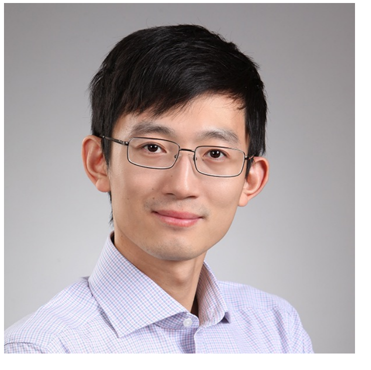
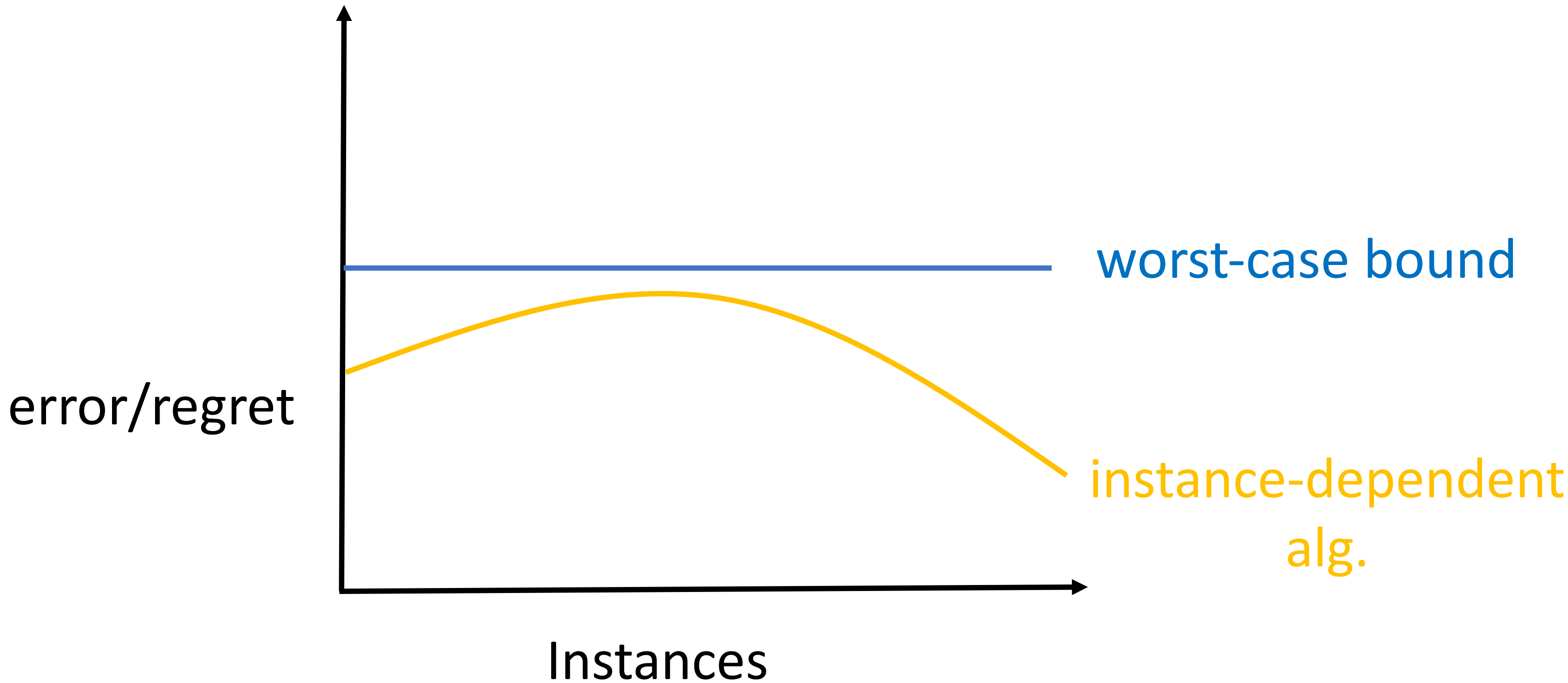
Asymptotic Instance-Optimal Algorithms for Interactive Decision Making

