Learning in Stackelberg Games with Non-myopic Agents

Nika Haghtalab[1] Thodoris Lykouris[2] Sloan Nietert[3] Alexander Wei[1]

[1] UC Berkeley [2] MIT [3] Cornell

Setup

Repeated Stackelberg Games

Each of T rounds:

- **Principal** commits to action $x_t \in \mathcal{X}$
- **Agent** responds with action $y_t \in \mathcal{Y}$
- Realize payoffs $u(x_t, y_t)$, $v(x_t, y_t)$
  - Agent's v is **unknown** to principal and must be learned
- Both observe actions x_t and y_t

Learning Objective and Regret

Principal aims to learn optimal action while minimizing **Stackelberg regret**

$$\max_{x \in \mathcal{X}} \mathbb{E}\left[\sum_t u(x, y_t) - u(x_t, y_t)\right].$$

Agent Behavior

Assume that agent is non-myopic, but γ-discounting, i.e., agent has payoff

$$\sum_t \gamma^t v(x_t, y_t).$$

- When $\gamma \to 0$, agent myopically best responds: $y_t \in \mathrm{br}(x_t)$
- When $\gamma \to 1$, no assumption on agent behavior

Agent maximizes expected discounted payoff given principal's policy

Overview

Against myopic (best responding) agents, principal in Stackelberg game can easily learn optimal behavior

Challenge: **Non-myopic agents** may have incentive to not best respond, to "mislead" principal's learning for higher future payoffs

Goal: Robust learning algorithms that still learn with discounting, non-myopic agents in Stackelberg games

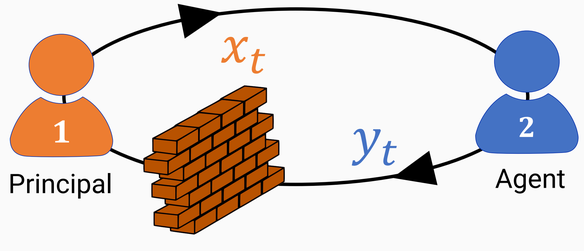
Framework

Learning with Delays

Idea: Delay principal's response to feedback

- Discounting agent won't deviate too far if they don't benefit w/in effective time horizon $T_\gamma := 1/(1-\gamma)$.
- Not "too far" = ε-approximate best response:

$$v(x_t, y_t) \geq v\big(x_t, \mathrm{br}(x_t)\big) - \varepsilon$$

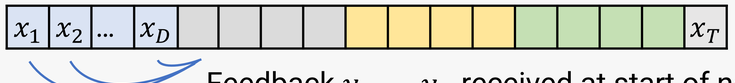


Information barrier via delays/batching incentivizes approximate best responses

Delayed Bandits ⇔ Batched Bandits

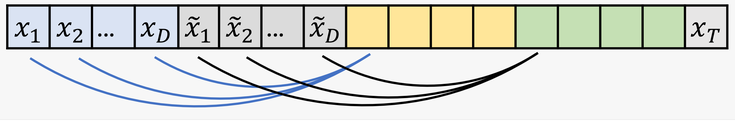
Reduces **delays of length D** to **batches of size $O(D)$** (and vice versa). Holds for all bandit problems!

Policy $\mathcal{A}$ selects principal actions in **batches** of D



Feedback $y_1, \ldots, y_D$ received at start of next batch

Interleaving copies $\mathcal{A}, \tilde{\mathcal{A}}$ gives policy w/ **delay** $D-1$



Algorithmic Desiderata

1. Efficient optimization of $f(x_t) = u\big(x_t, \mathrm{br}(x_t)\big)$ from **bandit** feedback
2. **Robustness** to misspecified $f(x_t)$ from approx. best responses. Two regimes:
   - $\mathcal{Y}$ continuous: $\|y_t - \mathrm{br}(x_t)\|$ bounded
   - $\mathcal{Y}$ discrete: $\{x_t : y_t \neq \mathrm{br}(x_t)\}$ bounded
3. **Non-adaptivity** to feedback *while still learning*: minimal overhead from batching

Applications

For each: Framework + reduction ⇒ learning algorithm for non-myopic agents

Stackelberg Security Games

- Principal commits to mixed strategy over n targets to defend; then agent chooses a single target to attack
- We give nearly optimal learning algorithm CLINCH for *myopic* agents
  - Key tool: **Grünbaum's theorem**

Demand Learning

- Principal sets price; then agent with stochastic utility fn reports demand
- Revenue maximization reduces to a stochastic bandit problem [KL07]
- We solve via efficient *batched* version of ACTIVEARMELIMINATION

+ Strategic Classification (and more)