can perfectly evaluate the gradient

$$\nabla_\theta V^{\pi_\theta^{(t)}}(\rho),$$

which takes the form (proof is left as an exercise!)

$$\nabla_\theta V^{\pi_\theta}(\rho) = \frac{1}{1-\gamma} d_\rho^{\pi_\theta}(s)\pi_\theta(a|s)A^{\pi_\theta}(s,a).$$

Does softmax policy gradient converge?

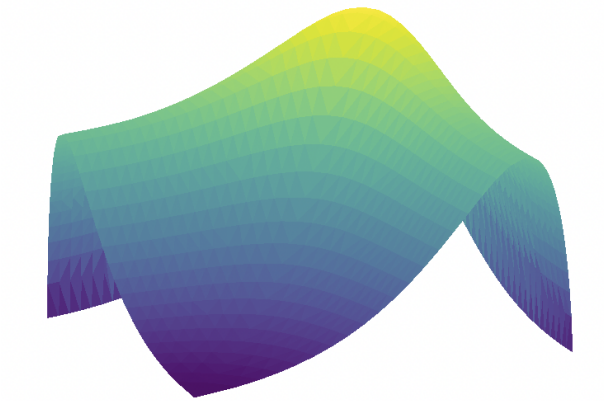
Theorem 2 ([Agarwal et al., 2021])

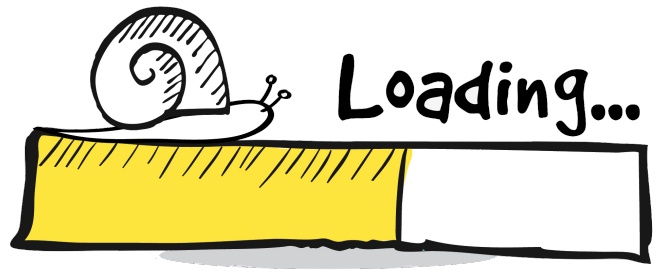
Assume ρ is strictly positive, i.e., $\rho(s) > 0$ for all states s. For $\eta \leq (1-\gamma)^3/8$, then we have that for all states s,

$$V^{(t)}(s) = V^{\pi_\theta^{(t)}}(s) \to V^\star(s), \qquad t \to \infty.$$

- Softmax PG finds the global optimal policy despite nonconcavity!

How fast does softmax PG converge?





- [Agarwal et al., 2021] showed that softmax PG converges asymptotically to the global optimal policy.
- [Mei et al., 2020b] showed that softmax PG converges to global opt in

$$c\big(|\mathcal{S}|, |\mathcal{A}|, \tfrac{1}{1-\gamma}, \cdots\big)\, O\big(\tfrac{1}{\varepsilon}\big) \text{ iterations}$$

How fast does softmax PG converge in finite time, really?

A negative message

Theorem 3 ([Li et al., 2023])

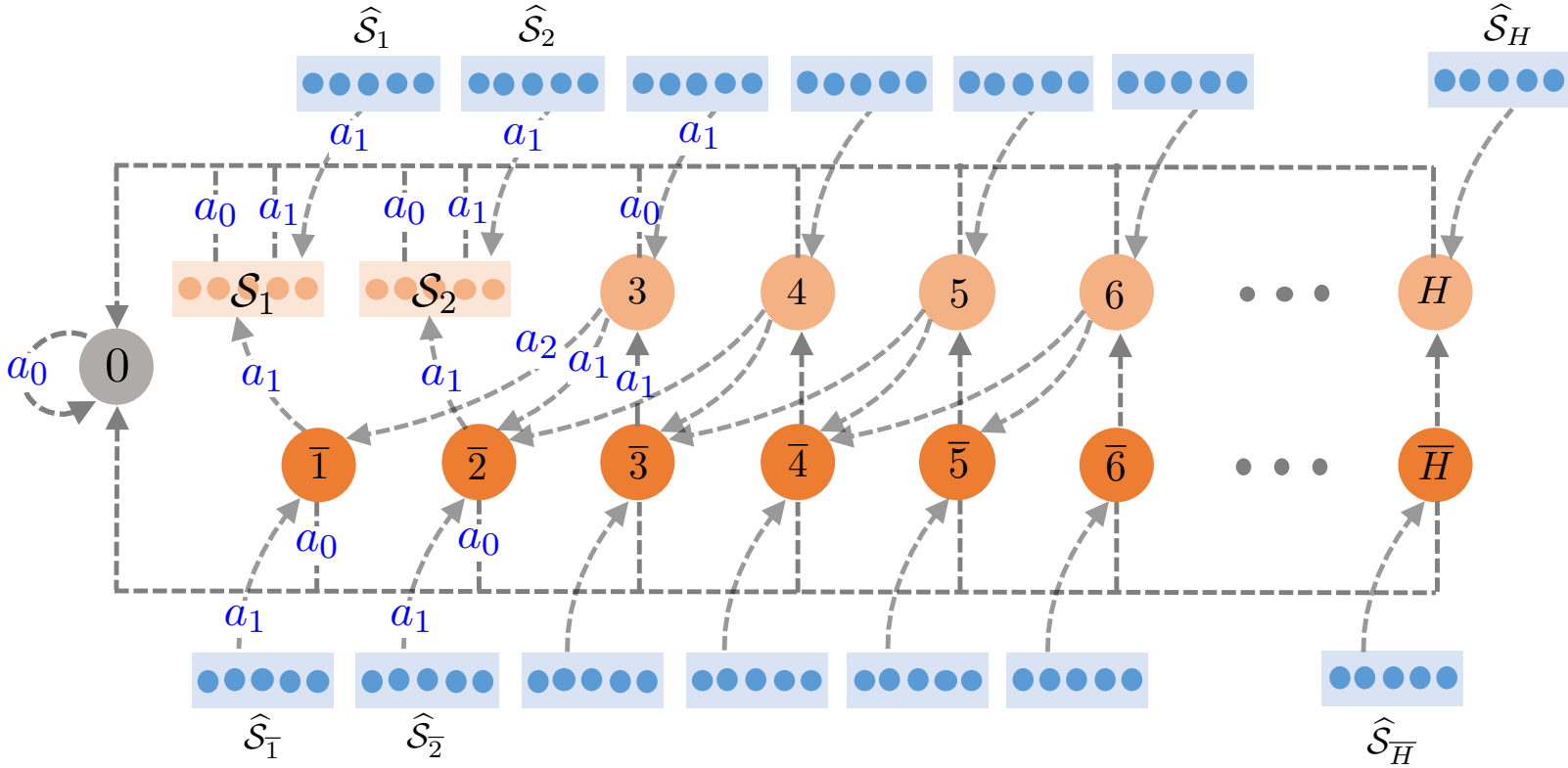
There exists an MDP s.t. it takes softmax PG at least

$$\frac{1}{\eta}\,|\mathcal{S}|^{2^{\Theta(\frac{1}{1-\gamma})}} \quad \textit{iterations}$$

to achieve $\|V^{(t)} - V^\star\|_\infty \leq 0.15$.

- Softmax PG can take (super)-exponential time to converge (in problems w/ large state space & long effective horizon)!
- Also hold for average sub-opt gap $\frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\left[V^{(t)}(s) - V^\star(s)\right]$.
- The convergence can be slow even for bandits [Mei et al., 2020a].