Kefan Dong, Tengyu Ma
Stanford University



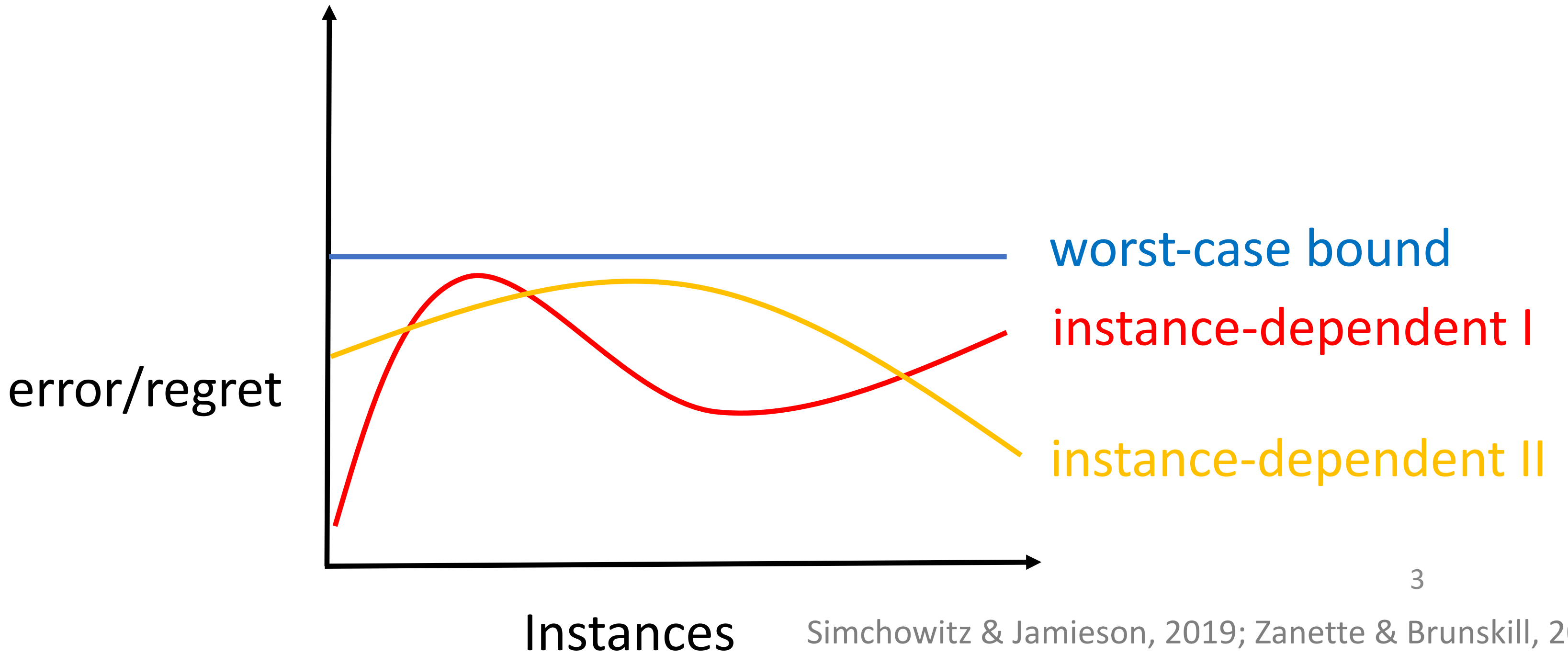
Instance-dependent algorithms and analysis

- Many classical ML algorithms & analyses focus on worst-case instances
- An ideal algorithm should perform better on "easy" instances
- The error/regret guarantee should depend on the instance



Instance-dependent algorithms and analysis (Cont'd)

