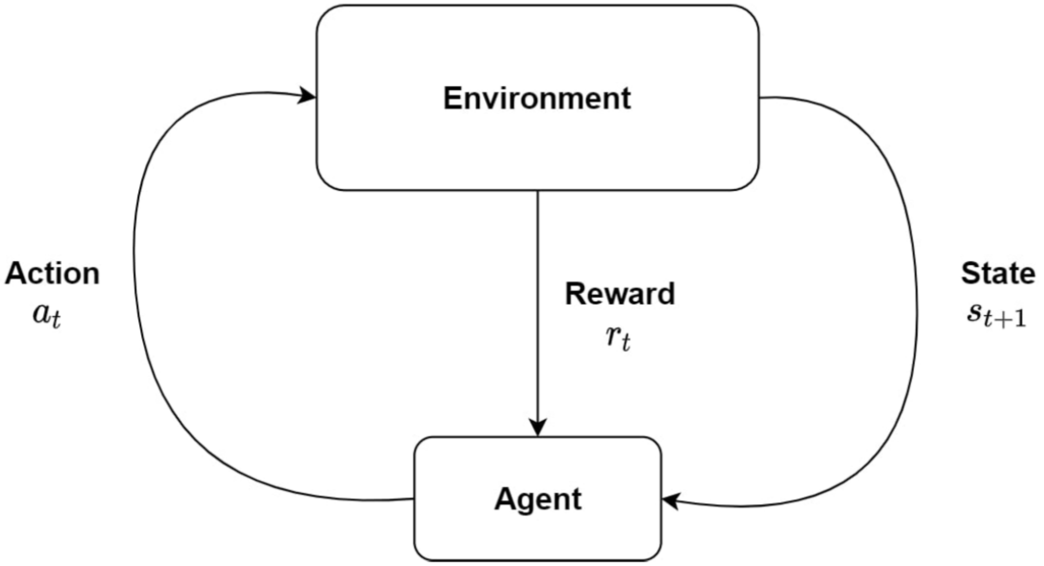
Exploration Bonus for Policy Optimization

with Haipeng Luo and Chung-Wei Lee in NeurIPS 2021

Chen-Yu Wei (MIT→UVA)

Jan. 19, 2023

(Online) Reinforcement Learning



(Fares & Younes, 2020)

Standard Methods

Method	Parameterizing …?	How to derive output policy?
Model-based	$r(s,a),\ \ p(s'\|s,a)$	Planning
Q-learning	$Q^\star(s,a)$	$\pi^\star(s) = \mathrm{argmax}_a\, Q^\star(s,a)$
Policy gradient	$\pi^\star(a\|s)$	--

Each has their strength and weakness. The choice is application dependent.

Policy Gradient

π_θ: policy parameterized by θ

$V(\pi)$: expected (long-term) reward under policy π

Policy gradient:

collect data using π_θ

$\theta \leftarrow \theta + \eta \nabla_\theta V(\pi_\theta)$

repeat

can only be "estimated"

Strength of Policy Gradient

- Folklore: more robust against modeling error

- Theoretical justification
  - More robust against **model mis-specification** or **data corruption**
    [Agarwal et al., 2020] PC-PG: Policy cover directed exploration for provable policy gradient learning.
    [Zhang et al., 2021] Robust policy gradient against strong data corruption.

  - PG handles the case where the **reward is adversarial**: consider the episodic setting, where the reward function is different in every episode.

  PG ≈ mirror descent in the online learning literature

  [Even-Dar et al. 2009] Online Markov decision processes.