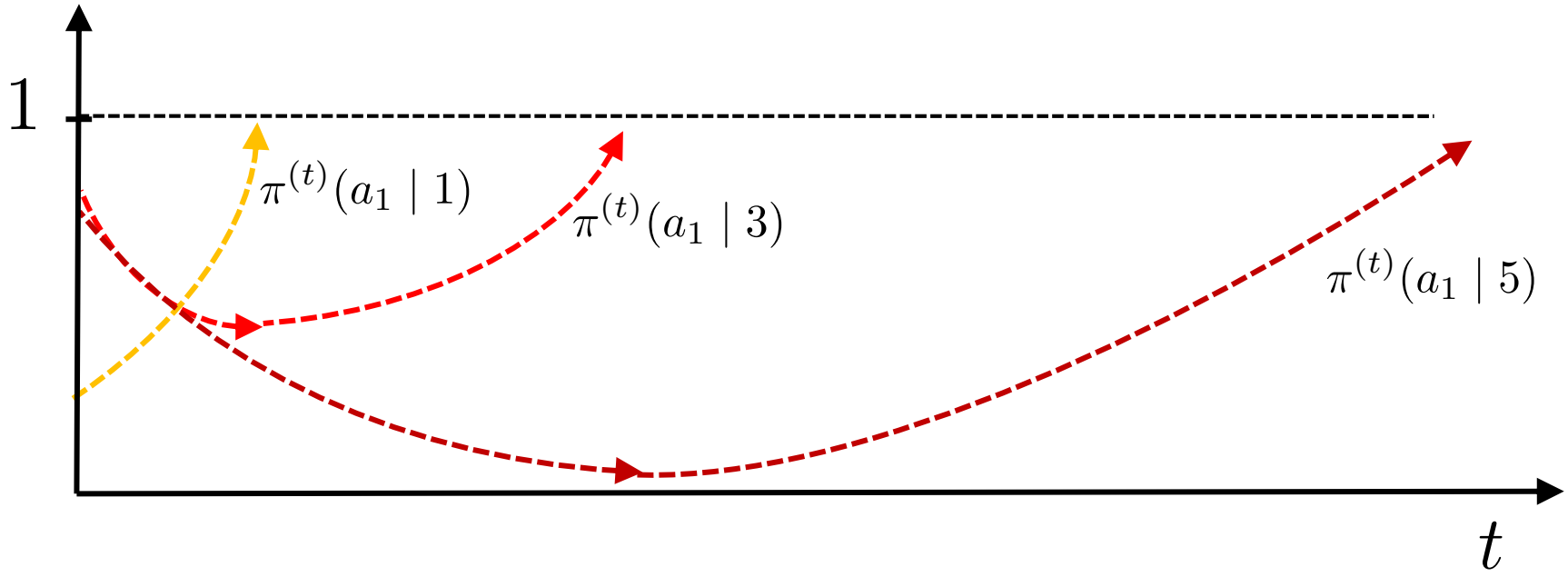
MDP construction for our lower bound



Key ingredients: for $3 \le s \le H \asymp \frac{1}{1-\gamma}$,

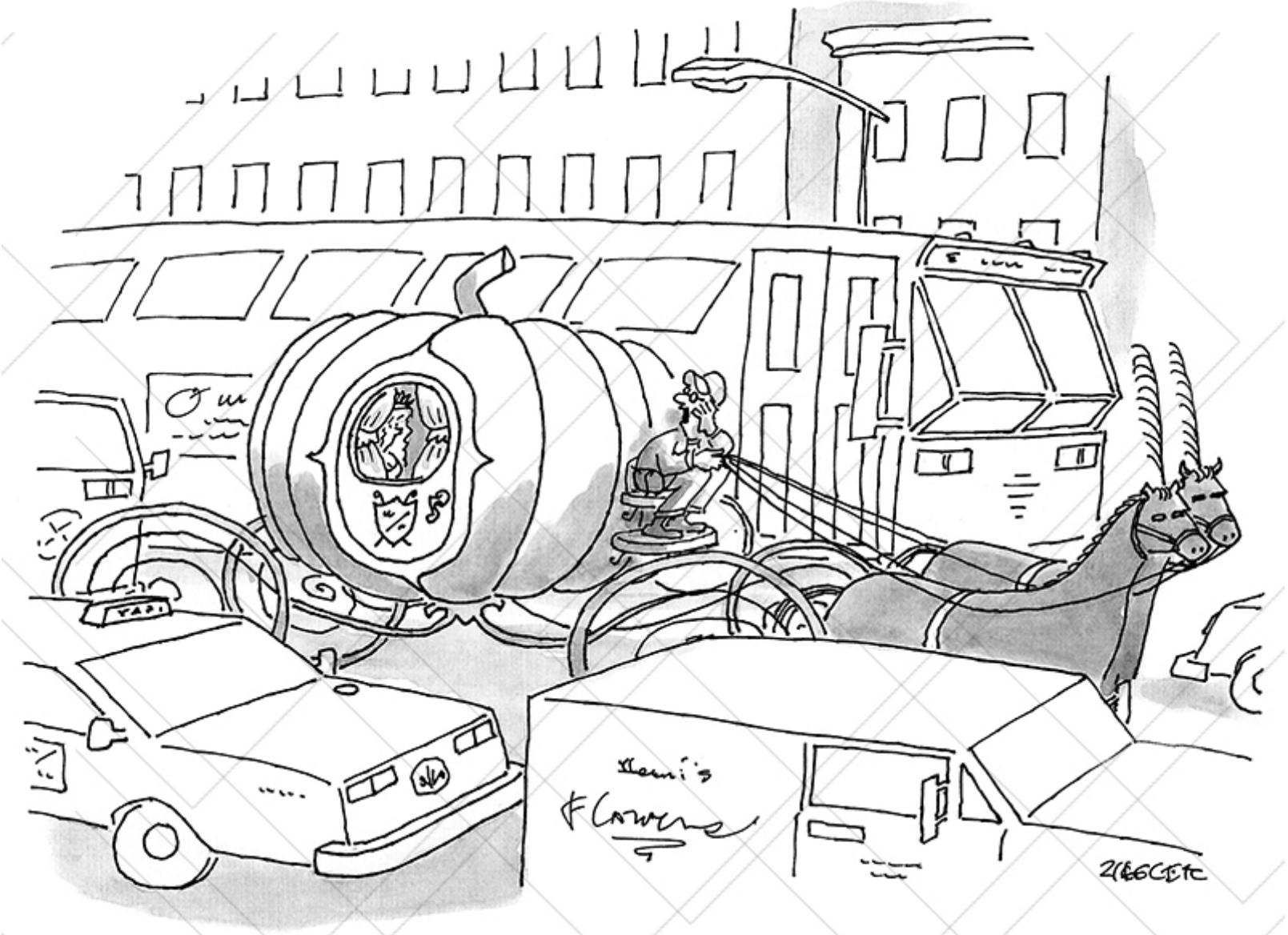
- $\pi^{(t)}(a_{\mathsf{opt}} \,|\, s)$ keeps decreasing until $\pi^{(t)}(a_{\mathsf{opt}} \,|\, s-2) \approx 1$

What is happening in our constructed MDP?



Convergence time for state s grows geometrically as s increases

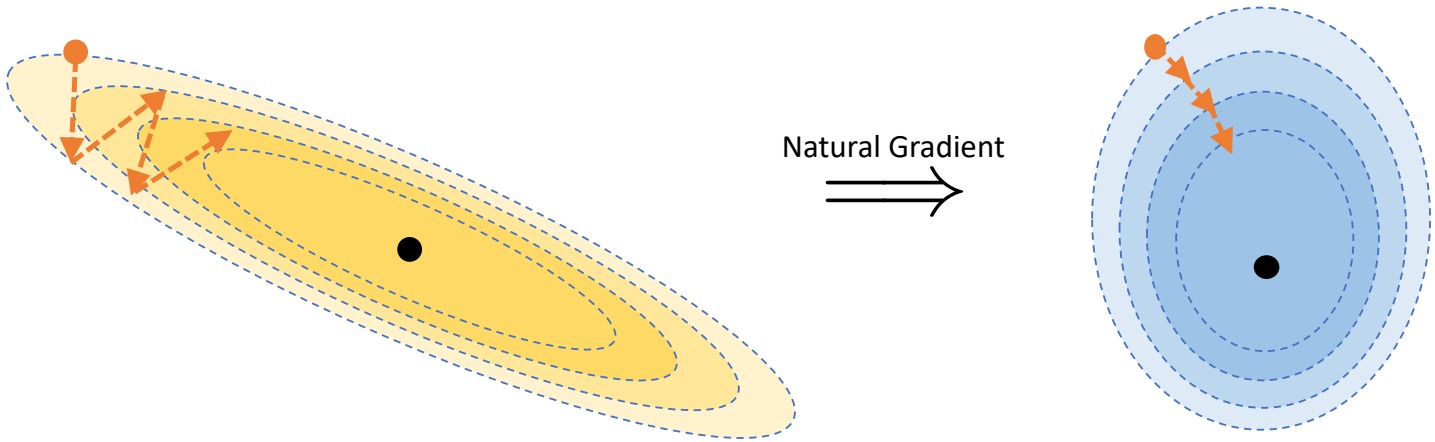
$$\text{convergence-time}(s) \gtrsim \big(\text{convergence-time}(s-2)\big)^{1.5}$$



"Seriously, lady, at this hour you'd make a lot better time taking the subway."