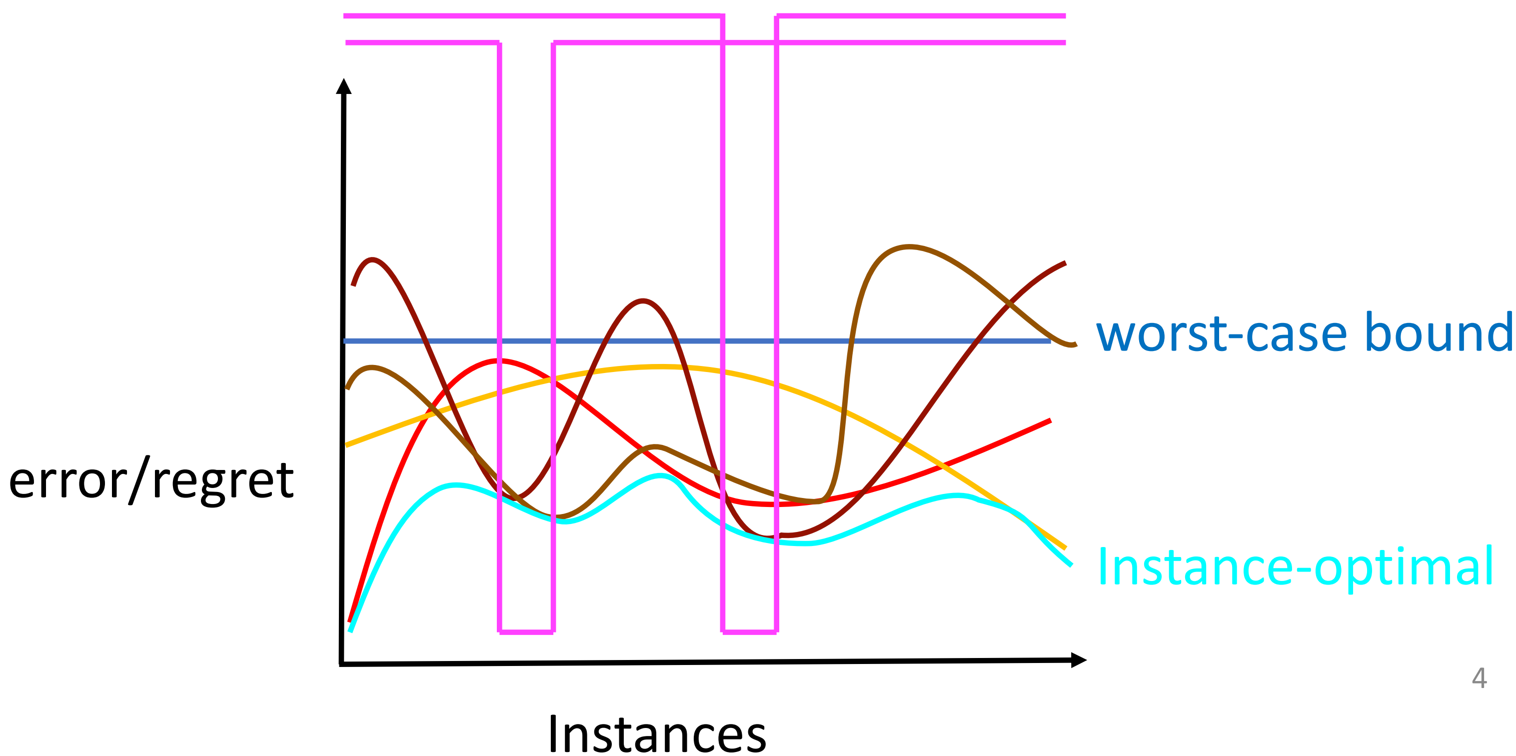
- Notable examples:
  - Bounds for multi-armed bandit / RL that depends on the gap condition or the value of optimal policy
  - Margin-based bounds for classification problems
- Challenge: how do we compare these bounds?



