No-Regret Learning for Trustworthy Online Decision-Making

Mahsa Ghasemi

Automated Knowledge Acquisition and DEcision-MakIng (AKADEMI) Group

Workshop on “Trustworthy AI-Enabled Cyber-Physical Systems”

85th IFIP WG 10.4 Meeting

February 2, 2024

PURDUE UNIVERSITY

Elmore Family School of Electrical and Computer Engineering

Trustworthy Online Decision-Making

Online Learning for Adversarial Markov Decision Processes

"No-Regret Learning with High-Probability in Adversarial Markov Decision Processes,"
M. Ghasemi*, A. Hashemi*,
H. Vikalo, and U. Topcu
UAI 2021

"No-Regret Learning in Dynamic Stackelberg Games,"
N. Lauffer, M. Ghasemi, A. Hashemi,
Y. Savas, and U. Topcu
TAC 2023

Online Learning in Two-Player Games

Online Learning with Causal Structure

"Approximate Allocation Matching for Structural Causal Bandits with Unobserved Confounders",
L. Wei, Q. Elahi, M. Ghasemi, and M. Kocaoglu
NeurIPS 2023

No-Regret Learning in Dynamic Stackelberg Games

[N. Lauffer, M. Ghasemi, A. Hashemi, Y. Savas, and U. Topcu, TAC 2023]

Motivating concepts

Non-cooperative multi-agent systems

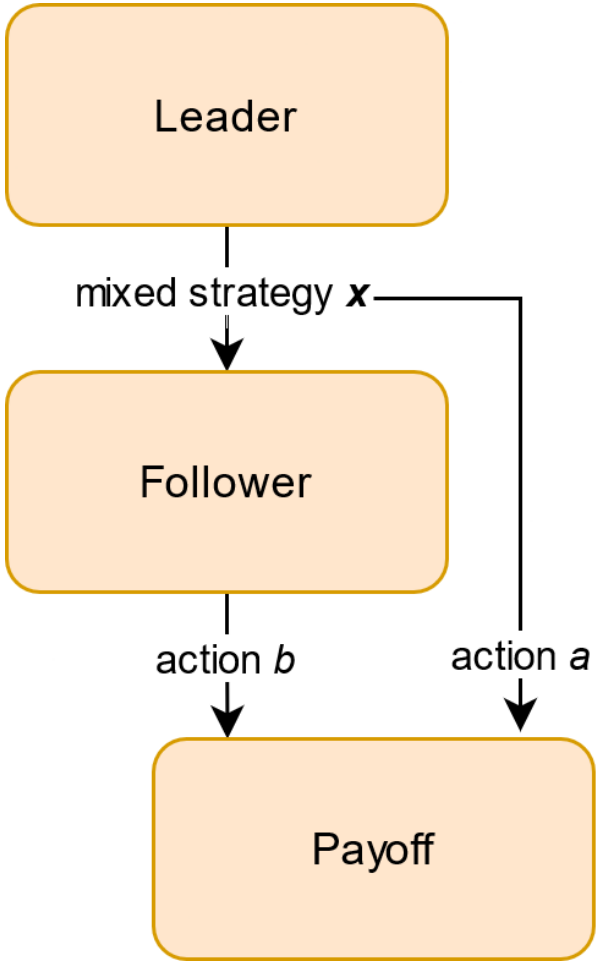
Sequential decision making in an environment

Online learning

Background

Stackelberg games [1934]

1. The *leader* plays a *mixed strategy* ***x***
2. The *follower* plays an action *b* in response
3. An action *a* is sampled from ***x***
4. The *leader* and *follower* receive payoff *r(a,b)* and *u(a,b), respectively*