Natural policy gradient methods

Natural policy gradient



Natural policy gradient (NPG) method [Kakade, 2001]

For $t = 0, 1, \cdots$

$$\theta^{(t+1)} = \theta^{(t)} + \eta (\mathcal{F}_\rho^\theta)^\dagger \nabla_\theta V^{\pi_\theta^{(t)}}(\rho)$$

where η is the learning rate and $\mathcal{F}_\rho^\theta$ is the Fisher information matrix:

$$\mathcal{F}_\rho^\theta := \mathbb{E}_{s \sim d_\rho^{\pi_\theta}, a \sim \pi_\theta(\cdot|s)} \left[\big(\nabla_\theta \log \pi_\theta(a|s)\big) \big(\nabla_\theta \log \pi_\theta(a|s)\big)^\top \right].$$

Connection with TRPO

Trust Region Policy Optimization (TRPO) [Schulman et al., 2015] is a popular heuristic in training RL algorithms, with **KL regularization**

$$\mathsf{KL}(\pi_\theta^{(t)} \| \pi_\theta) \approx \frac{1}{2}(\theta - \theta^{(t)})^\top \mathcal{F}_\rho^\theta (\theta - \theta^{(t)})$$

via constrained or proximal terms:

$$\begin{aligned}\theta^{(t+1)} &= \operatorname*{argmax}_\theta V^{\pi_\theta^{(t)}}(\rho) + (\theta - \theta^{(t)})^\top \nabla_\theta V^{\pi_\theta^{(t)}}(\rho) - \eta \mathsf{KL}(\pi_\theta^{(t)} \| \pi_\theta) \\ &\approx \theta^{(t)} + \eta (\mathcal{F}_\rho^\theta)^\dagger \nabla_\theta V^{\pi_\theta^{(t)}}(\rho),\end{aligned}$$

leading to exactly NPG!

NPG $\approx$ TRPO!

NPG in the tabular setting

Natural policy gradient (NPG) method (Tabular setting)

For $t = 0, 1, \cdots$, NPG updates the policy via

$$\pi^{(t+1)}(\cdot|s) \propto \underbrace{\pi^{(t)}(\cdot|s)}_{\text{current policy}} \underbrace{\exp\left(\frac{\eta Q^{(t)}(s,\cdot)}{1-\gamma}\right)}_{\text{soft greedy}} \propto \pi^{(t)}(\cdot|s)\exp\left(\frac{\eta A^{(t)}(s,\cdot)}{1-\gamma}\right)$$

where $Q^{(t)} := Q^{\pi^{(t)}}$ and $A^{(t)} := A^{\pi^{(t)}}$ is the Q/advantage function of $\pi^{(t)}$, and $\eta > 0$ is the learning rate.

- the derivation is left as an exercise; see [Agarwal et al., 2019].
- invariant with the choice of ρ
- Reduces to policy iteration (PI) when $\eta = \infty$.

Global convergence of NPG

Theorem 4 ([Agarwal et al., 2021])

Set $\pi^{(0)}$ as a uniform policy. For all $t \geq 0$, we have

$$V^{(t)}(\rho) \geq V^{\star}(\rho) - \left(\frac{\log|\mathcal{A}|}{\eta} + \frac{1}{(1-\gamma)^2}\right)\frac{1}{t}.$$

Implication: set $\eta \geq (1-\gamma)^2 \log|\mathcal{A}|$, we find an ϵ-optimal policy within at most

$$\frac{2}{(1-\gamma)^2\epsilon} \text{ iterations.}$$

Global convergence at a sublinear rate independent of $|\mathcal{S}|$, $|\mathcal{A}|$!

Key ingredients of the proof

Lemma 5 (Performance difference lemma)

For all policies π, π' and distributions ρ over $\mathcal{S}$,

$$V^{\pi}(\rho) - V^{\pi'}(\rho) = \frac{1}{1-\gamma}\mathbb{E}_{s'\sim d_{\rho}^{\pi}}\mathbb{E}_{a'\sim\pi(\cdot|s')}\left[A^{\pi'}(s',a')\right].$$

- measures the performance difference for any pairs of policies

Lemma 6 (Policy improvement of NPG)

$$V^{(t+1)}(\rho) - V^{(t)}(\rho) \geq \frac{(1-\gamma)}{\eta}\mathbb{E}_{s\sim\rho}\log Z_t(s) \geq 0$$

where $Z_t(s) = \sum_a \pi^{(t)}(a|s)\exp\left(\eta A^{(t)}(s,a)/(1-\gamma)\right)$.

- monotonic performance improvement of NPG

Step 1: bounding the optimality gap

Denote $d^\star := d_\rho^\star$, and $\pi_s := \pi(\cdot|s)$. By the performance difference lemma,

$$
\begin{aligned}
& V^\star(\rho) - V^{(t)}(\rho) \\
&= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^\star}\sum_a \pi^\star(a|s)A^{(t)}(s,a) \\
&= \frac{1}{\eta}\mathbb{E}_{s\sim d^\star}\sum_a \pi^\star(a|s)\log\frac{\pi^{(t+1)}(a|s)Z_t(s)}{\pi^{(t)}(a|s)} \\
&= \frac{1}{\eta}\mathbb{E}_{s\sim d^\star}\left(\mathsf{KL}(\pi_s^\star\|\pi_s^{(t)}) - \mathsf{KL}(\pi_s^\star\|\pi_s^{(t+1)}) + \sum_a \pi^\star(a|s)\log Z_t(s)\right) \\
&= \frac{1}{\eta}\mathbb{E}_{s\sim d^\star}\left(\mathsf{KL}(\pi_s^\star\|\pi_s^{(t)}) - \mathsf{KL}(\pi_s^\star\|\pi_s^{(t+1)}) + \log Z_t(s)\right).
\end{aligned}
$$

Step 2: telescoping

By the improvement lemma $V^{(t+1)}(\rho) \geq V^{(t)}(\rho)$,

$$
\begin{aligned}
V^\star(\rho) - V^{(T-1)}(\rho) &\leq \frac{1}{T}\sum_{t=0}^{T-1}\Big(V^\star(\rho) - V^{(t)}(\rho)\Big) \\
&= \frac{1}{\eta T}\sum_{t=0}^{T-1}\mathbb{E}_{s\sim d^\star}\left(\mathsf{KL}(\pi_s^\star \| \pi_s^{(t)}) - \mathsf{KL}(\pi_s^\star \| \pi_s^{(t+1)}) + \log Z_t(s)\right) \\
&\leq \frac{1}{\eta T}\mathbb{E}_{s\sim d^\star}\mathsf{KL}(\pi_s^\star \| \pi_s^{(0)}) + \frac{1}{\eta T}\sum_{t=0}^{T-1}\mathbb{E}_{s\sim d^\star}\log Z_t(s),
\end{aligned}
$$

where the second term is bounded by the policy improvement lemma

$$
\begin{aligned}
\frac{1}{\eta}\sum_{t=0}^{T-1}\mathbb{E}_{s\sim d^\star}\log Z_t(s) &\leq \frac{1}{1-\gamma}\sum_{t=0}^{T-1}\Big(V^{(t+1)}(d^\star) - V^{(t)}(d^\star)\Big) \\
&\leq \frac{1}{1-\gamma}\Big(V^{(T)}(d^\star) - V^{(0)}(d^\star)\Big)
\end{aligned}
$$

Step 3: finishing up

Putting the above together,

$$
\begin{aligned}
& V^\star(\rho) - V^{(T-1)}(\rho) \\
& \leq \frac{1}{\eta T} \mathbb{E}_{s \sim d^\star} \mathsf{KL}(\pi_s^\star \| \pi_s^{(0)}) + \frac{1}{(1-\gamma)T} \left(V^{(T)}(d^\star) - V^{(0)}(d^\star) \right) \\
& \leq \frac{\log |\mathcal{A}|}{\eta T} + \frac{1}{(1-\gamma)^2 T},
\end{aligned}
$$

where we used $\mathsf{KL}(\pi_s^\star \| \pi_s^{(0)}) \leq \log |\mathcal{A}|$ and $V \leq \frac{1}{1-\gamma}$.