Weakness of Policy Gradient

- Folklore: less sample efficient, only perform local policy search

- Theoretical understanding:
  - The sample complexity for PG involves **distribution mismatch factor** $C = \max_s \frac{\mu^{\pi^\star}(s)}{\mu^{\pi_{\text{learner}}}(s)}$

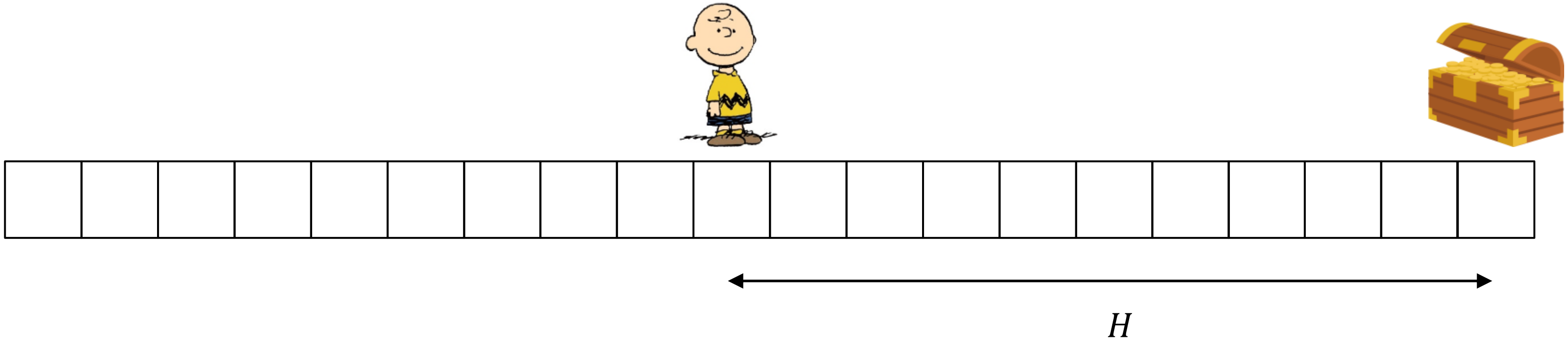
    [Agarwal et al., 2020] On the theory of policy gradient methods: optimality, approximation, and distribution shift

  - The issue is not specific for PG. But for model-based method or Q-learning, there were solutions:

    [Jaksch et al. 2010] Near-optimal regret bounds for reinforcement learning. **(UCRL)**
    [Jin et al., 2018] Is Q-learning provably efficient? NeurIPS 2018. **(UCB-Q)**

- Theoretically less unclear: can we / how to perform **global policy search** with PG?