Simchowitz & Jamieson, 2019; Zanette & Brunskill, 2019

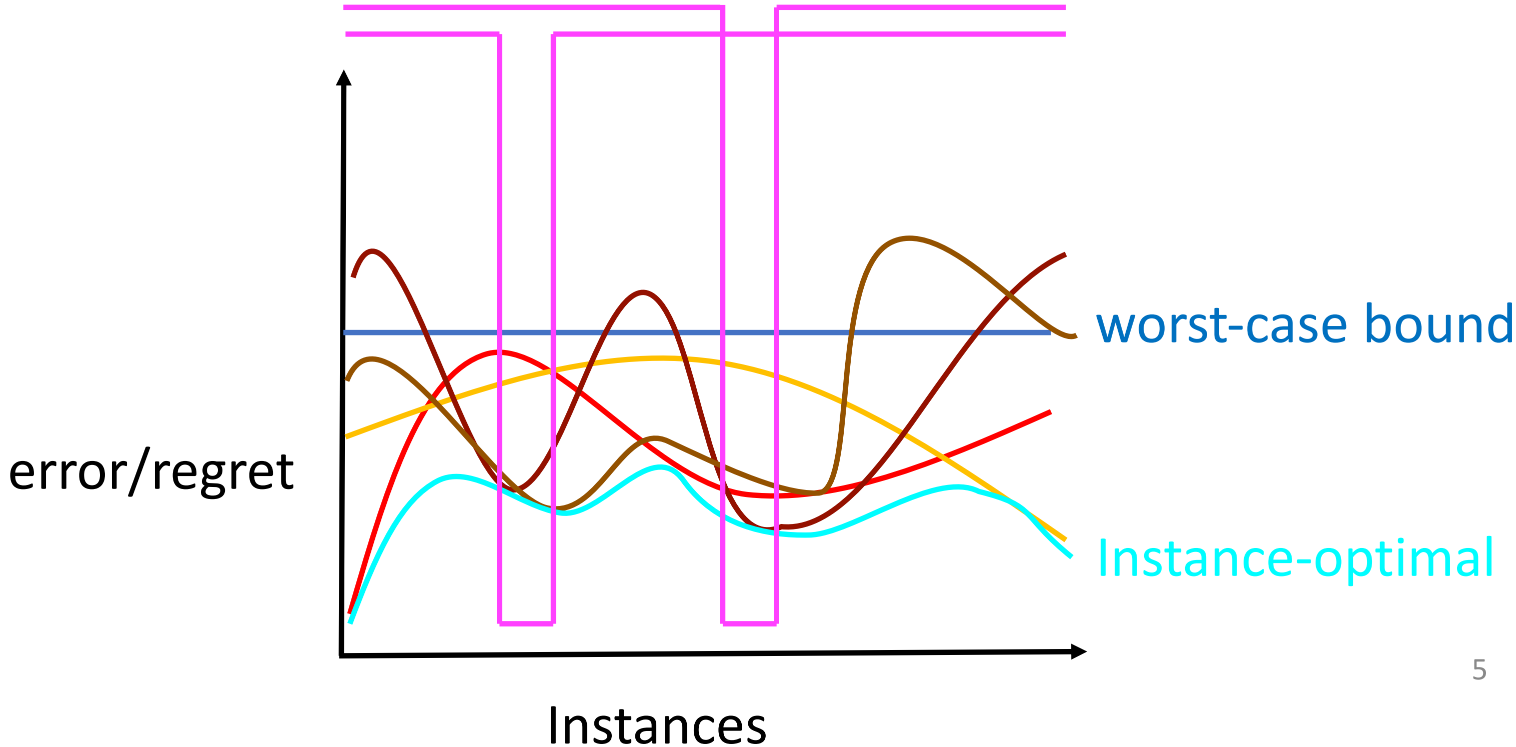
Instance-optimal algorithms

- Can one alg. outperform all other algorithms on every instance?
- Issue: an alg. can perfectly memorize one special instance and fail on all other instances
- Impossible to beat every other algorithm