Applications of Stackelberg games

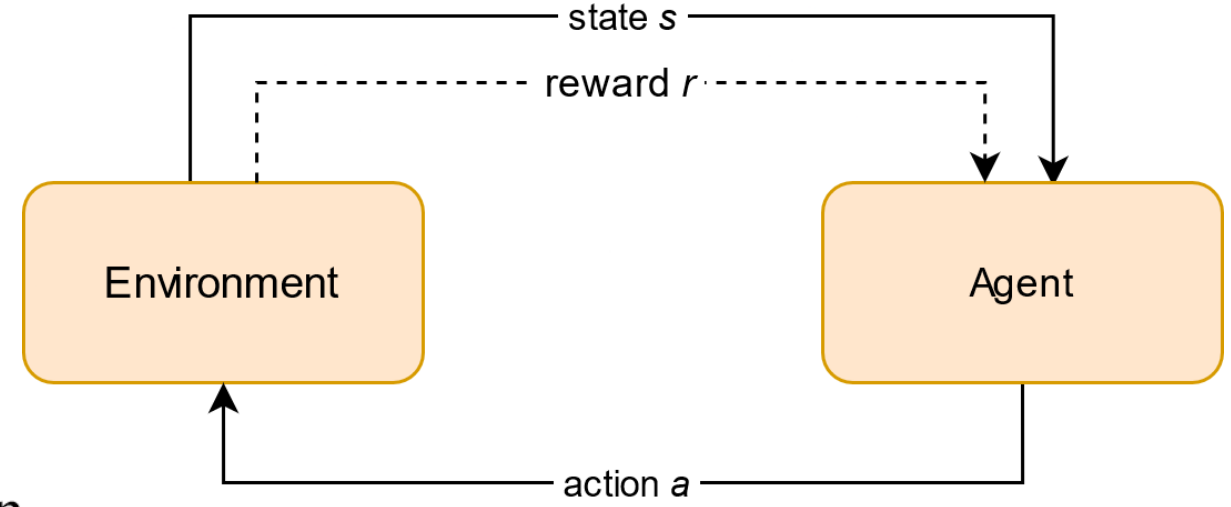
Security scheduling at the LA airport

Randomized patrol routes by the US Coast Guard

Park ranger patrol patterns to fight illegal poaching

Markov decision process

Defined by a tuple $(\mathcal{S}, \mathcal{A}, r, P)$:



- $\mathcal{S}$ is the state space.
- $\mathcal{A}$ is the action space.
- $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function.
- $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition function.