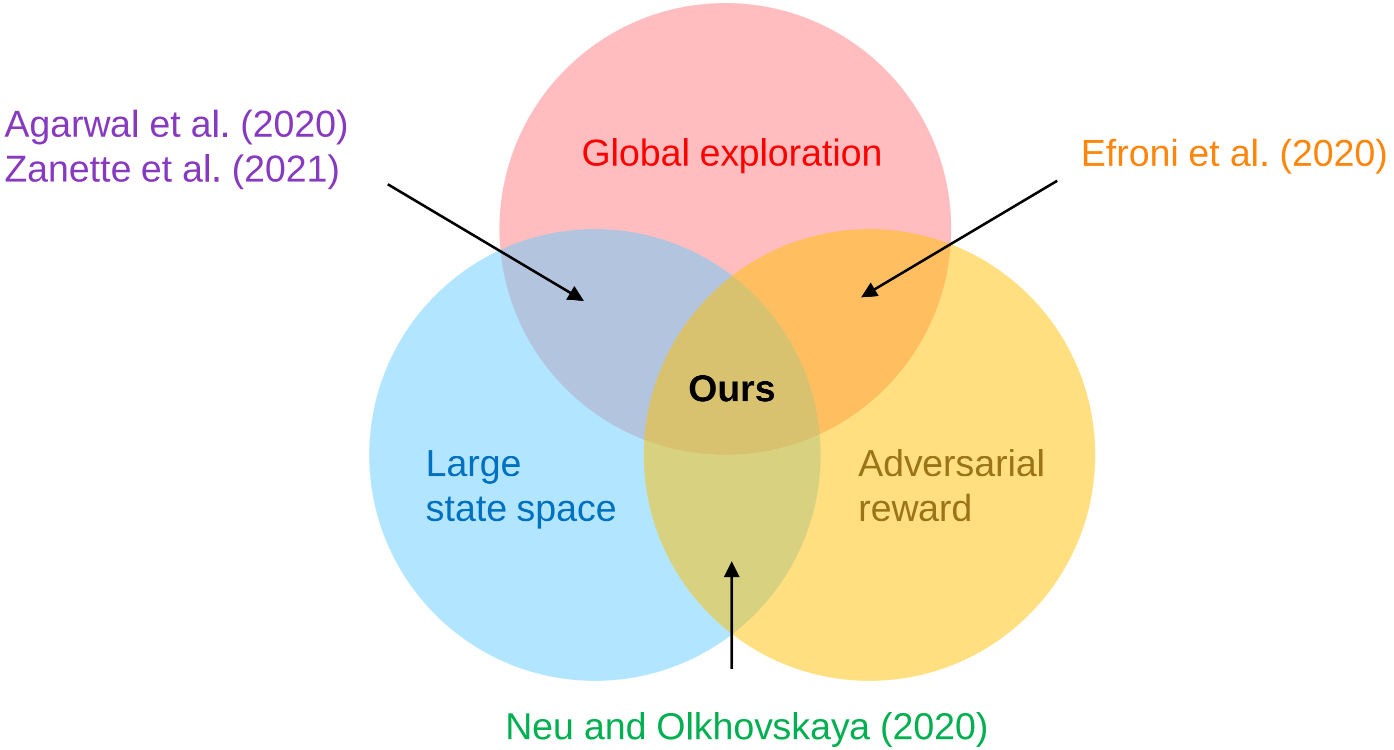
Motivating Example



Initial policy: $\frac{1}{2}$ go left, $\frac{1}{2}$ go right

⇒ Sample complexity under standard policy gradient $\geq 2^H$

We are going to address this issue in this talk.



[Agarwal et al., 2020] PC-PG: Policy cover directed exploration for provable policy gradient learning.
[Zanette et al., 2021] Cautiously optimistic policy optimization and exploration with linear function approximation.
[Efroni et al., 2020] Optimistic policy optimization with bandit feedback.
[Neu and Olkhovskaya, 2021] Online learning in MDPs with linear function approximation and bandit feedback.

Our solution is comparatively more elegant and the theory is easier to understand.

Outline

- Preliminaries on Multi-Armed Bandits
- RL Setting
- Algorithm
- Results for finite MDP
- Results for MDP with linear structure

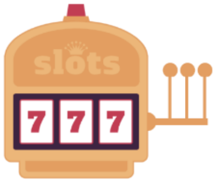
The Multi-Armed Bandit (MAB) Problem

For $t = 1, \ldots, T$:

(Environment decides $R_t(a) \in [-C, C]$ arbitrarily for all a)

Choose an arm/action $a_t \in \{1, 2, \ldots, A\}$.

Receive $R_t(a_t)$.



$$\text{Regret } = \max_{a^\star} \sum_{t=1}^{T} R_t(a^\star) - \sum_{t=1}^{T} R_t(a_t)$$

Exponential Weight Algorithm for MAB

Exponential Weight Algorithm [Auer et al. 2002] The non-stochastic multi-armed bandit problem

$p_1(a) = 1/A$

Repeat:

Sample $a_t \sim p_t(\cdot)$

Update $p_{t+1}(a) \propto p_t(a)\ e^{\eta\ \hat{R}_t(a)}$

η: learning rate

$\hat{R}_t(a) = \frac{\mathbb{I}\{a_t=a\}}{p_t(a)} R_t(a)$ (unbiased reward estimator)

$$\text{Regret} \lesssim \underbrace{\frac{1}{\eta}}_{\text{bias}} + \underbrace{\eta \sum_{t=1}^{T} \sum_{a} p_t(a) \hat{R}_t(a)^2}_{\text{variance}} \leq \frac{1}{\eta} + \eta C^2 AT \lesssim C\sqrt{AT}$$

