Policy Mirror Descent Inherently Explores Action Space

Yan Li

Georgia Institute of Technology

INFORMS Optimization Society Conference, 2024

Joint work with George Lan

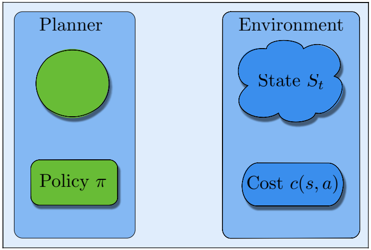
Markov Decision Process & Policy Optimization

Markov Decision Process

▷ **Sequential decision making over multiple timesteps ..**

Key elements

- policy π
- finite state space: $\mathcal{S}$
- finite action space: $\mathcal{A}$
- cost function c
- transition kernel $\mathbb{P}$

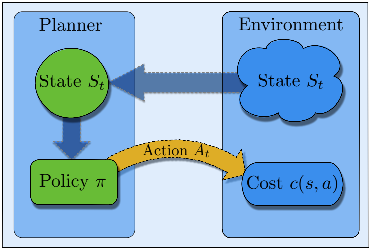


Markov Decision Process

▷ **Sequential decision making over multiple timesteps ..**

Key elements

- policy π
- finite state space: $\mathcal{S}$
- finite action space: $\mathcal{A}$
- cost function c
- transition kernel $\mathbb{P}$



Decision making:

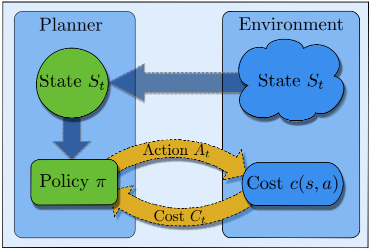
1. Observe current state S_t and feed into policy
2. Make A_t following distribution $\pi(\cdot|S_t)$

Markov Decision Process

▷ **Sequential decision making over multiple timesteps ..**

Key elements

- policy π
- finite state space: $\mathcal{S}$
- finite action space: $\mathcal{A}$
- cost function c
- transition kernel $\mathbb{P}$



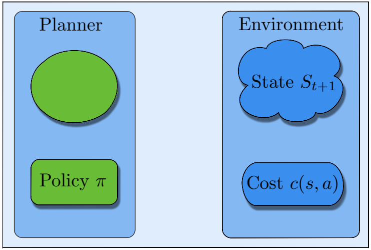
Observing loss: $C_t = c(S_t, A_t) \in [0, 1]$

Markov Decision Process

▷ **Sequential decision making over multiple timesteps ..**

Key elements

- policy π
- finite state space: $\mathcal{S}$
- finite action space: $\mathcal{A}$
- cost function c
- transition kernel $\mathbb{P}$



State transition: S_{t+1} follows distribution $\mathbb{P}(\cdot|S_t, A_t)$

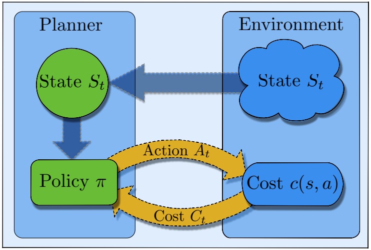
Repeat decision process ..

Markov Decision Process

▷ **Sequential decision making over multiple timesteps ..**

Key elements

- policy π
- finite state space: $\mathcal{S}$
- finite action space: $\mathcal{A}$
- cost function c
- transition kernel $\mathbb{P}$



Trajectory:

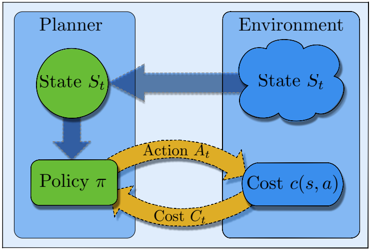
$$\{(S_0, A_0, C_0), (S_1, A_1, C_1), \ldots, (S_t, A_t, C_t), \ldots\}$$