Dynamic Stackelberg game

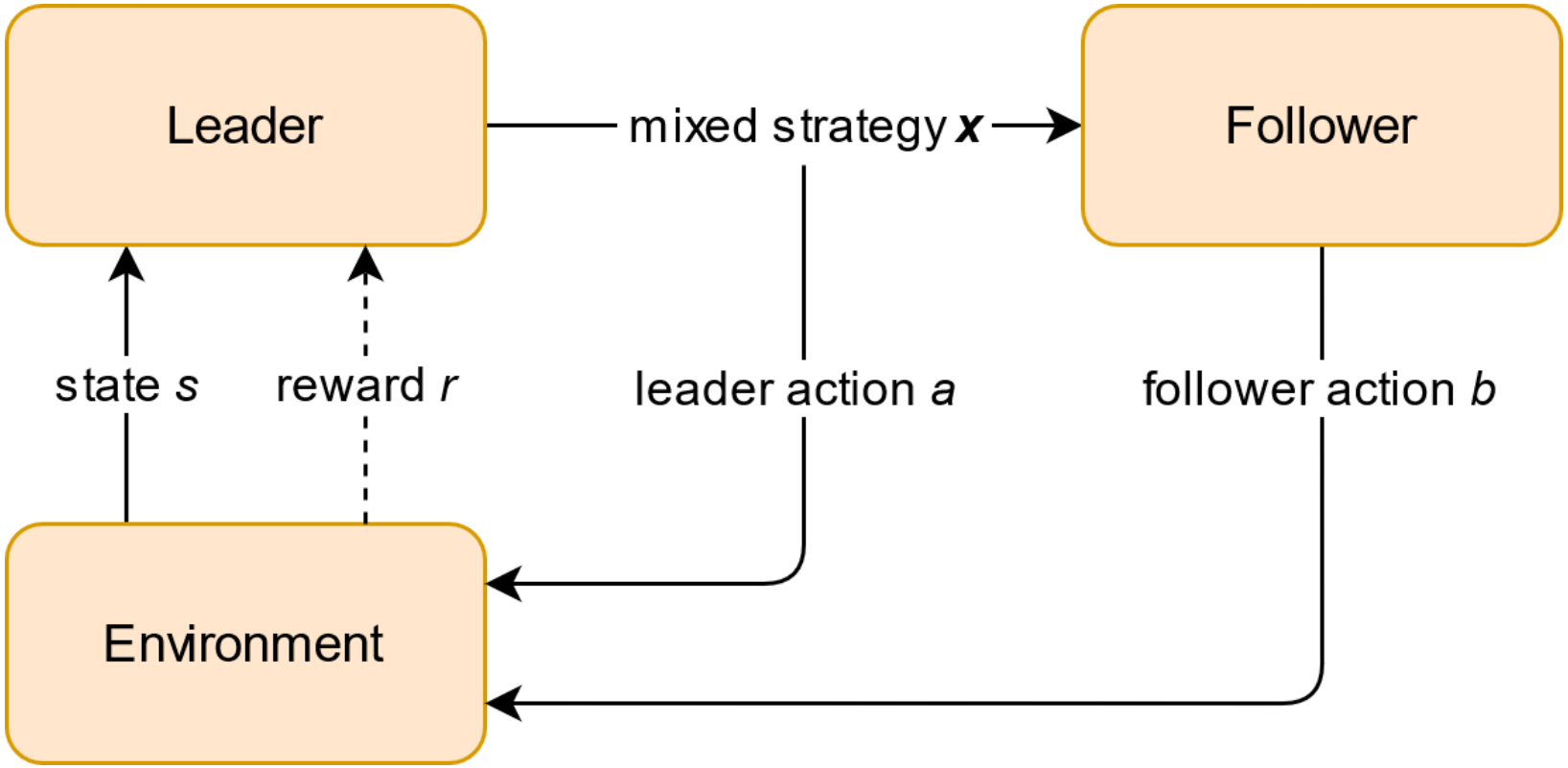
Played on a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{B}, r, u, P)$ defined as follows.

- $\mathcal{S}$ is the state space.
- $\mathcal{A}$ is the set of actions available to the leader.
- $\mathcal{B}$ is the set of actions available to the follower.
- $r : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \mathbb{R}$ is the reward function for the leader agent.
- $u : \mathcal{A} \times \mathcal{B} \to \mathbb{R}$ is the utility function for the follower.
- $P : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \times \mathcal{S} \to [0, 1]$ is the transition function.

Dynamic Stackelberg game

1. The leader observes state *s*
2. The *leader* plays a *mixed* strategy ***x***
3. The *follower* plays an action *b* in response
4. An action *a* is sampled from ***x***
5. The leader and follower receive payoff *r(s,a,b)* and *u(a,b)*
6. The state transitions to *s'* according to probabilities *P(s,a,b,s')*

Dynamic Stackelberg game



Related work

Repeated Stackelberg games [Balcan et al., 2018], [Blum et al., 2014]: *repeated interactions, but without dynamics*

Stochastic games [Wei et al., 2017], [Ouyang, et al., 2017]: *agents choose actions simultaneously*

Feedback Stackelberg games [Li and Sethi, 2017], [Chen and Cruz, 1972]: *typically studied in continuous settings modeled by differential equations with perfect information*

Connection to *model-free* RL

A dynamic Stackelberg game $(\mathcal{S}, \mathcal{A}, \mathcal{B}, r, u, P)$, can be reduced to a Markov decision process $(\mathcal{S}, \Delta(\mathcal{A}), r', P')$.

$$P'(s, \mathbf{x}, s') = \mathop{\mathbb{E}}_{a \sim \mathbf{x}} [P(s, a, \varphi(\mathbf{x}), s)]$$

$$r'(s, \mathbf{x}) = \mathop{\mathbb{E}}_{a \sim \mathbf{x}} [r(s, a, \varphi(\mathbf{x}))]$$