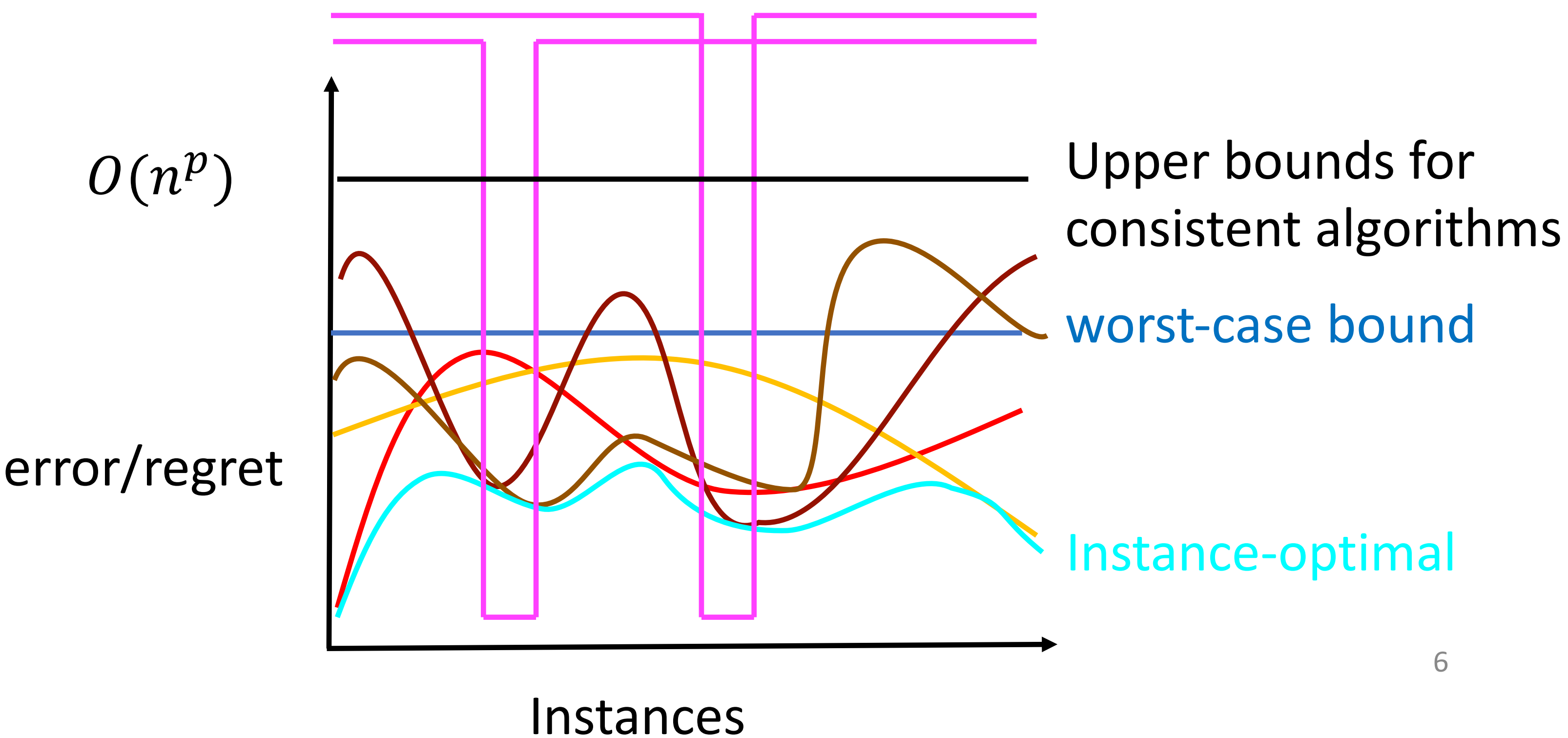
Instance-optimal algorithms in RL

- Can one alg. outperform all other reasonable algorithms on every instance?
    - Reasonable algorithms shouldn't completely fail on any instance
- [Lai & Robbins, 85] for bandits/RL:
    - reasonable := non-trivial regret on all instances



Consistent algorithms and instance-optimality

An algorithm is *consistent* if it achieves $O(n^p)$ regret for every $p > 0$ on every instance