Proof of Lemma 6

Proof of $\log Z_t(s) \geq 0$**:**

$$\begin{aligned}
\log Z_t(s) &= \log \sum_a \pi^{(t)}(a|s) \exp\Big(\eta A^{(t)}(s,a)/(1-\gamma)\Big) \\
&\geq \sum_a \pi^{(t)}(a|s) \log \exp\Big(\eta A^{(t)}(s,a)/(1-\gamma)\Big) \qquad \text{(Jensen's inequality)} \\
&= \frac{\eta}{1-\gamma} \sum_a \pi^{(t)}(a|s) A^{(t)}(s,a) \\
&= \frac{\eta}{1-\gamma} \sum_a \pi^{(t)}(a|s) (Q^{\pi^{(t)}}(s,a) - V^{\pi^{(t)}}(s)) \\
&= 0
\end{aligned}$$

Proof of Lemma 6

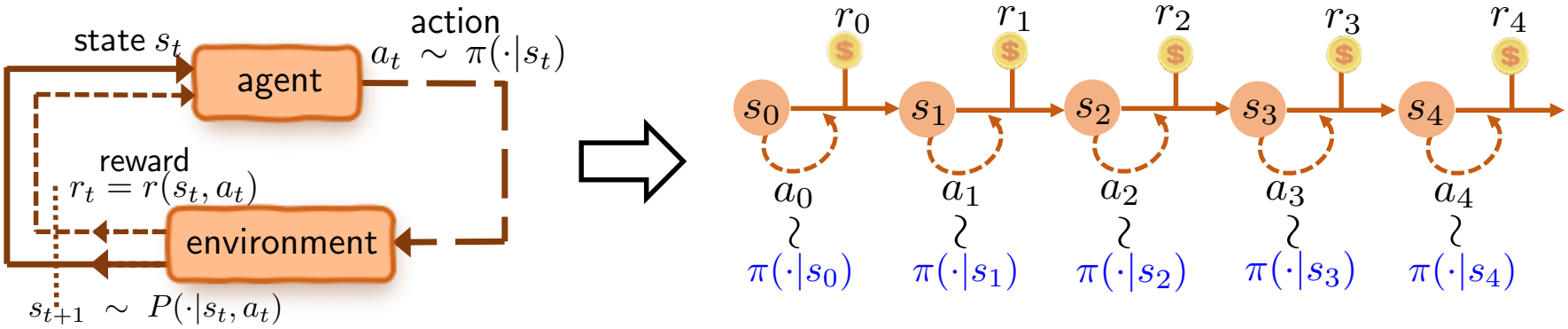
Bounding $V^{(t+1)}(\rho) - V^{(t)}(\rho)$: by the performance difference lemma,

$$\begin{aligned}
V^{(t+1)}(\rho) - V^{(t)}(\rho) &= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\rho^{(t+1)}} \sum_a \pi^{(t+1)}(a|s) A^{(t)}(s,a) \\
&= \frac{1}{\eta}\mathbb{E}_{s\sim d_\rho^{(t+1)}} \sum_a \pi^{(t+1)}(a|s) \log \frac{\pi^{(t+1)}(a|s) Z_t(s)}{\pi^{(t)}(a|s)} \\
&= \frac{1}{\eta}\mathbb{E}_{s\sim d_\rho^{(t+1)}} \mathsf{KL}(\pi^{(t+1)}(s) \| \pi^{(t)}(s)) + \frac{1}{\eta}\mathbb{E}_{s\sim d_\rho^{(t+1)}} \log Z_t(s) \\
&\geq \frac{(1-\gamma)}{\eta}\mathbb{E}_{s\sim\rho} \log Z_t(s),
\end{aligned}$$

where we use $d_\rho^{(t+1)} \geq (1-\gamma)\rho$ and $\log Z_t(s) \geq 0$.

Entropy/KL-regularized policy optimization

Entropy regularization



To encourage exploration, promote the stochasticity of the policy using the **"soft"** value function (Williams and Peng, 1991):

$$\forall s \in \mathcal{S}: \qquad V_\tau^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \big(r_t + \tau \mathcal{H}(\pi(\cdot|s_t))\big) \,\Big|\, s_0 = s\right]$$

where $\mathcal{H}$ is the Shannon entropy, and $\tau \geq 0$ is the reg. parameter.

$$\text{maximize}_\theta \quad V_\tau^{\pi_\theta}(\rho) := \mathbb{E}_{s\sim\rho}\left[V_\tau^{\pi_\theta}(s)\right]$$

KL regularization

Sometimes we want to optimize the policy while keeping it not too far way from some reference/base/teacher policy π_{ref}, which gives rise to a KL-regularized objective function (related to distillation):

$$\forall s \in \mathcal{S}: \qquad V_\tau^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \big(r_t - \tau \mathsf{KL}(\pi(\cdot|s_t) \,\|\, \pi_{\text{ref}}(\cdot|s_t))\big) \,\middle|\, s_0 = s\right]$$

where KL is the KL divergence, and $\tau \geq 0$ is the reg. parameter.

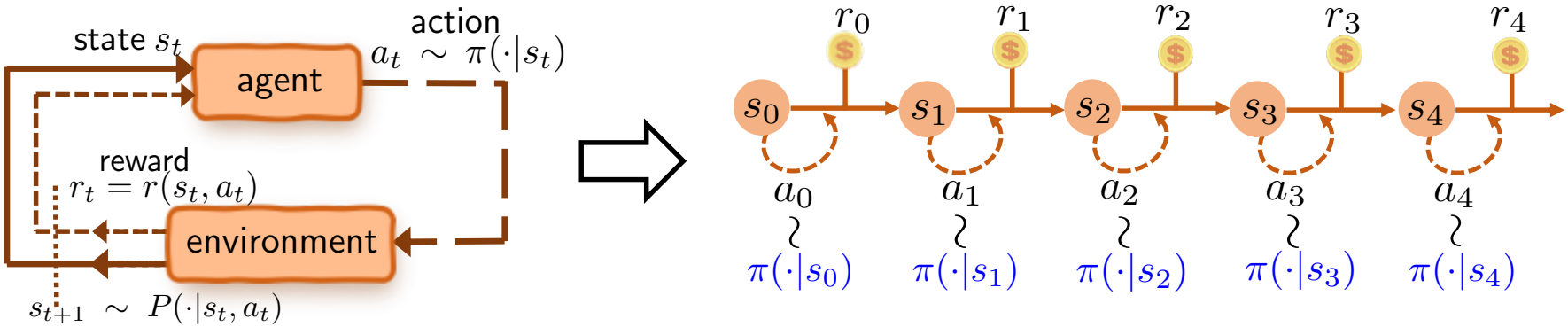
- Entropy regularization is a special case of KL-regularization:

$$\mathsf{KL}(\pi(\cdot|s) \,\|\, \mathsf{unif}) = \log|\mathcal{A}| - \mathcal{H}(\pi(\cdot|s))$$

- The KL regularization balances diversity and closeness to π_{ref}:

$$-\mathsf{KL}(\pi(\cdot|s) \,\|\, \pi_{\text{ref}}(\cdot|s)) = \mathcal{H}(\pi(\cdot|s)) + \mathbb{E}_\pi[\log \pi_{\text{ref}}]$$

Policy gradient of KL-regularized objectives



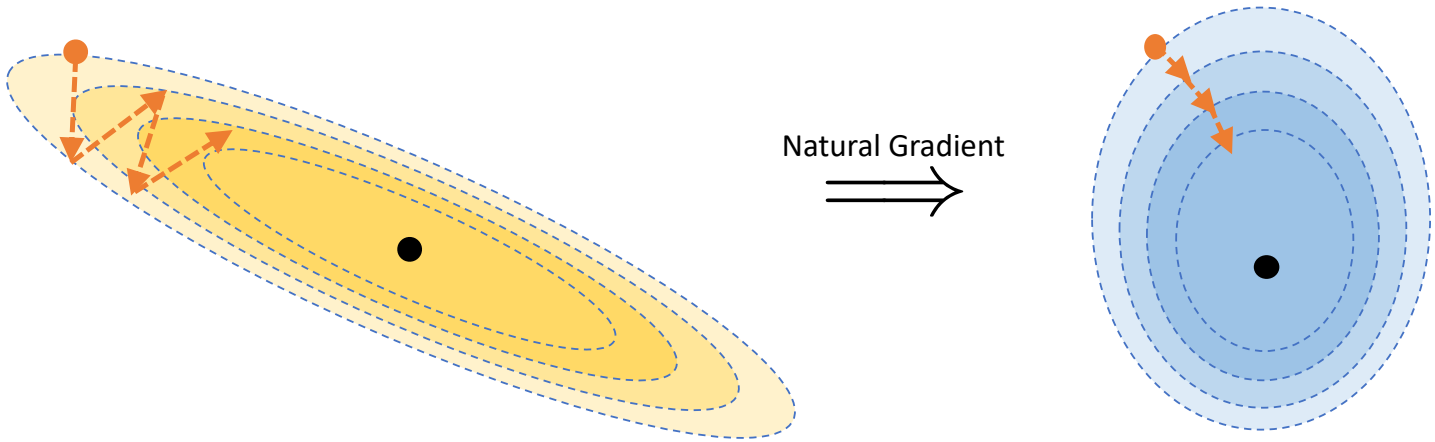
$$V_\tau^\pi(\rho) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \big(r_t + \tau \mathcal{H}(\pi(\cdot|s_t))\big) \,\Big|\, s_0 = s\right]$$

Vanilla policy gradient: just replace $r(s_t, a_t)$ by $r(s_t, a_t) - \tau \log \frac{\pi_\theta(a_t|s_t)}{\pi_{\text{ref}}(a_t|s_t)}$?