Markov Decision Process

▷ **Sequential decision making over multiple timesteps ..**

Key elements

- policy π
- finite state space: $\mathcal{S}$
- finite action space: $\mathcal{A}$
- cost function c
- transition kernel $\mathbb{P}$



Performance (value function):

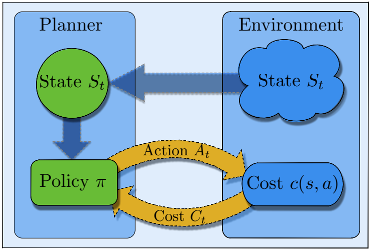
$$V_{\mathbb{P}}^{\pi}(s) = \mathbb{E}_{\mathbb{P}}^{\pi}\left[\sum_{t=0}^{\infty} \underbrace{\gamma^t C_t}_{\text{discounting future}} \,\middle|\, S_0 = s\right]$$

Markov Decision Process

▷ **Sequential decision making over multiple timesteps ..**

Key elements

- policy π
- finite state space: $\mathcal{S}$
- finite action space: $\mathcal{A}$
- cost function c
- transition kernel $\mathbb{P}$



Planning: find the optimal policy of

$$\min_{\pi} V^{\pi}_{\mathbb{P}}(s), \ \forall s \in \mathcal{S}$$

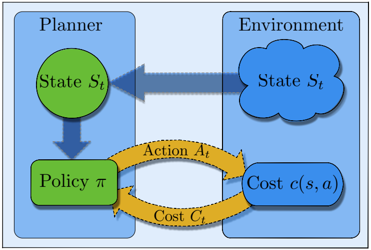
- Covers bandit as special case $(|\mathcal{S}| = 1, \gamma = 0)$

Markov Decision Process

▷ **Sequential decision making over multiple timesteps ..**

Key elements

- policy π
- finite state space: $\mathcal{S}$
- finite action space: $\mathcal{A}$
- cost function c
- transition kernel $\mathbb{P}$



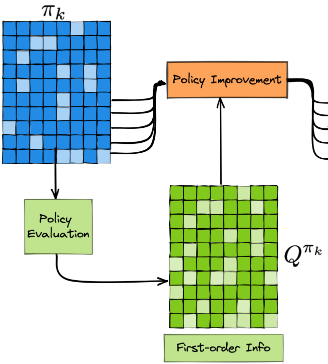
Planning with an equivalent objective:

$$\min_{\pi} f_\rho(\pi) = \sum_{s\in\mathcal{S}} \rho(s) V^{\pi}_{\mathbb{P}}(s) \quad \Rightarrow \quad \boxed{\text{Non-convex}}$$

- Covers bandit as special case ($|\mathcal{S}| = 1$, $\gamma = 0$)

Policy Gradients – Overview

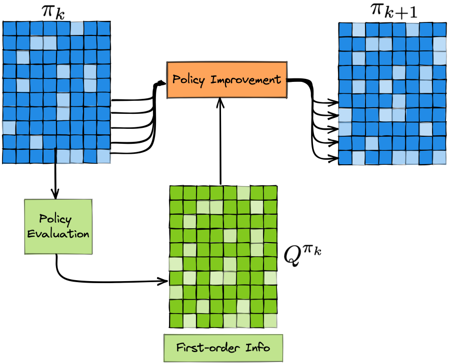
Policy Gradients - A Basic Skeleton



▷ **First-order policy optimization:**

1. $\text{Eval}(\pi_k) \to Q_{\mathbb{P}}^{\pi_k}$
2. Construct gradient information G_k
3. $\text{Update}(\pi_k, G_k) \to \pi_{k+1}$
4. Repeat ..