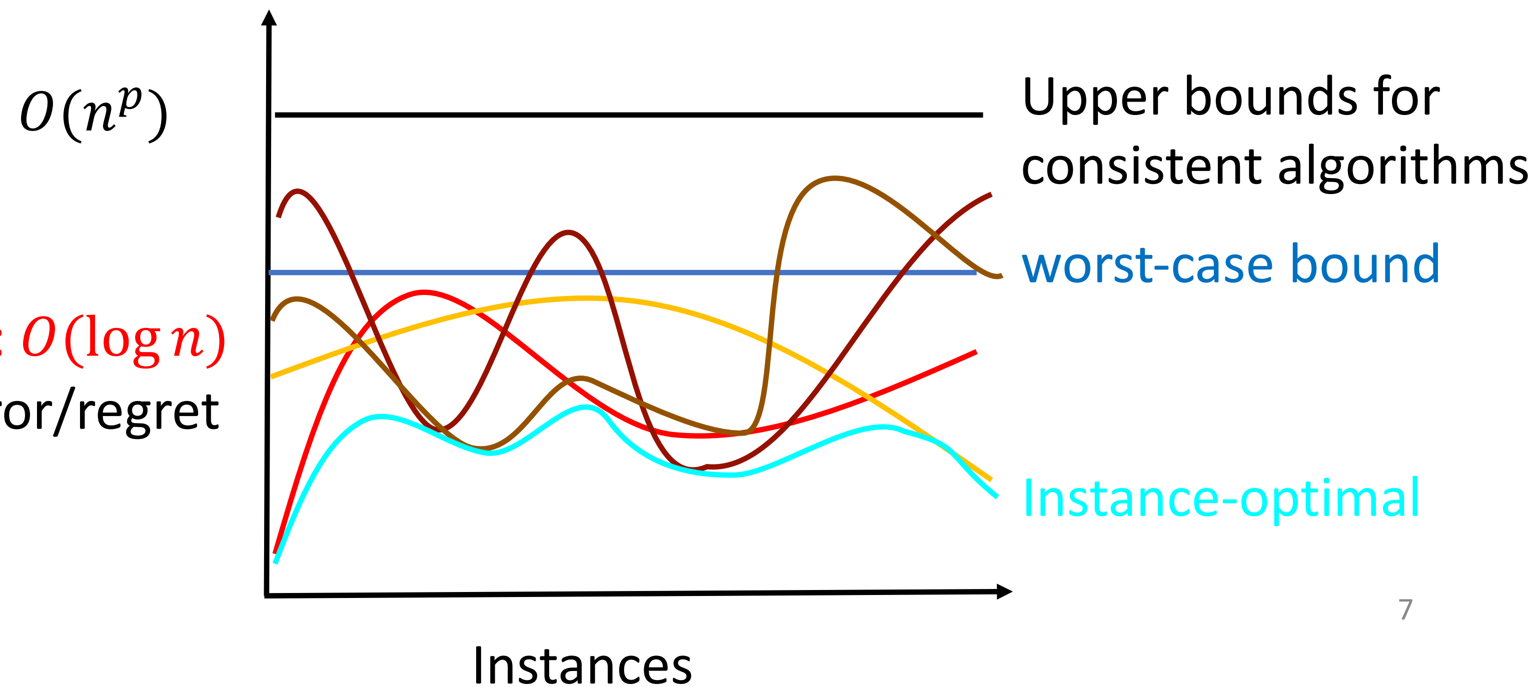


Consistent algorithms and instance-optimality

An algorithm is *consistent* if it achieves $O(n^p)$ regret for every $p > 0$ on every instance

An algorithm is instance-optimal if its regret is as good as *every* consistent algorithm on *every* instance

- Replacing $O(n^p)$ to $O(n^{0.9})$ in the def. only affects constant factor



Prior works

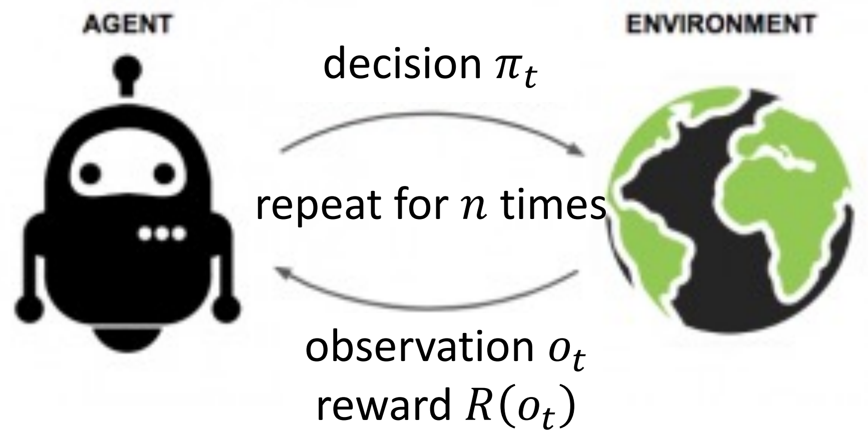
- Instance-optimal algorithms for multi-armed bandits, linear bandits, contextual bandits, ergodic MDPs, etc.

This paper

- We design instance-optimal algorithms for general interactive decision making problems with finite actions (including episodic MDPs)
- We derive the optimal rate of instance-optimal algorithms

Graves & Lai, 1997; Lai & Robbins, 1985; Lattimore & Szepesvari, 2017; Li et al., 2022; Tirinzoni et al., 2021

Interactive decision making



- Includes RL, POMDP, contextual bandits, multi-armed bandits…
  - E.g., episodic RL, $o_t = (s_1, a_1, r_1, s_2, a_2, r_2 \cdots, s_H, a_H, r_H)$
- An instance ⇔ an environment (e.g., a MDP in RL)
- Goal: minimize regret

$$\sum_{t=1}^{n}\big(R(\pi^\star) - R(\pi_t)\big)$$

- NB: the abstract view helps simplify the analysis

Foster, et al. The statistical complexity of interactive decision making, 2021
Figure from https://vitalflux.com/reinforcement-learning-real-world-examples/

Main contributions: an overview

- The exact leading term of the optimal asymptotic regret: $\mathcal{C}(f)\ln n$ on every instance f

Theorem (lower bound): on every instance f, the regret of every consistent algorithm must have

$$\limsup_{n\to\infty} \frac{\mathrm{Reg}_{f,n}}{\ln n} \geq \mathcal{C}(f).$$

Theorem (upper bound): with mild conditions, there exists an algorithm whose regret on every instance f satisfies

$$\limsup_{n\to\infty} \frac{\mathrm{Reg}_{f,n}}{\ln n} \leq \mathcal{C}(f).$$

- The first asymptotic instance-optimal algorithm for general interactive decision making

Additional Notations

- Decision: $\pi \in \Pi$ (we assume Π is finite)

- Instance: $f \in \mathcal{F}$
  - $f[\pi]$: the distribution of observations
- Reward: $R_f(\pi)$
- Optimal decision: $\pi^\star(f) \stackrel{\text{def}}{=} \text{argmax}_{\pi\in\Pi} R_f(\pi)$

- **Assumption**: the optimal decision is unique for every $f \in \mathcal{F}$

The complexity measure $\mathcal{C}(f)$: the intuition

- f: the true instance
- g: another instance in $\mathcal{F}$ with a different optimal decision