Yes (!), since for fixed π_{ref} and $a_t \sim \pi_\theta(\cdot|s_t)$,

$$\mathbb{E}\left[\nabla_\theta \left(-\tau \log \frac{\pi_\theta(a_t|s_t)}{\pi_{\text{ref}}(a_t|s_t)}\right) \,\Big|\, s_t\right] = -\tau\, \mathbb{E}[\nabla_\theta \log \pi_\theta(a_t|s_t) \mid s_t] = 0.$$

Entropy-regularized NPG



Entropy-regularized NPG

For $t = 0, 1, \cdots$

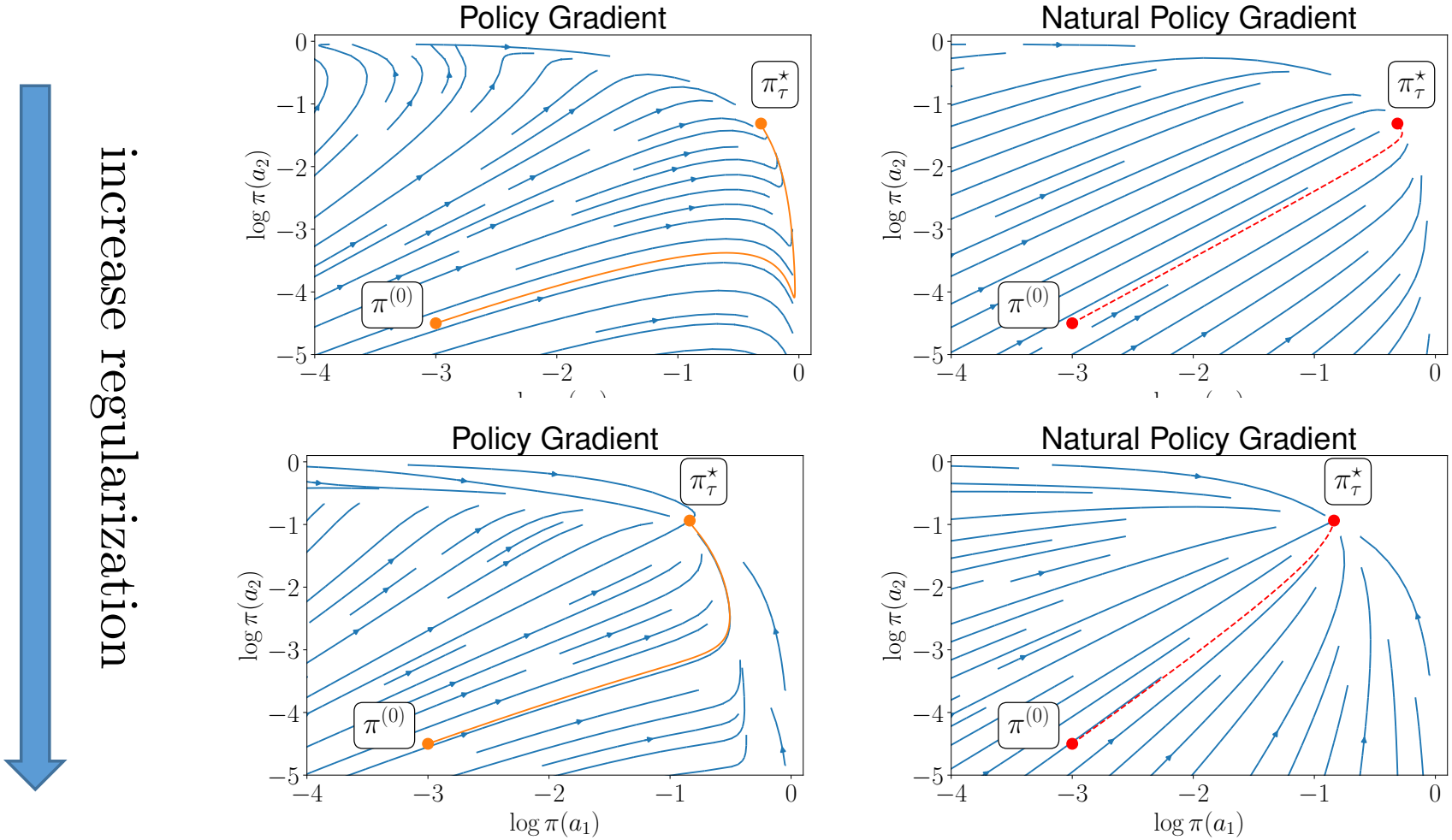
$$\theta^{(t+1)} = \theta^{(t)} + \eta (\mathcal{F}_\rho^\theta)^\dagger \nabla_\theta V_\tau^{\pi_\theta^{(t)}}(\rho)$$

where η is the learning rate and $\mathcal{F}_\rho^\theta$ is the Fisher information matrix:

$$\mathcal{F}_\rho^\theta := \mathbb{E}\left[\big(\nabla_\theta \log \pi_\theta(a|s)\big)\big(\nabla_\theta \log \pi_\theta(a|s)\big)^\top\right].$$

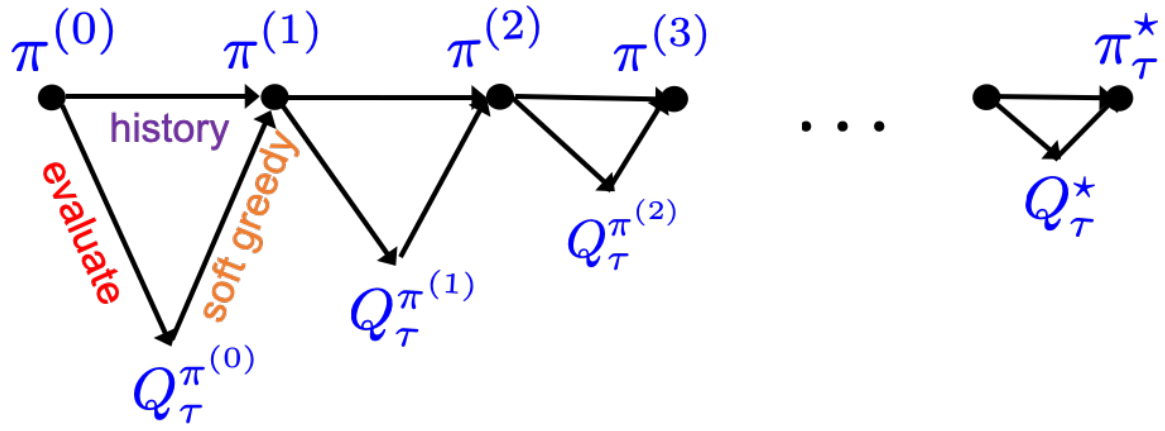
Entropy-regularized natural gradient helps!

Toy example: a bandit with 3 arms of rewards 1, 0.9 and 0.1.



Can we justify the efficacy of entropy-regularized NPG?

Entropy-regularized NPG in the tabular setting



Entropy-regularized NPG

For $t = 0, 1, \cdots$, the policy is updated via

$$\pi^{(t+1)}(\cdot|s) \propto \underbrace{\pi^{(t)}(\cdot|s)}_{\text{current policy}}^{1-\frac{\eta\tau}{1-\gamma}} \underbrace{\exp(Q_\tau^{(t)}(s,\cdot)/\tau)}_{\text{soft greedy}}^{\frac{\eta\tau}{1-\gamma}}$$

where $Q_\tau^{(t)} := Q_\tau^{\pi^{(t)}}$ is the soft Q-function of $\pi^{(t)}$, and $0 < \eta \leq \frac{1-\gamma}{\tau}$.

- invariant with the choice of ρ
- Reduces to soft policy iteration (SPI) when $\eta = \frac{1-\gamma}{\tau}$.

