Uniform Last-Iterate Guarantee for Bandits and Reinforcement Learning

Junyan Liu [1] Yunfan Li [2] Ruosong Wang [3] Lin F. Yang [2]

[1] University of Washington

[2] University of California, Los Angeles

[3] Peking University

What We Hope in Online Learning?

- In online learning framework, a learner is given a policy set and sequentially interacts with environment. At each round, the learner
  - Play a policy from the policy set.
  - Observe the reward(s).

What We Hope in Online Learning?

- In online learning framework, a learner is given a policy set and sequentially interacts with environment. At each round, the learner
  - Play a policy from the policy set.
  - Observe the reward(s).
- This framework can be instantiated by bandit problems and online reinforcement learning.

What We Hope in Online Learning?

- In online learning framework, a learner is given a policy set and sequentially interacts with environment. At each round, the learner
  - Play a policy from the policy set.
  - Observe the reward(s).
- This framework can be instantiated by bandit problems and online reinforcement learning.
- **Common Goal.** Obtain a good cumulative performance typically measured by regret or PAC bounds.

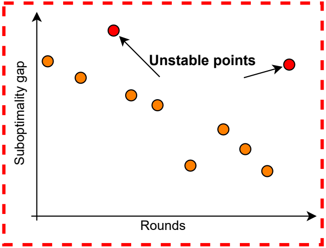
What We Hope in Online Learning?

- In online learning framework, a learner is given a policy set and sequentially interacts with environment. At each round, the learner
  - Play a policy from the policy set.
  - Observe the reward(s).
- This framework can be instantiated by bandit problems and online reinforcement learning.
- **Common Goal.** Obtain a good cumulative performance typically measured by regret or PAC bounds.

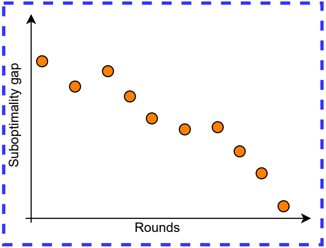
How about instantaneous performance?

What We Hope in Online Learning?

- **Issue:** for high-stakes applications such as medical trials, a good cumulative performance is not enough. Every policy matters!



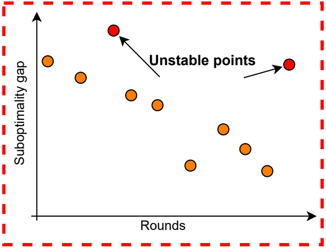
Good cumulative performance, but some very bad policies



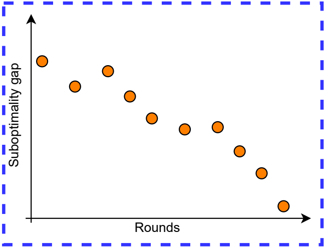
What we hope: good cumulative performance & every policy is not too bad

What We Hope in Online Learning?

- **Issue:** for high-stakes applications such as medical trials, a good cumulative performance is not enough. Every policy matters!



Good cumulative performance, but some very bad policies



What we hope: good cumulative performance & every policy is not too bad

- A natural question arises:

> Is there a metric characterizing both cumulative and instantaneous performance?

Contributions: a new metric

Definition: uniform last-iterate (ULI)

Let Δ_t be the suboptimality gap at round t. An algorithm is ULI, if for a given $\delta \in (0, 1)$,

$$\mathbb{P}\left(\forall t \in \mathbb{N} : \Delta_t \leq F(\delta, t)\right) \geq 1 - \delta,$$

where $F(\delta, t)$ is polynomial in $\log(1/\delta)$ and proportional to the product of power functions of $\log t$ and $1/t$ (e.g., $F(\delta, t) \approx \sqrt{\frac{\log t}{t}}$).

- By definition, ULI itself characterizes the instantaneous performance.

Contributions: a new metric

Definition: uniform last-iterate (ULI)

Let Δ_t be the suboptimality gap at round t. An algorithm is ULI, if for a given $\delta \in (0, 1)$,

$$\mathbb{P}\left(\forall t \in \mathbb{N} : \Delta_t \leq F(\delta, t)\right) \geq 1 - \delta,$$

where $F(\delta, t)$ is polynomial in $\log(1/\delta)$ and proportional to the product of power functions of $\log t$ and $1/t$ (e.g., $F(\delta, t) \approx \sqrt{\frac{\log t}{t}}$).

- By definition, ULI itself characterizes the instantaneous performance.
- ULI implies uniform-PAC.
  - Not only cumulative but also instantaneous performance.

Contributions: a new metric

Definition: uniform last-iterate (ULI)

Let Δ_t be the suboptimality gap at round t. An algorithm is ULI, if for a given $\delta \in (0,1)$,

$$\mathbb{P}\left(\forall t \in \mathbb{N} : \Delta_t \leq F(\delta, t)\right) \geq 1 - \delta,$$

where $F(\delta, t)$ is polynomial in $\log(1/\delta)$ and proportional to the product of power functions of $\log t$ and $1/t$ (e.g., $F(\delta, t) \approx \sqrt{\frac{\log t}{t}}$).