choose optimal $\eta \approx \frac{1}{C\sqrt{AT}}$

Outline

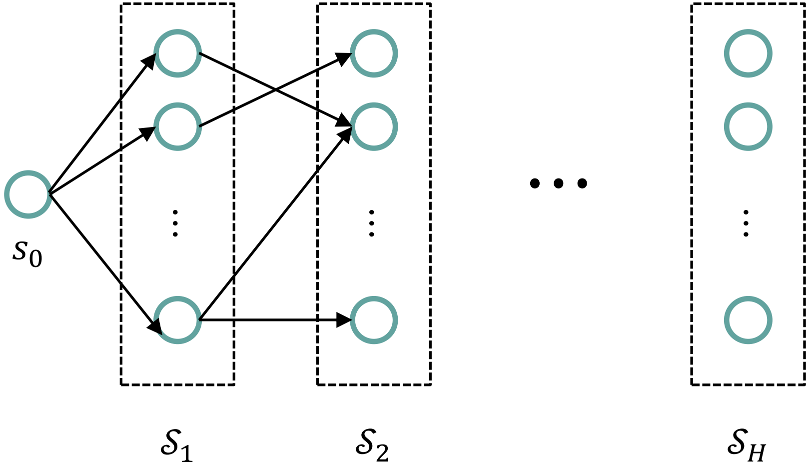
- Preliminaries on Multi-Armed Bandits
- RL Setting
- Algorithm
- Results for finite MDP
- Results for MDP with linear structure

MDP Setting

Horizon length: H
Set of states: $\mathcal{S} = \{s_0\} \cup \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_H$
Set of actions: $\mathcal{A}$

Policy $\pi(\cdot|s)$: distribution over actions

Episode: walk from s_0 to $\mathcal{S}_H$ once



Interaction Protocol and Regret

For episode $t = 1, \ldots, T$:

Choose a policy π_t

Interact with the MDP for one episode using π_t, and generate

$\left(s_0,\ a_{t0},\ r_t(s_{t0}, a_{t0}),\ s_1,\ a_{t1},\ r_t(s_{t1}, a_{t1}),\ \ldots\ldots, s_{tH},\ \ a_{tH},\ r_t(s_{tH}, a_{tH})\right)$

where $r_t(s, a)$ is the reward function in episode t (can vary across episodes)

$$\text{Regret } = \max_{\pi^\star} \sum_{t=1}^{T} V^{\pi^\star}(s_0;\ r_t) - \sum_{t=1}^{T} V^{\pi_t}(s_0;\ r_t)$$

$$V^{\pi}(s; r) \triangleq \mathbb{E}\left[\sum_{k=h}^{H} r(s_{k,} a_k)\ \middle|\ \text{executing } \pi \text{ from } s \in \mathcal{S}_h\right]$$

Regret Decomposition

Performance difference lemma: For any policies $\pi^\star$ and π, and any reward function r,

$$V^{\pi^\star}(s_0; r) - V^{\pi}(s_0; r) = \sum_{s \in \mathcal{S}} \mu^{\pi^\star}(s) \sum_{a \in \mathcal{A}} \left(\pi^\star(a|s) - \pi(a|s)\right) Q^{\pi}(s, a; r)$$

$\mu^{\pi}(s)$: expected number of times of visiting s (in an episode) under π

$Q^{\pi}(s, a; r) = \mathbb{E}\left[\sum_{k=h}^{H} r(s_k, a_k) \;\middle|\; \text{executing } \pi \text{ from } s \in \mathcal{S}_h \text{ and take } a_h = a\right]$

$$\text{Regret} = \sum_{t=1}^{T} \left(V^{\pi^\star}(s_0; r_t) - V^{\pi_t}(s_0; r_t)\right) = \sum_{s \in \mathcal{S}} \mu^{\pi^\star}(s) \sum_{t=1}^{T} \sum_{a \in \mathcal{A}} \left(\pi^\star(a|s) - \pi_t(a|s)\right) Q^{\pi_t}(s, a; r_t)$$