Claim: every consistent algorithm must distinguish g from f

- This is because sublinear regret means:
  - on f, play $\pi^\star(f)$ for most of the time
  - on g, play $\pi^\star(g)$ for most of the time
- Failing to distinguish f and $g \Rightarrow \Theta(n)$ regret on f or g
- Therefore, every consistent algorithm must distinguish f, g with prob. $\approx 1/n$

Distinguishability = Large KL divergence

- f: the true instance
- g: another instance in $\mathcal{F}$ with a different optimal decision

Lemma [Chernoff, 59]: Given a sequence of decisions $\pi_1, \cdots, \pi_m$ and corresponding observations $o_1, \cdots, o_m$, for any $\delta > 0$, the following statements are equivalent under mild assumptions:

1. There is an estimator $\hat{f}$ that distinguishes f, g in the sense that
$$\Pr_f(\hat{f} = f) \geq 1 - o(1), \qquad \Pr_g(\hat{f} = f) < \delta$$
2. $\mathbb{E}_f[\sum_{i=1}^m \mathrm{KL}(f[\pi_i] \parallel g[\pi_i])] > \ln(1/\delta)$

- We need $\delta = \tilde{O}(1/n)$, hence

$$\sum_{\pi \in \Pi} w_\pi \, \mathrm{KL}(f[\pi] \parallel g[\pi]) = \mathbb{E}_f[\sum_{i=1}^m \mathrm{KL}(f[\pi_i] \parallel g[\pi_i])] \geq \ln n$$

$w_\pi \stackrel{\text{def}}{=} \mathbb{E}_f[\sum_{i=1}^m 1[\pi_i = \pi]]$: the unnormalized "frequency" of decision π

The complexity measure $\mathcal{C}(f)$

- Find the frequency of decisions $w \in \mathbb{R}_+^\Pi$ that
  - Minimize the regret
  - Collect enough information to distinguish f and g

$$\mathcal{C}(f,n) \stackrel{\text{def}}{=} \min_{w \in \mathbb{R}_+^\Pi} \sum_{\pi\in\Pi} w_\pi \left(R_f\left(\pi^\star(f)\right) - R_f(\pi)\right)$$
$$\text{s.t.} \sum_{\pi\in\Pi} w_\pi \,\mathrm{KL}(f[\pi] \parallel g[\pi]) \geq 1, \ \forall g \in \mathcal{F}, \pi^\star(g) \neq \pi^\star(f)$$
$$\|w\|_\infty \leq n$$

- The final complexity measure:
$$\mathcal{C}(f) \stackrel{\text{def}}{=} \lim_{n\to\infty} \mathcal{C}(f,n)$$
- The constraint $\|w\|_\infty \leq n$ is to ensure mathematical rigor, and it's possible that $\lim_{n\to\infty} \mathcal{C}(f,n) \neq \mathcal{C}(f,\infty)$

The lower bound

Theorem (lower bound): on every instance f, the regret of every consistent algorithm must have

$$\limsup_{n\to\infty} \frac{\mathrm{Reg}_{f,n}}{\ln n} \geq \mathcal{C}(f).$$

- The proof is similar to prior works
- $\mathcal{C}(f)$ recovers the instance-optimal bounds for multi-armed bandits and linear bandits, and improves instance-dependent bounds for tabular RL

Lai & Robbins, 1985; Lattimore & Szepesvari, 2017; Li et al., 2022; Tirinzoni et al., 2021

Instance-optimal algorithm via hypothesis testing

- Our algorithm: collect samples with the best tradeoff between
  - incurring minimal regret on f
  - distinguishing g from f (hypothesis testing!)
- Essentially, we reduce the problem to *hypothesis testing* with active data collection

Algorithm: Test-to-Commit (T2C)

Goal: $\limsup_{n\to\infty} \frac{\mathrm{Reg}_{f,n}}{\ln n} \le \mathcal{C}(f)$ for every $f \in \mathcal{F}$

- Step 1: Initialization
  - Explore uniformly for $o(\ln n)$ steps and use MLE to get $\hat{f}$
- Step 2: Identification
  - Compute the best "frequency" w_π by solving $\mathcal{C}(\hat{f})$ and collect $\Theta(\ln n)$ samples correspondingly
  - Hypothesis testing: is $\hat{f}$ the true instance?
- Step 3: Exploitation
  - If $\hat{f}$ passes the test: run $\pi^\star(\hat{f})$ forever
  - Otherwise: run UCB
- Inspired by [Lattimore & Szepesvari, 2017]