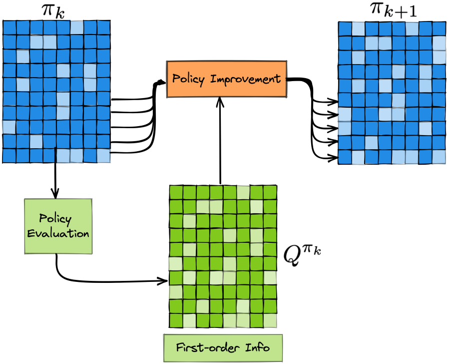
Policy Gradients - A Basic Skeleton



Q-function:

$$Q^{\pi}_{\mathbb{P}}(s,a) = \mathbb{E}^{\pi}_{\mathbb{P}}\left[\sum_{t=0}^{\infty} \gamma^t c(S_t, A_t) \middle| S_0 = s, A_0 = a\right]$$

Policy Gradients - A Basic Skeleton



⋆ **Challenges:**

- Non-convex landscape
- Model ($\mathbb{P}$ and c) can be unknown

Policy Gradients – Existing Development

1. Stochastic setting – unknown $\mathbb{P}$ & c
   - Agarwal, Kakade, Lee, Mahajan '19: $\mathcal{O}(1/\epsilon^4)$ samples
   - Shani, Efroni, Mannor '20: $\mathcal{O}(1/\epsilon^4)$ and $\mathcal{O}(1/\epsilon^3)$
   - Lan, '21: $\mathcal{O}(1/\epsilon^2)$ samples for entropy regularized MDPs

Policy Gradients – Existing Development

1. Stochastic setting – unknown $\mathbb{P}$ & c
   - Agarwal, Kakade, Lee, Mahajan '19: $\mathcal{O}(1/\epsilon^4)$ samples
   - Shani, Efroni, Mannor '20: $\mathcal{O}(1/\epsilon^4)$ and $\mathcal{O}(1/\epsilon^3)$
   - Lan, '21: $\mathcal{O}(1/\epsilon^2)$ samples for entropy regularized MDPs

Current status of policy gradients

An ϵ-optimal policy can be attained using $\mathcal{O}(1/\epsilon^2)$ samples, **IF ...**

"IF ..."

Tension Between Evaluation and Optimization

Classical Stochastic Policy Evaluation

▷ **Q-function:**

$$Q^{\pi}_{\mathbb{P}}(s,a) = \mathbb{E}^{\pi}_{\mathbb{P}}\left[\sum_{t=0}^{\infty} \gamma^t c(S_t, A_t) \middle| S_0 = s, A_0 = a\right]$$

▷ Classical-Eval(π_k) **with unknown** $\mathbb{P}$**:**

Classical Stochastic Policy Evaluation

▷ **Q-function:**

$$Q^{\pi}_{\mathbb{P}}(s,a) = \mathbb{E}^{\pi}_{\mathbb{P}}\left[\sum_{t=0}^{\infty} \gamma^t c(S_t, A_t) \middle| S_0 = s, A_0 = a\right]$$

▷ Classical-Eval(π_k) **with unknown $\mathbb{P}$:**

1. Deploy π_k, generate trajectory:

$$\xi = \{(S_0, A_0, C_0), (S_1, A_1, C_1), \ldots, (S_t, A_t, C_t), \ldots\}$$

where $A_t \sim \pi_k(\cdot|S_t),\ S_{t+1} \sim \mathbb{P}(\cdot|S_t, A_t)$

- We assume $\{S_t\}$ is ergodic (by no means trivial)

Classical Stochastic Policy Evaluation

▷ **Q-function:**

$$Q^{\pi}_{\mathbb{P}}(s,a) = \mathbb{E}^{\pi}_{\mathbb{P}}\left[\sum_{t=0}^{\infty} \gamma^t c(S_t, A_t) \middle| S_0 = s, A_0 = a\right]$$

▷ Classical-Eval(π_k) **with unknown $\mathbb{P}$:**

1. Deploy π_k, generate trajectory:

$$\xi = \{(S_0, A_0, C_0), (S_1, A_1, C_1), \ldots, (S_t, A_t, C_t), \ldots\}$$
where $A_t \sim \pi_k(\cdot|S_t),\ S_{t+1} \sim \mathbb{P}(\cdot|S_t, A_t)$