- By definition, ULI itself characterizes the instantaneous performance.
- ULI implies uniform-PAC.
  - Not only cumulative but also instantaneous performance.

Is ULI optimally achievable by bandit and RL algorithms?

Achievability in finite-armed bandit problems

For finite arm setting, we show (K is # of arms; Δ is minimum gap):

Achievability in finite-armed bandit problems

For finite arm setting, we show (K is # of arms; Δ is minimum gap):

- Phased elimination (PE) holds ULI with

$$F(t,\delta) \lesssim t^{-\frac{1}{2}} \sqrt{K \log\left(\delta^{-1} K \log(t)\right)}.$$

Achievability in finite-armed bandit problems

For finite arm setting, we show (K is # of arms; Δ is minimum gap):

- Phased elimination (PE) holds ULI with

$$F(t,\delta) \lesssim t^{-\frac{1}{2}}\sqrt{K\log\left(\delta^{-1}K\log(t)\right)}.$$

- An algorithmic lower bound for lil'UCB [Jamieson et al., 2014];

$$\exists t = \Omega\left(\Delta^{-2}\right) \text{ such that } F(t,\delta) \gtrsim t^{-\frac{1}{4}}\sqrt{\log\left(\delta^{-1}\log\left(\Delta^{-1}\right)\right)}.$$

- lil'UCB is uniform-PAC since bonus function is as $\sqrt{\log\log n/n}$ rather than $\sqrt{\log\log t/n}$.
- Near-opt ULI implies near-opt uniform-PAC, but not the other way around, i.e., ULI is strictly stronger than uniform-PAC.

Achievability in infinite-armed linear bandit problems

- For compact large arm space (possibly infinite), we propose an oracle-efficient linear bandit algorithm that holds the ULI.

Achievability in infinite-armed linear bandit problems

- For compact large arm space (possibly infinite), we propose an oracle-efficient linear bandit algorithm that holds the ULI.
- Starting point: phased elimination [Chapter 22, Bandit Algorithms].

Achievability in infinite-armed linear bandit problems

- For compact large arm space (possibly infinite), we propose an oracle-efficient linear bandit algorithm that holds the ULI.
- Starting point: phased elimination [Chapter 22, Bandit Algorithms].
- Computational issue of phased elimination (PE):
  - complexity of G-optimal design scales linearly with arm set.

Achievability in infinite-armed linear bandit problems

- For compact large arm space (possibly infinite), we propose an oracle-efficient linear bandit algorithm that holds the ULI.
- Starting point: phased elimination [Chapter 22, Bandit Algorithms].
- Computational issue of phased elimination (PE):
  - complexity of G-optimal design scales linearly with arm set.
- Key idea: select finite arms to represent all well-behaved active arms. Then, do G-optimal design on finite arms.

Achievability in infinite-armed linear bandit problems

- For compact large arm space (possibly infinite), we propose an oracle-efficient linear bandit algorithm that holds the ULI.
- Starting point: phased elimination [Chapter 22, Bandit Algorithms].
- Computational issue of phased elimination (PE):
  - complexity of G-optimal design scales linearly with arm set.
- Key idea: select finite arms to represent all well-behaved active arms. Then, do G-optimal design on finite arms.
- Key technique: Adaptive barycentric spanner, generalize that of [Awerbuch & Kleinberg, 2008];
  - Adaptively find proper linear subspace in which active arms span.
  - call a linearly-constrained optimization oracle $\texttt{poly}(d)$ times.
- ULI guarantee: $F(\delta, t) \lesssim t^{-\frac{1}{2}} \sqrt{d^3 \log(dt)}$.

Achievability in online RL

- For tabular episodic MDPs, we propose a model-based alg. with ULI guarantee:

$$F(\delta, t) \lesssim t^{-\frac{1}{2}} \log(\delta^{-1} t) \cdot \texttt{poly}(H, S, A),$$

where H is the horizon, S : # of states, and A : # of actions.

Achievability in online RL

- For tabular episodic MDPs, we propose a model-based alg. with ULI guarantee:

$$F(\delta, t) \lesssim t^{-\frac{1}{2}} \log(\delta^{-1} t) \cdot \texttt{poly}(H, S, A),$$

where H is the horizon, S : # of states, and A : # of actions.

- High-level: exhaustively learn the transition model; then conduct policy elimination over all *deterministic* policies.
- Starting point: UCB-VI [Atar et al.,'17] and our adjustment:
  - Use uncertainty-driven reward functions $r(s, a) \approx \frac{1}{\sqrt{n(s,a)}}$;
  - Play policies that maximize the uncertainty to aggressively explore the transition model;
  - Conduct policy elimination when model is well-approximated.