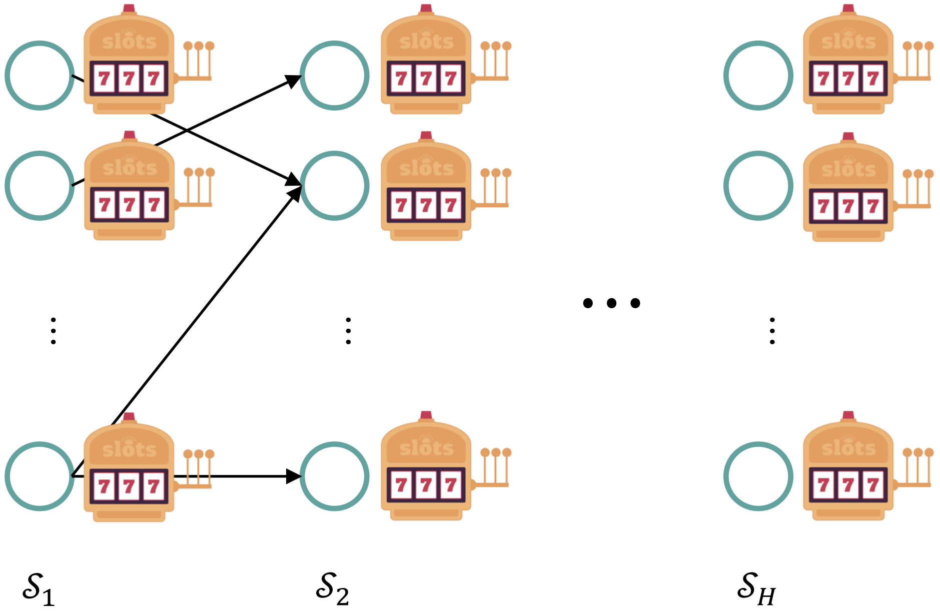
The regret of a MAB problem on state s with reward of arm a being $Q^{\pi_t}(s, a; r_t)$

If we can do well on the bandit problem on every state, we can also do well on the MDP.

Outline

- Preliminaries on Multi-Armed Bandits
- RL Setting
- Algorithm
- Results for finite MDP
- Results for MDP with linear structure

A "Natural" Algorithm Inspired by PDL



Run a MAB algorithm on every state!

For the MAB algorithm on state s
Reward of arm a in round t = $Q^{\pi_t}(s, a; r_t)$

Running exponential weight on every state is equivalent to *Natural Policy Gradient* and closely related to *TRPO* & *PPO*.

[Neu et al., 2017] A unified view of entropy-regularized Markov decision processes.
[Agarwal et al., 2020] On the theory of policy gradient methods: optimality, approximation, and distribution shift

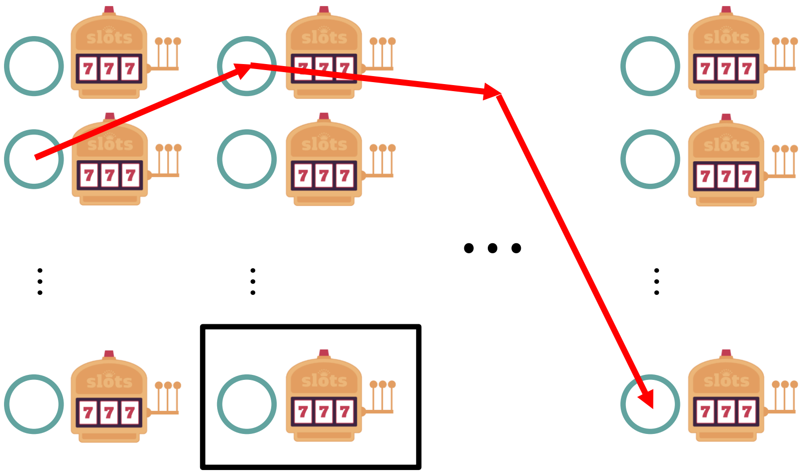
Regret Analysis

$$\text{Regret}^{\text{MDP}} = \sum_{s \in \mathcal{S}} \mu^{\pi^\star}(s)\ \text{Regret}^{(s)} \quad \text{(By PDL)}$$

$$\leq \sum_{s \in \mathcal{S}} \mu^{\pi^\star}(s)\ H\sqrt{AT} \quad \text{(Regret bound of exponential weight)}$$