Linear convergence with exact gradient

Exact oracle: perfect evaluation of $Q_\tau^{\pi^{(t)}}$ given $\pi^{(t)}$;

Theorem 7 ([Cen et al., 2022])

For any learning rate $0 < \eta \le (1-\gamma)/\tau$, the entropy-regularized NPG updates satisfy

- **Linear convergence of soft value functions:**

$$\|V_\tau^\star - V_\tau^{(t+1)}\|_\infty \le 3C_1 \left(1-\eta\tau\right)^t,$$

- **Linear convergence of soft Q-functions:**

$$\|Q_\tau^\star - Q_\tau^{(t+1)}\|_\infty \le \gamma C_1 \left(1-\eta\tau\right)^t,$$

for all $t \ge 0$, where $Q_\tau^\star$ is the optimal soft Q-function, and

$$C_1 = \|Q_\tau^\star - Q_\tau^{(0)}\|_\infty + 2\tau\left(1 - \frac{\eta\tau}{1-\gamma}\right)\|\log\pi_\tau^\star - \log\pi^{(0)}\|_\infty.$$

Implications

To reach $\|Q_\tau^\star - Q_\tau^{(t+1)}\|_\infty \leq \epsilon$, the iteration complexity is at most

- **General learning rates ($0 < \eta < \frac{1-\gamma}{\tau}$):**

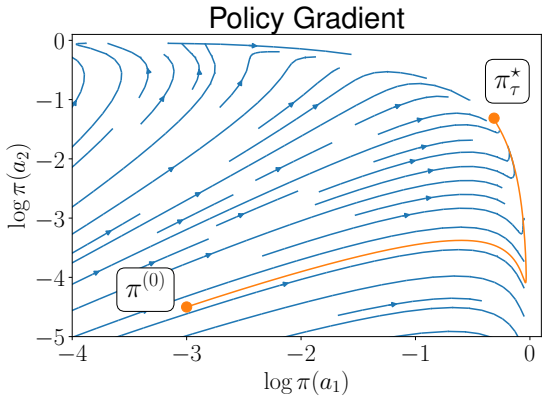
$$\frac{1}{\eta\tau} \log\left(\frac{C_1\gamma}{\epsilon}\right)$$

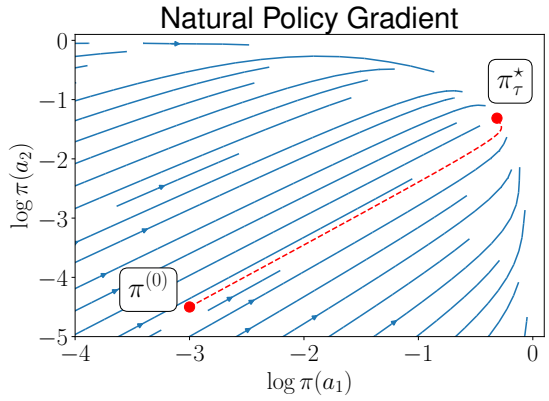
- **Soft policy iteration ($\eta = \frac{1-\gamma}{\tau}$):**

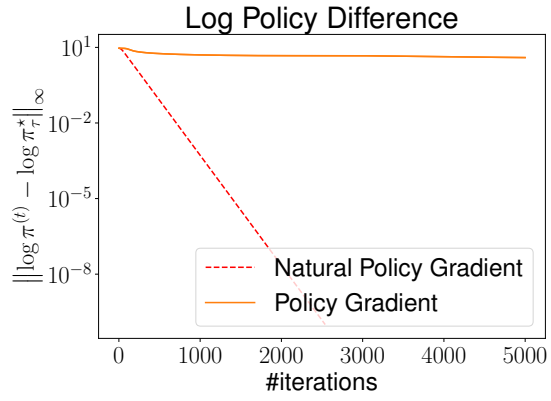
$$\frac{1}{1-\gamma} \log\left(\frac{\|Q_\tau^\star - Q_\tau^{(0)}\|_\infty \gamma}{\epsilon}\right)$$

Global linear convergence of entropy-regularized NPG
at a rate independent of $|\mathcal{S}|$, $|\mathcal{A}|$!

Comparisons with entropy-regularized PG







[Mei et al., 2020b] showed entropy-regularized PG achieves

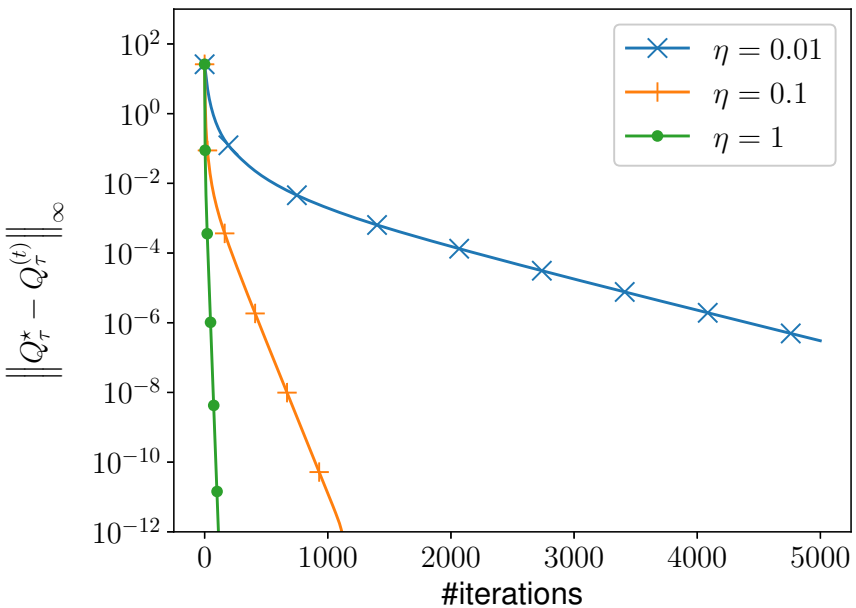
$$V_\tau^\star(\rho) - V_\tau^{(t)}(\rho) \le \left(V_\tau^\star(\rho) - V_\tau^{(0)}(\rho)\right)$$

$$\cdot \exp\left(-\frac{(1-\gamma)^4 t}{(8/\tau + 4 + 8\log|\mathcal{A}|)|\mathcal{S}|} \left\| \frac{d_\rho^{\pi_\tau^\star}}{\rho} \right\|_\infty^{-1} \min_s \rho(s) \underbrace{\left(\inf_{0 \le k \le t-1} \min_{s,a} \pi^{(k)}(a|s) \right)^2}_{\text{can be exponential in } |\mathcal{S}| \text{ and } \frac{1}{1-\gamma}} \right)$$

Much faster convergence of entropy-regularized NPG
at a **dimension-free** rate!

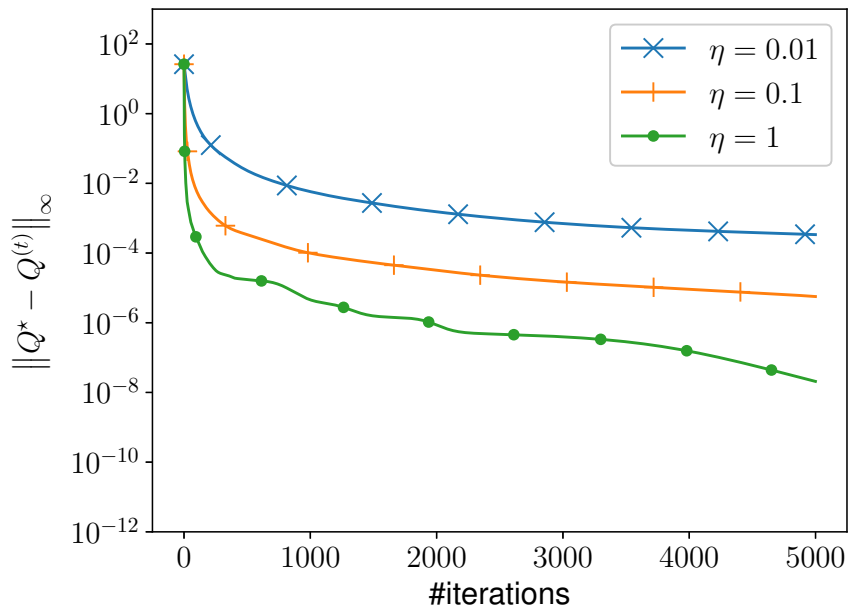
Comparison with unregularized NPG

Regularized NPG
$\tau = 0.001$



Linear rate: $\frac{1}{\eta\tau}\log\left(\frac{1}{\epsilon}\right)$
[Cen et al., 2022]

Vanilla NPG
$\tau = 0$



Sublinear rate: $\frac{1}{\min\{\eta,(1-\gamma)^2\}\epsilon}$
[Agarwal et al., 2020]

Entropy regularization enables fast convergence!