   - We assume $\{S_t\}$ is ergodic (by no means trivial)

2. Apply learning procedure that makes clever use of trajectories

Classical Stochastic Policy Evaluation

▷ **Q-function:**

$$Q^{\pi}_{\mathbb{P}}(s,a) = \mathbb{E}^{\pi}_{\mathbb{P}}\left[\sum_{t=0}^{\infty} \gamma^t c(S_t, A_t) \middle| S_0 = s, A_0 = a\right]$$

▷ Classical-Eval(π_k) **with unknown $\mathbb{P}$:**

1. Deploy π_k, generate trajectory:

$$\xi = \{(S_0, A_0, C_0), (S_1, A_1, C_1), \ldots, (S_t, A_t, C_t), \ldots\}$$

where $A_t \sim \pi_k(\cdot|S_t),\ S_{t+1} \sim \mathbb{P}(\cdot|S_t, A_t)$

   - We assume $\{S_t\}$ is ergodic (by no means trivial)

2. Apply learning procedure that makes clever use of trajectories
   - On-policy Monte-Carlo
   - On-policy temporal-difference (TD)

Requirement for Accurate Evaluation

▷ **Observation**: action with zero probability never gets explored

If $\pi_k(a|s) = 0 \quad \Rightarrow \quad (s, a)$ does not appear in trajectory ξ

$\Rightarrow \quad Q^{\pi_k}(s, a)$ not learnable

"Hopefully benign" assumption (IF ...)

$$\inf_{s \in \mathcal{S}, a \in \mathcal{A}} \pi_k(a|s) \geq \underline{\sigma} > 0, \text{ for any } k$$

"Policy π_k needs to explore every action"

Requirement for Accurate Evaluation

▷ **Observation**: action with zero probability never gets explored

$$\text{If } \pi_k(a|s) = 0 \quad \Rightarrow \quad (s,a) \text{ does not appear in trajectory } \xi$$
$$\Rrightarrow \quad Q^{\pi_k}(s,a) \text{ not learnable}$$

"Hopefully benign" assumption (IF ...)

$$\inf_{s\in\mathcal{S},a\in\mathcal{A}} \pi_k(a|s) \geq \underline{\sigma} > 0, \text{ for any } k$$

"Policy π_k needs to explore every action"

- Widely assumed in various form (Abbasi-Yadkori et al., '19; Xu et al., '20; Alacaoglu et al., '22; Liu et al., '20; many others)

Tension Between Evaluation and Optimization

Purpose of planning (structure of optimal policies)

$$\mathcal{A}^*(s) \coloneqq \mathrm{Argmin}_{a \in \mathcal{A}} Q^*(s,a) \implies \pi^*(a|s) = 0 \text{ if } a \notin \mathcal{A}^*(s)$$

"Optimal policy π^* <u>does not</u> explore every action"

Tension Between Evaluation and Optimization

Purpose of planning (structure of optimal policies)

$$\mathcal{A}^*(s) \coloneqq \mathrm{Argmin}_{a\in\mathcal{A}}\, Q^*(s,a) \implies \pi^*(a|s) = 0 \text{ if } a \notin \mathcal{A}^*(s)$$

"Optimal policy π^* <u>does not</u> explore every action"

- If $\pi_k \to \pi^*$ then $\underline{\sigma} = 0$
  - "benign assumption" does not hold for any meaningful methods

New Evaluation Procedures

▷ **Some prior development on removing "BIG IF":**

- Explicit exploration: force policy to explore every action
  - mix with uniform distribution (ϵ-exploration): $\mathcal{O}(1/\epsilon^6)$ (Khodadadian et al., '21)
  - policy perturbation within evaluation (Li et al., '22): $\mathcal{O}(1/\epsilon^2)$
  - need to modify the policy within evaluation
  - repeatedly taking high-risk actions

New Evaluation Procedures

▷ **Some prior development on removing "BIG IF":**

- Explicit exploration: force policy to explore every action
  - mix with uniform distribution (ϵ-exploration): $\mathcal{O}(1/\epsilon^6)$ (Khodadadian et al., '21)
  - policy perturbation within evaluation (Li et al., '22): $\mathcal{O}(1/\epsilon^2)$
  - need to modify the policy within evaluation
  - repeatedly taking high-risk actions
- No exploration:
  - weighted policy evaluation (Hu et al., '22): $\mathcal{O}(1/\epsilon^{16})$
  - simple, but inefficient

How about best of both worlds?