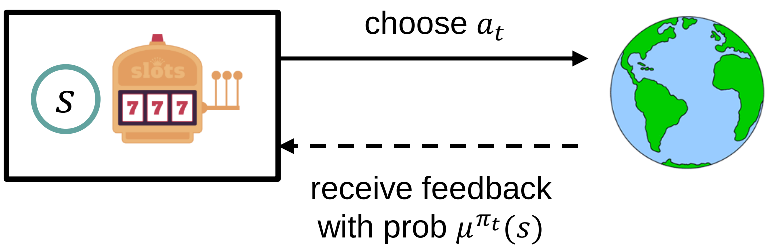
$$\leq H^2\sqrt{AT} \quad \textbf{(?!)}$$

The Issue



the sum of reward starting from (s, a)

Issue: the value of $\{Q^{\pi_t}(s, a; r_t)\}_a$ can be estimated only when the learner visit s.



Corrected Regret Analysis

$$\hat{R}_t(a)$$

reward estimator for **standalone** MAB

→

$$\frac{\mathbb{I}\{s \text{ is visited}\}}{\mu^{\pi_t}(s)} \hat{R}_t(a)$$

reward estimator for MAB **on state** $\boldsymbol{s}$

$$\text{Regret}^{(s)} \le \frac{1}{\eta} + \eta \sum_{t=1}^{T} \sum_{a} p_t(a) \frac{\mathbb{I}\{s \text{ is visited}\}}{\mu^{\pi_t}(s)^2} \hat{R}_t(a)^2 \lesssim \frac{1}{\eta} + \eta H^2 \sum_{t=1}^{T} \frac{A}{\mu^{\pi_t}(s)}$$

Variance term increased!

$$\text{Regret}^{(\text{MDP})} = \sum_{s \in \mathcal{S}} \mu^{\pi^\star}(s)\, \text{Regret}^{(s)}$$

$$\le \sum_{s \in \mathcal{S}} \mu^{\pi^\star}(s) \left(\frac{1}{\eta} + \eta H^2 \sum_{t=1}^{T} \frac{A}{\mu^{\pi_t}(s)} \right) \le \sqrt{H^3 A \sum_{t=1}^{T} \sum_{s} \frac{\mu^{\pi^\star}(s)}{\mu^{\pi_t}(s)}}$$

distribution mismatch factor

Outline

- Preliminaries on Multi-Armed Bandits
- RL Setting
- Algorithm
- Solution for finite MDP
- Solution for MDP with linear structure

Removing the Distribution Mismatch (our contribution)

Define $$b_t(s) = \frac{\eta H^2 A}{\mu^{\pi_t}(s)}$$ $$\left(\text{in the paper, } \; b_t(s) = \frac{\eta H^2 A}{\mu^{\pi_t}(s) + \gamma} \leq 1\right)$$

Instead of running the NPG on the original reward $r_t(s,a)$,
run it on $r_t(s,a) + b_t(s)$.

$\widetilde{MDP}$ is the MDP with reward $r_t + b_t$

$$\text{Regret}^{(\widetilde{MDP})} \le \sum_{s\in\mathcal{S}} \mu^{\pi^\star}(s)\left(\frac{1}{\eta} + \eta H^2 \sum_{t=1}^{T} \frac{A}{\mu^{\pi_t}(s)}\right)$$

$$\sum_{t=1}^{T}\left[V^{\pi^\star}(s_0; r_t + b_t) - V^{\pi_t}(s_0; r_t + b_t)\right] \le \frac{H}{\eta} + \sum_{t=1}^{T}\sum_{s\in\mathcal{S}} \mu^{\pi^\star}(s)\left(\frac{\eta H^2 A}{\mu^{\pi_t}(s)}\right)$$