The end of optimism? an asymptotic analysis of finite-armed linear bandits. Lattimore & Szepesvari, 2017

The key step: Identification

- The estimation $\hat{f}$ in Step 1 is not accurate enough
  - With only $o(\ln n)$ samples, failure prob. $> n^{-0.1}$
- We boost the failure prob. to n^{-1} via hypothesis testing
- Solve $\mathcal{C}(\hat{f}, (\ln \ln n)^{1/4})$ to get w_π (the "frequency" of decisions) and run decision π for $(w_\pi \ln n)$ rounds
  - Incur $\mathcal{C}\big(\hat{f}\big)\ln n + o(\ln n)$ regret
  - Get enough information: $\sum_{i=1}^{m} \mathrm{KL}(\hat{f}[\pi_i] || g[\pi_i]) \geq \ln n$ for every g with a different optimal decision
- Run log-likelihood ratio test on $\hat{f}$:

$$\mathcal{E}^{\hat{f}} = \mathbb{I}\left[\forall g \in \mathcal{F} \text{ and } \pi^\star(g) \neq \pi^\star\big(\hat{f}\big), \sum_{i=1}^{m} \ln \frac{\hat{f}[\pi_i](o_i)}{g[\pi_i](o_i)} \geq \ln n\right]$$

Guarantees (when $\mathcal{F}$ is finite):

1. When $\hat{f} = f$, $\Pr\big(\mathcal{E}^{\hat{f}}\big) \geq 1 - o(1)|\mathcal{F}| = 1 - o(1)$
2. When $\pi^\star\big(\hat{f}\big) \neq \pi^\star(f)$, $\Pr\big(\mathcal{E}^{\hat{f}}\big) < 1/n$

Sequential design of experiments, Chernoff, 1959

Regret analysis for finite hypothesis

Guarantees of Step 2:

1. When $\hat{f} = f$, $\Pr(\mathcal{E}^{\hat{f}}) \geq 1 - o(1)$
2. When $\pi^\star(\hat{f}) \neq \pi^\star(f)$, $\Pr(\mathcal{E}^{\hat{f}}) < 1/n$

- In Step 1, $\hat{f} = f$ with probability at least $1 - O\big((\ln n)^{-1}\big)$ due to convergence of MLE estimators
- When $\hat{f} = f$
  - Step 2 has regret $\mathcal{C}(f) \ln n$
  - Step 3 has regret $O(\ln n)$ with probability at most $o(1)$
- When $\hat{f} \neq f$
  - Step 2 has regret $O(\ln n\ (\ln\ln n)^{1/4})$
  - Step 3 has regret $O(n)$ with probability at most $1/n$; otherwise has regret at most $O(\ln n)$
- Overall expected regret: $\mathcal{C}(f)\ln n + o(\ln n)$

Sequential design of experiments, Chernoff, 1959

Asymptotic regret of T2C

Theorem (upper bound): with mild conditions, the regret of the T2C algorithm on every instance f satisfies

$$\limsup_{n\to\infty} \frac{\mathrm{Reg}_{f,n}}{\ln n} \le \mathcal{C}(f).$$

- Instantiated on tabular episodic RL, T2C is the first instance-optimal algorithm

Extension to infinite hypothesis class

- In Step 1, we need to prove $\hat{f}$ is close to f (instead $\hat{f} = f$) with probability $1 - o(1)$ in the sense that
  - $\pi^\star(\hat{f}) = \pi^\star(f)$
  - The solution of $\mathcal{C}(\hat{f})$ is close to that of $\mathcal{C}(f)$
  - $\hat{f}$ can pass the log-likelihood ratio test
  - Essentially we need $\mathrm{KL}(\hat{f}[\pi] \parallel g[\pi]) \approx \mathrm{KL}(f[\pi] \parallel g[\pi]), \forall g$
- In Step 2, we need a small covering number for $\mathcal{F}$ to prove uniform concentration

Takeaways

- Instance-optimality can be achieved by hypothesis testing with active data collection
  - Key step: tradeoff between collecting information and incurring regret
- The abstraction (interactive decision making) helps simplify the analysis
  - we only need some classic assumptions (e.g., uniform convergence) on the distribution of observations

Remaining open questions

- Improving T2C
  - Computational efficiency on concrete settings
  - Instance-optimality for infinite decisions
  - Non-asymptotic performance [Wagenmaker & Foster, 23]
- Instance-optimality for other questions: sample complexity (instead of regret), offline RL, supervised learning
  - How to define consistent algorithms?
  - Does instance-optimal algorithm exist?

Thank you for your attention!