Preview of our development

Theorem (Li and Lan, '23 – Informal)

An ϵ-optimal policy can be attained by a policy gradient method using $\mathcal{O}(1/\epsilon^2)$ samples, ~~IF~~ ...

Preview of our development

Theorem (Li and Lan, '23 – Informal)

An ϵ-optimal policy can be attained by a policy gradient method using $\mathcal{O}(1/\epsilon^2)$ samples, ~~IF~~ ...

▷ **Some key ingredients**:

1. Policy improvement: stochastic policy mirror descent (Lan, '21)
2. Policy evaluation: new evaluation operator
   - Truncated Monte-Carlo (biased, converge in high probability)
   - No changes to policy (exploits inherent exploration)
3. Analysis:

   Prior development – optimization and evaluation are independent

Preview of our development

Theorem (Li and Lan, '23 – Informal)

An ϵ-optimal policy can be attained by a policy gradient method using $\mathcal{O}(1/\epsilon^2)$ samples, ~~IF~~ ...

▷ **Some key ingredients**:

1. Policy improvement: stochastic policy mirror descent (Lan, '21)
2. Policy evaluation: new evaluation operator
   - Truncated Monte-Carlo (biased, converge in high probability)
   - No changes to policy (exploits inherent exploration)
3. Analysis:

   Our perspective – interaction between optimization and evaluation

SPMD with Truncated On-policy Monte-Carlo

* inherent exploration over action space

Mirror-descent-style Policy Improvement

Algorithm SPMD update: $\pi_k \to \pi_{k+1}$

Input: Estimate $\widehat{Q}_{\mathbb{P}}^{\pi_k}$ from $\mathrm{Eval}(\pi_k)$

Update: For every state $s \in \mathcal{S}$:

$$\pi_{k+1}(\cdot|s) = \mathrm{argmin}_{p \in \Delta_{\mathcal{A}}} \eta_k \langle \widehat{Q}_{\mathbb{P}}^{\pi_k}(s, \cdot), p \rangle + \mathcal{D}_{\pi_k}^{p}(s)$$

- η_k – stepsize
- $\mathcal{D}_{\pi_k}^{p}(s) = w(p) - w(\pi_k(\cdot|s)) - \langle \nabla w(\pi_k(\cdot|s)), p - \pi_k(\cdot|s) \rangle$
  1. $w(\cdot)$: distance generating function (many choices)

Mirror-descent-style Policy Improvement

Algorithm SPMD update: $\pi_k \to \pi_{k+1}$

Input: Estimate $\widehat{Q}_{\mathbb{P}}^{\pi_k}$ from $\mathrm{Eval}(\pi_k)$

Update: For every state $s \in \mathcal{S}$:

$$\pi_{k+1}(\cdot|s) = \mathrm{argmin}_{p \in \Delta_{\mathcal{A}}} \eta_k \langle \widehat{Q}_{\mathbb{P}}^{\pi_k}(s, \cdot), p \rangle + \mathcal{D}_{\pi_k}^{p}(s)$$

- η_k – stepsize
- $\mathcal{D}_{\pi_k}^{p}(s) = w(p) - w(\pi_k(\cdot|s)) - \langle \nabla w(\pi_k(\cdot|s)), p - \pi_k(\cdot|s) \rangle$
  1. $w(\cdot)$: distance generating function (many choices)
  2. projected gradient: $w(p) = \|p\|_2^2$