$$\sum_{t=1}^{T}\Big[\underbrace{V^{\pi^\star}(s_0; r_t) - V^{\pi_t}(s_0; r_t)}_{\text{Regret}^{(MDP)}} + V^{\pi^\star}(s_0; b_t) - V^{\pi_t}(s_0; b_t)\Big] \le \frac{H}{\eta} + \sum_{t=1}^{T}\sum_{s\in\mathcal{S}} \mu^{\pi^\star}(s)\ b_t(s) = \frac{H}{\eta} + \sum_{t=1}^{T} V^{\pi^\star}(s_0; b_t)$$

$$\text{Regret}^{(MDP)} \le \frac{H}{\eta} + \sum_{t=1}^{T} V^{\pi_t}(s_0; b_t)$$

$$\le \frac{H}{\eta} + \sum_{t=1}^{T}\sum_{s\in\mathcal{S}} \mu^{\pi_t}(s)\left(\frac{\eta H^2 A}{\mu^{\pi_t}(s)}\right)$$

$$\le \frac{H}{\eta} + \eta H^2 SAT = \boxed{\sqrt{H^3 SAT}}$$

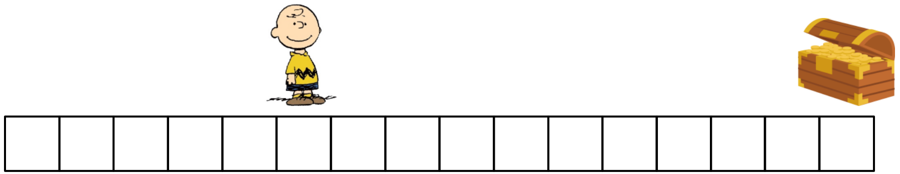
c.f. without bonus: $\sqrt{H^3 A \sum_{t=1}^{T}\sum_{s} \frac{\mu^{\pi^\star}(s)}{\mu^{\pi_t}(s)}}$

Algorithm Overview

- **Eliminating distribution mismatch:** Run over reward function

$$r_t(s,a) + b_t(s) = r_t(s,a) + \frac{\eta H^2 A}{\mu^{\pi_t}(s) + \gamma}$$

- **Extra effort:** need to estimate $\frac{1}{\mu^{\pi_t}(s)}$ for all states
  - Sampling
  - Learn transitions directly
  - Use another model to fit $\frac{1}{\mu^{\pi_t}(s)}$



Outline

- Preliminaries on Multi-Armed Bandits
- RL Setting
- Algorithm
- Solution for finite MDP
- Solution for MDP with linear structure

Generalization to Linear Function Approximation

- **Assumption:** there exists some known $\phi(s,a) \in \mathbb{R}^d$ such that for any π, $Q^\pi(s,a;r_t) = \phi(s,a)^\top \theta_t^\pi$ for some $\theta_t^\pi \in \mathbb{R}^d$.

- Similarly, run policy gradient over $r_t(s,a) + b_t(s,a)$, with

$$b_t(s,a) = \eta \ \phi(s,a)^\top \Sigma_t^{-1} \phi(s,a)$$

where $\Sigma_t = \mathbb{E}_{(s,a)\sim\pi_t}[\phi(s,a)\phi(s,a)^\top]$

- Do we need to run a bandit algorithm on every state? (#state could be ∞)
  - No. It's still equivalent to NPG, which is implementable.
  - In the mathematical analysis, it's equivalent to run a *linear bandit* algorithm on *every* state.

Summary

The steps to derive the form of the bonus:

1. Use the performance difference lemma:
   PG $\approx$ running individual bandit on every state, with feedback observed with prob $\mu^{\pi_t}(s)$

2. Write out the regret of individual MAB under importance weight

3. Set $b_t(s,a)$ based on the regret bound in Step 2

Remarks

- **Potential Issue 1:** The bonus will introduce very **dense, time-varying** reward to guide policy search. This is reasonable if the goal is to find **globally optimal** policy. In some applications, this might not be necessary / too costly.

- **Potential Issue 2:** Calculating the bonus requires extra sampling.

- **Empirical study:** How to adapt this idea in practice remains open.

Thanks!

More questions?
Contact me via **chenyu.wei@usc.edu**