Entropy-regularized NPG with inexact gradients

Inexact oracle: inexact evaluation of $Q_\tau^{\pi^{(t)}}$ given $\pi^{(t)}$, which returns $\widehat{Q}_\tau^{(t)}$ that

$$\left\|\widehat{Q}_\tau^{(t)} - Q_\tau^{(t)}\right\|_\infty \leq \delta,$$

e.g., using sample-based estimators such as REINFORCE (Williams, 1992).

Inexact entropy-regularized NPG:

$$\pi^{(t+1)}(a|s) \;\propto\; \left(\pi^{(t)}(a|s)\right)^{1-\frac{\eta\tau}{1-\gamma}} \exp\left(\frac{\eta \widehat{Q}_\tau^{(t)}(s,a)}{1-\gamma}\right)$$

Question: Robustness of entropy-regularized NPG?

Linear convergence with inexact gradients

Theorem 8 ([Cen et al., 2022]; improved)

For any learning rate $0 < \eta \leq (1-\gamma)/\tau$, the entropy-regularized NPG updates achieve the same iteration complexity as the exact case, as long as

$$\delta \leq \frac{1-\gamma}{\gamma} \cdot \min\left\{\frac{\epsilon}{4}, \sqrt{\frac{\epsilon\tau}{2}}\right\}$$

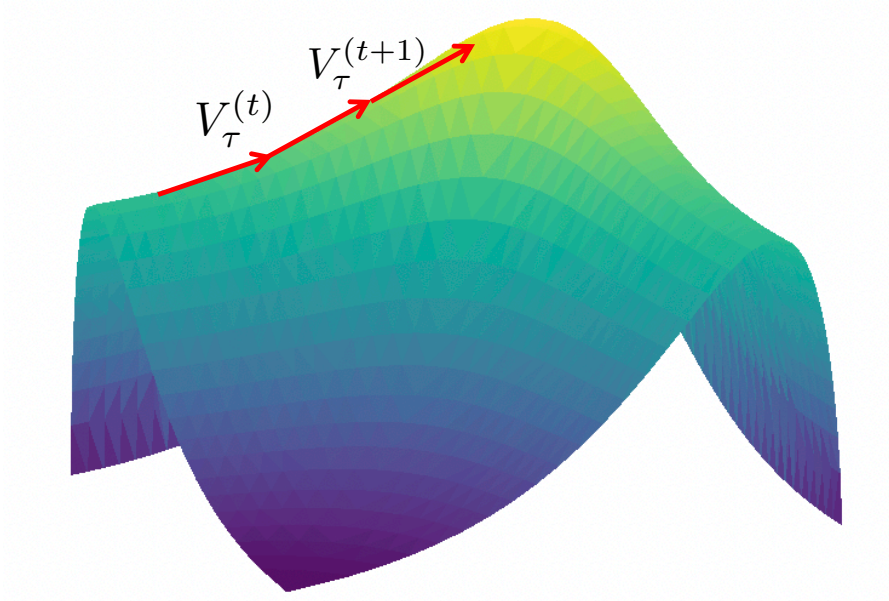
- **Crude sample complexity for finding an ϵ-optimal policy in the original MDP using a generative model:**

$$\widetilde{\mathcal{O}}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^7\epsilon^2}\right)$$

  - set $\tau = (1-\gamma)\epsilon/\log|\mathcal{A}|$;
  - in a generative model takes no larger than $\widetilde{O}(|\mathcal{S}||\mathcal{A}|(1-\gamma)^{-3}\delta^{-2})$ samples to achieve δ-accurate estimate of $Q_\tau^{(t)}$ per iteration;

A glimpse of the analysis

A key lemma: monotonic performance improvement



$$V_\tau^{(t+1)}(\rho) - V_\tau^{(t)}(\rho) = \mathbb{E}_{s\sim d_\rho^{(t+1)}}\left[\left(\frac{1}{\eta} - \frac{\tau}{1-\gamma}\right)\underbrace{\mathsf{KL}\left(\pi^{(t+1)}(\cdot|s)\,\middle\|\,\pi^{(t)}(\cdot|s)\right)}_{\text{KL divergence}}\right.$$

discounted state visitation distribution

$$\left. + \frac{1}{\eta}\underbrace{\mathsf{KL}\left(\pi^{(t)}(\cdot|s)\,\middle\|\,\pi^{(t+1)}(\cdot|s)\right)}_{\text{KL divergence}}\right]$$

Implication: monotonic improvement of $V_\tau(s)$ and $Q_\tau(s,a)$.

Recall: Bellman's optimality principle

Bellman operator

$$\mathcal{T}(Q)(s,a) := \underbrace{r(s,a)}_{\text{immediate reward}} + \gamma \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} \underbrace{\left[\max_{a' \in \mathcal{A}} Q(s',a')\right]}_{\text{next state's value}}$$

- one-step look-ahead

Bellman equation: $Q^\star$ is *unique* solution to

$$\mathcal{T}(Q^\star) = Q^\star$$

γ-contraction of Bellman operator:

$$\|\mathcal{T}(Q_1) - \mathcal{T}(Q_2)\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty$$



Richard Bellman

A key operator: soft Bellman operator

Soft Bellman operator

$$\mathcal{T}_\tau(Q)(s,a) := \underbrace{r(s,a)}_{\text{immediate reward}} + \gamma \mathop{\mathbb{E}}_{s' \sim P(\cdot|s,a)} \left[\max_{\pi(\cdot|s')} \mathop{\mathbb{E}}_{a' \sim \pi(\cdot|s')} \left[\underbrace{Q(s',a')}_{\text{next state's value}} - \underbrace{\tau \log \pi(a'|s')}_{\text{entropy}} \right] \right],$$

Soft Bellman equation: $Q_\tau^\star$ is *unique* solution to

$$\mathcal{T}_\tau(Q_\tau^\star) = Q_\tau^\star$$

γ-contraction of soft Bellman operator:

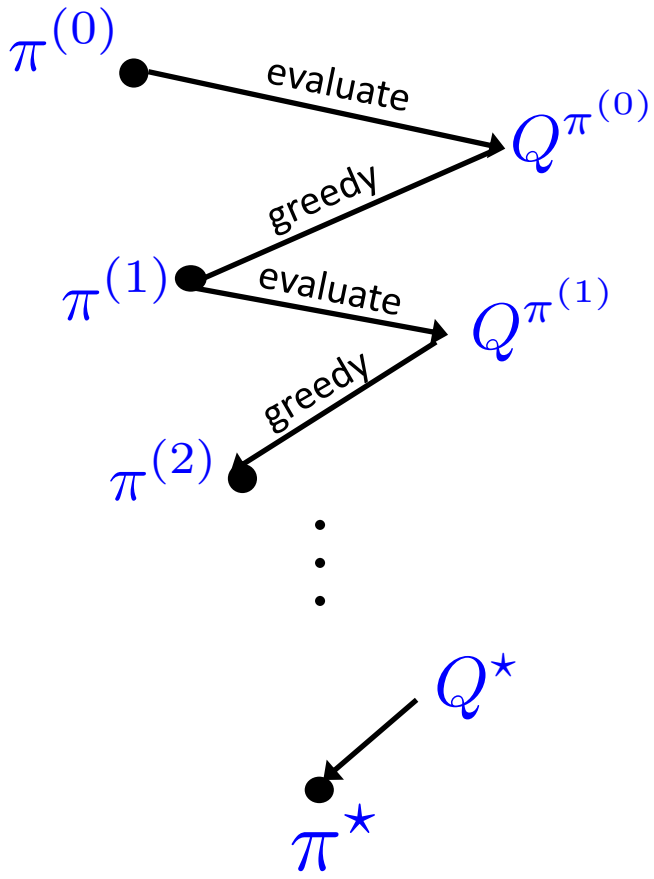
$$\|\mathcal{T}_\tau(Q_1) - \mathcal{T}_\tau(Q_2)\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty$$