Mirror-descent-style Policy Improvement

Algorithm SPMD update: $\pi_k \to \pi_{k+1}$

Input: Estimate $\widehat{Q}_{\mathbb{P}}^{\pi_k}$ from $\mathrm{Eval}(\pi_k)$

Update: For every state $s \in \mathcal{S}$:

$$\pi_{k+1}(\cdot|s) = \mathrm{argmin}_{p\in\Delta_{\mathcal{A}}} \eta_k \langle \widehat{Q}_{\mathbb{P}}^{\pi_k}(s,\cdot), p\rangle + \mathcal{D}_{\pi_k}^{p}(s)$$

- η_k – stepsize
- $\mathcal{D}_{\pi_k}^{p}(s) = w(p) - w(\pi_k(\cdot|s)) - \langle \nabla w(\pi_k(\cdot|s)), p - \pi_k(\cdot|s)\rangle$
  1. $w(\cdot)$: distance generating function (many choices)
  2. projected gradient: $w(p) = \|p\|_2^2$
  3. natural policy gradient: $w(p) = \sum_{a\in\mathcal{A}} p_a \log(p_a)$:

$$\pi_{k+1}(a|s) \propto \pi_k(a|s) \exp\left(-\eta_k Q_r^{\pi_k}(s,a)\right)$$

Mirror-descent-style Policy Improvement

Algorithm SPMD update: $\pi_k \to \pi_{k+1}$

Input: Estimate $\widehat{Q}_{\mathbb{P}}^{\pi_k}$ from $\mathrm{Eval}(\pi_k)$

Update: For every state $s \in \mathcal{S}$:

$$\pi_{k+1}(\cdot|s) = \mathrm{argmin}_{p \in \Delta_{\mathcal{A}}} \eta_k \langle \widehat{Q}_{\mathbb{P}}^{\pi_k}(s,\cdot), p \rangle + \mathcal{D}_{\pi_k}^{p}(s)$$

- η_k – stepsize
- $\mathcal{D}_{\pi_k}^{p}(s) = w(p) - w(\pi_k(\cdot|s)) - \langle \nabla w(\pi_k(\cdot|s)), p - \pi_k(\cdot|s) \rangle$
  1. $w(\cdot)$: distance generating function (many choices)
  2. projected gradient: $w(p) = \|p\|_2^2$
  3. natural policy gradient: $w(p) = \sum_{a \in \mathcal{A}} p_a \log(p_a)$:
  $$\pi_{k+1}(a|s) \propto \pi_k(a|s) \exp\left(-\eta_k Q_r^{\pi_k}(s,a)\right)$$
  4. Tsallis divergence with index $q \in (0,1)$: $w(p) = -\sum_{a \in \mathcal{A}} p_a^q$
     - π_{k+1} can be computed via simple bisection

Truncated On-policy Monte-Carlo (TOMC)

Algorithm Truncated Monte-Carlo: $\pi_k \to \widehat{Q}^{\pi_k}$

Generate a trajectory of length n

$$\{(S_0, A_0, C_0), (S_1, A_1, C_1), \ldots, (S_{n-1}, A_{n-1}, C_{n-1})\}$$

for every state-action pair (s, a) **do**

$$t(s,a) = \begin{cases} \text{first timestep hitting } (s,a) \text{ before } n \\ n, \text{otherwise} \end{cases}$$

$$\widehat{Q}^{\pi_k}(s,a) = \sum_{t=t(s,a)}^{n-1} \gamma^{t-t(s,a)} C_t$$

if $\pi_k(a|s) < \tau$:

$$\widehat{Q}^{\pi_k}(s,a) = \frac{1}{1-\gamma}$$ [Truncation step]

end for

1. No changes to the policy, no explicit exploration
2. Forsake learning $Q^{\pi_k}(s,a)$ if $\pi_k(a|s) < \tau$
   - Biased estimate when $\pi_k \approx \pi^*$

SPMD with TOMC

Theorem (Li and Lan, '23 – Informal)

1. *Apply TOMC for evaluation with proper $\tau > 0$*
   - *Choose $n = \mathcal{O}(\log(1/\epsilon)/\tau)$ at each iteration*
2. *Set proper stepsize*

Then SPMD returns an ϵ-optimal policy in $\mathcal{O}(M/\epsilon^2)$ iterations, with high probability.

▷ **Some remarks**:

1. M depend on the divergence $\mathcal{D}^p_{\pi_k}(s)$ in SPMD
   - KL divergence: exponential on $1/(1-\gamma)$ (effective horizon)
   - Tsallis divergence: polynomial

SPMD with TOMC

Theorem (Li and Lan, '23 – Informal)

1. *Apply TOMC for evaluation with proper $\tau > 0$*
   - *Choose $n = \mathcal{O}(\log(1/\epsilon)/\tau)$ at each iteration*
2. *Set proper stepsize*

Then SPMD returns an ϵ-optimal policy in $\mathcal{O}(M/\epsilon^2)$ iterations, with high probability.

▷ **Some remarks**:

1. M depend on the divergence $\mathcal{D}^p_{\pi_k}(s)$ in SPMD
   - KL divergence: exponential on $1/(1-\gamma)$ (effective horizon)
   - Tsallis divergence: polynomial
2. Has certain "memory"

$$\pi_k(a|s) < \tau \;\Rightarrow \pi_{k+1}(a|s) < \pi_k(a|s) < \tau$$

Analysis - High-level Overview

▷ **Suppose the policy optimization method comes with:**

A certificate $\mathbb{C}$

If $\pi_k(a|s) < \tau$, then $a \notin \mathcal{A}^*(s)$ (i.e., a is non-optimal at state s)

▷ **Existence of $\mathbb{C}$ for policy optimization methods:**

1. Policy iteration does not have $\mathbb{C}$ (Li et al., '22)
2. PMD with certain divergences (KL, Tsallis) does!
   - pretty straightforward, but requires exact Q-function
3. Stepsize and noise are both important factors in the existence of $\mathbb{C}$

Analysis - High-level Overview

▷ **What can we do with $\mathbb{C}$?**

1. If $\pi_k(a|s) \geq \tau$, then a is explored by π_k, and

   $Q^{\pi_k}(s, a)$ can be learned well

   This is a good thing.

Analysis - High-level Overview

▷ **What can we do with $\mathbb{C}$?**

1. If $\pi_k(a|s) \geq \tau$, then a is explored by π_k, and

$$Q^{\pi_k}(s,a) \text{ can be learned well}$$

   This is a good thing.

2. If $\pi_k(a|s) < \tau$, then SPMD + TOMC makes sure

$$\pi_{k+1}(a|s) < \pi_k(a|s)$$

   Another good thing! As $\mathbb{C}$ already guarantees $a \notin \mathcal{A}^*(s)$

"TOMC learns every action that still matters"

Does $\mathbb{C}$ Exist in the Stochastic Setting?

▷ **Requirement: $\mathbb{C}$ should hold in high probability**

Approach I: Multiple Trajectory

Suppose $\mathbb{C}$ holds at iter k with prob p, use proper stepsize and trajectory configuration in TOMC so $\mathbb{C}$ holds at iter $k+1$ with prob $p' \approx p$

- Requires large number of trajectories ($\mathcal{O}(1/\epsilon^2)$), each of length $\mathcal{O}(\log(1/\epsilon)/\tau)$, for every SPMD step
- $\#$ SPMD steps: $\mathcal{O}(1/\epsilon^2)$
- Total sample complexity: $\widetilde{\mathcal{O}}(1/(\tau\epsilon^4))$

Does $\mathbb{C}$ Exist in the Stochastic Setting?