Richard Bellman

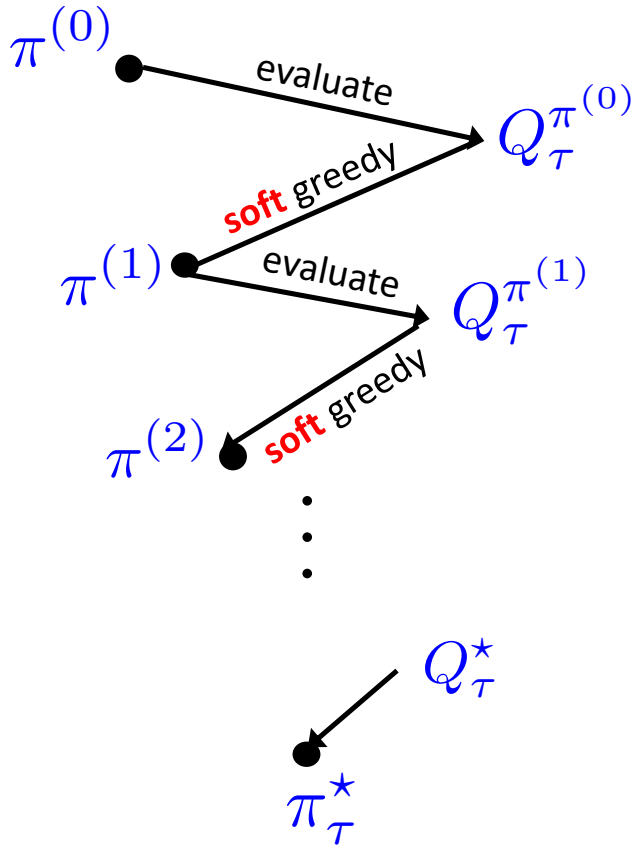
Analysis of soft policy iteration ($\eta = \frac{1-\gamma}{\tau}$)

Policy iteration



Bellman operator

Soft policy iteration



Soft Bellman operator

A key linear system: general learning rates

Let $x_t := \begin{bmatrix} \left\|Q_\tau^\star - Q_\tau^{(t)}\right\|_\infty \\ \left\|Q_\tau^\star - \tau \log \xi^{(t)}\right\|_\infty \end{bmatrix}$ and $y := \begin{bmatrix} \left\|Q_\tau^{(0)} - \tau \log \xi^{(0)}\right\|_\infty \\ 0 \end{bmatrix}$,

where $\xi^{(t)} \propto \pi^{(t)}$ is an auxiliary sequence, then

$$x_{t+1} \leq A x_t + \gamma \left(1 - \frac{\eta\tau}{1-\gamma}\right)^{t+1} y,$$

where

$$A := \begin{bmatrix} \gamma \\ 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{\eta\tau}{1-\gamma} & 1 - \frac{\eta\tau}{1-\gamma} \end{bmatrix}$$

is a rank-1 matrix with a non-zero eigenvalue $\underbrace{1-\eta\tau}_{\text{contraction rate!}}$.

Actor-critic methods

Recall: policy gradient methods

Recall the policy gradient expression

$$\nabla_\theta V^{\pi_\theta}(\rho) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\rho^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)}\Big[Q^{\pi_\theta}(s,a)\nabla\log\pi_\theta(a|s)\Big],$$

where

- $d_\rho^{\pi_\theta}$ is the state visitation distribution,
- $\nabla\log\pi_\theta(a|s)$ is the score function.

Function approximation in PG

How do we inject function approximation into policy gradient methods?

Answer: using a **critic** with function approximation

$$Q^{\pi_\theta}(s,a) \approx Q_w(s,a)$$

parameterized by some w.

Actor-critic framework

- **Critic:** update the parameter w of the Q-function $Q_w(s,a)$ by approximately minimizing

$$J_{\text{critic}}(w) = \mathbb{E}_{s\sim d_\rho^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)}\left[\left(Q_w(s,a) - Q^{\pi_\theta}(s,a)\right)^2\right]$$

- **Actor:** update the parameter θ of the policy π_θ, by moving along the policy gradient

$$\nabla_\theta V^{\pi_\theta}(\rho) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\rho^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)}\left[\cancel{Q^{\pi_\theta}(s,a)}\, Q_w(s,a)\nabla \log \pi_\theta(a|s)\right],$$

Reducing variance using a baseline

Instead of using $Q^{\pi_\theta}(s,a)$ in the policy gradient, we can use the advantage function

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

which helps reduce the variance.

- We can set the critic to estimate the advantage function instead
- **Key observation:** the TD error

$$\delta^{\pi_\theta} = r + \gamma V^{\pi_\theta}(s') - V^{\pi_\theta}(s)$$

is an unbiased estimate of the advantage function

$$\begin{aligned}\mathbb{E}[\delta^{\pi_\theta}|s,a] &= \mathbb{E}[r + \gamma V^{\pi_\theta}(s')|s,a] - V^{\pi_\theta}(s) \\ &= Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s) \\ &= A^{\pi_\theta}(s,a)\end{aligned}$$

Actor-critic with TD error

Use the TD error for policy gradient

$$\nabla_\theta V^{\pi_\theta}(\theta) = \mathbb{E}\left[\nabla_\theta \log \pi_\theta(s|a)\delta^{\pi_\theta}\right]$$

This only requires one set of critic parameter:

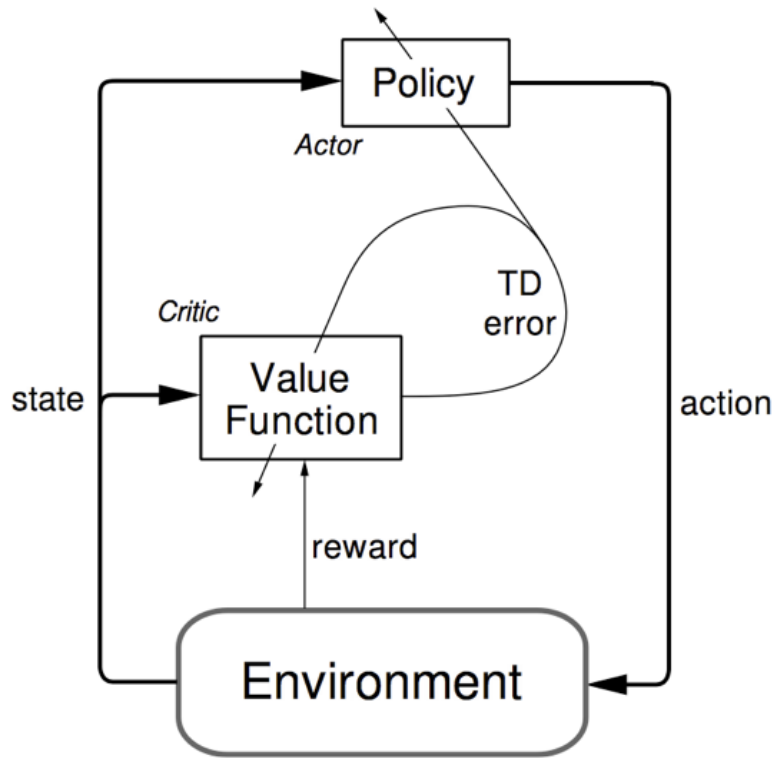
- Compute the TD error

$$\delta^{\pi_\theta} = r + \gamma V^{\pi_\theta}(s') - V^{\pi_\theta}(s)$$

- Update the policy parameter

$$\theta \leftarrow \theta + \beta\delta^{\pi_\theta}\nabla_\theta \log \pi_\theta(a|s)$$

where β is the learning rate.



References I


Agarwal, A., Jiang, N., Kakade, S. M., and Sun, W. (2019).
Reinforcement learning: Theory and algorithms.


Agarwal, A., Kakade, S., and Yang, L. F. (2020).
Model-based reinforcement learning with a generative model is minimax optimal.
Conference on Learning Theory, pages 67–83.


Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. (2021).
On the theory of policy gradient methods: Optimality, approximation, and distribution shift.
The Journal of Machine Learning Research, 22(1):4431–4506.


Cen, S., Cheng, C., Chen, Y., Wei, Y., and Chi, Y. (2022).
Fast global convergence of natural policy gradient methods with entropy regularization.
Operations Research, 70(4):2563–2578.


Kakade, S. M. (2001).
A natural policy gradient.
Advances in Neural Information Processing Systems, 14.


Li, G., Wei, Y., Chi, Y., and Chen, Y. (2023).
Softmax policy gradient methods can take exponential time to converge.
Mathematical Programming.

References II



Mei, J., Xiao, C., Dai, B., Li, L., Szepesvári, C., and Schuurmans, D. (2020a).
Escaping the gravitational pull of softmax.
Advances in Neural Information Processing Systems, 33:21130–21140.



Mei, J., Xiao, C., Szepesvari, C., and Schuurmans, D. (2020b).
On the global convergence rates of softmax policy gradient methods.
In *International Conference on Machine Learning*, pages 6820–6829. PMLR.



Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015).
Trust region policy optimization.
In *International conference on machine learning*, pages 1889–1897.



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