▷ **Requirement: $\mathbb{C}$ should hold in high probability**

Approach I: Multiple Trajectory

Suppose $\mathbb{C}$ holds at iter k with prob p, use proper stepsize and trajectory configuration in TOMC so $\mathbb{C}$ holds at iter $k+1$ with prob $p' \approx p$

- Requires large number of trajectories ($\mathcal{O}(1/\epsilon^2)$), each of length $\mathcal{O}(\log(1/\epsilon)/\tau)$, for every SPMD step
- $\#$ SPMD steps: $\mathcal{O}(1/\epsilon^2)$
- Total sample complexity: $\widetilde{\mathcal{O}}(1/(\tau\epsilon^4))$

1. Interaction between evaluation and optimization
   - ⋆ it suffices to bound the accumulated noise across iterations
   - can bound each noise term in high probability

Does $\mathbb{C}$ Exist in the Stochastic Setting?

▷ **Requirement: $\mathbb{C}$ should hold in high probability**

Approach I: Multiple Trajectory

Suppose $\mathbb{C}$ holds at iter k with prob p, use proper stepsize and trajectory configuration in TOMC so $\mathbb{C}$ holds at iter $k+1$ with prob $p' \approx p$

- Requires large number of trajectories ($\mathcal{O}(1/\epsilon^2)$), each of length $\mathcal{O}(\log(1/\epsilon)/\tau)$, for every SPMD step
- # SPMD steps: $\mathcal{O}(1/\epsilon^2)$
- Total sample complexity: $\widetilde{\mathcal{O}}(1/(\tau\epsilon^4))$

1. Interaction between evaluation and optimization
   - ⋆ it suffices to bound the accumulated noise across iterations
   - can bound each noise term in high probability
2. Selection of τ depends on the Bregman divergence
   - KL divergence: $\tau \asymp |\mathcal{A}|^{-1/(1-\gamma)}$
   - Tsallis divergence with index $q = 1/2$ (non-optimal selection): $\tau \asymp (1-\gamma)^4 \, |\mathcal{A}|^{-1}$

Does $\mathbb{C}$ Exist in the Stochastic Setting?

▷ **Requirement: $\mathbb{C}$ should hold in high probability**

Approach II: Single Trajectory

- A single trajectory of length $\mathcal{O}(\log(1/\epsilon)/\tau)$, for every SPMD step
- # SPMD steps: $\mathcal{O}(1/\epsilon^2)$
- Total sample complexity: $\widetilde{\mathcal{O}}(1/(\tau\epsilon^2))$

1. Interaction between evaluation and optimization
   - ⋆ it suffices to bound the accumulated noise across iterations

Does $\mathbb{C}$ Exist in the Stochastic Setting?

▷ **Requirement: $\mathbb{C}$ should hold in high probability**

Approach II: Single Trajectory

- A single trajectory of length $\mathcal{O}(\log(1/\epsilon)/\tau)$, for every SPMD step
- # SPMD steps: $\mathcal{O}(1/\epsilon^2)$
- Total sample complexity: $\widetilde{\mathcal{O}}(1/(\tau\epsilon^2))$

1. Interaction between evaluation and optimization
   - ⋆ it suffices to bound the accumulated noise across iterations
2. A more refined probabilistic argument to bound the accumulated noise up to iter t as $\mathcal{O}(\sqrt{t})$
   - ⋆ noise in $Q^{\pi_k,\xi_k}(s,a)$ is policy-dependent (grows when $\pi_k(a|s)$ decays)
   - ⋆ truncation in TOMC is essential

Summary

1. Evaluation seems at odds with optimization
2. SPMD + TOMC with proper divergence exhibits inherent exploration
   - Optimal actions maintain divergence-dependent prob lower bound
   - Non-optimal actions get appropriately ignored
3. More details in the paper
   - ⋆ An alternative evaluation procedure (unbiased estimate of Q-function regardless of the policy)

Presentation based on Preprint

- Li, Y., & Lan, G. (2023). Policy Mirror Descent Inherently Explores Action Space. arXiv preprint arXiv:2303.04386, under revision at SIOPT