Model-free RL, e.g., Q-Learning and SARSA

Regret dependent on the number of states

Learning Problem, Regret, and Assumptions

Online learning in a dynamic Stackelberg game

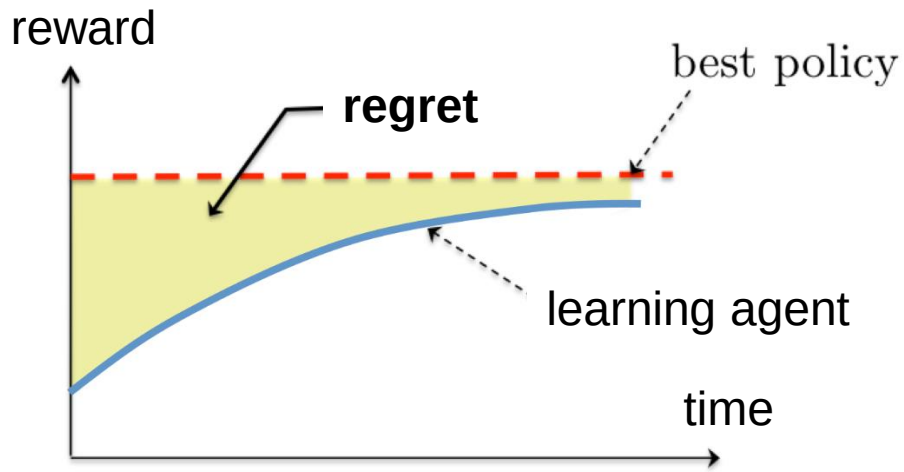
Problem 1. *Suppose the follower's utility function u is fixed and unknown. Give an online learning algorithm that computes policies $\pi_{t,h} : \mathcal{S} \to \Delta(\mathcal{A})$ for each episode t and time step h that minimize the leader agent's regret.*

$$R_T = \sup_{\pi \in \Pi} \mathbb{E}[\sum_{t=1}^{T} \sum_{h=1}^{H} R(s_{t,h}^{\pi}, \mathbf{x}_{t,h}^{\pi})] - \mathbb{E}[\sum_{t=1}^{T} \sum_{h=1}^{H} R(s_{t,h}, \mathbf{x}_{t,h})]$$

set of *all* policies

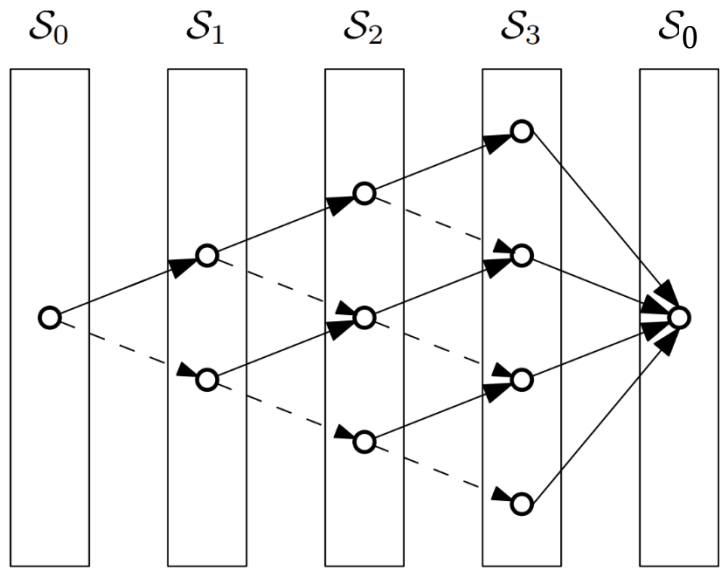
the best policy's reward

the learner's reward



Episodic state space

- States are partitioned into *H* layers.
- Transitions only exist from one layer to the next.



Linear function approximation

Assume that the follower's utility function is *linearly parameterized*.

$$u(a, b) = \langle f(a, b), \boldsymbol{\theta}^* \rangle$$

for some function $f : \mathcal{A} \times \mathcal{B} \to \mathbb{R}^p$ and parameter $\boldsymbol{\theta}^*$.

For each $b \in \mathcal{B}$, we have a *feature matrix* $\mathbf{M}_b \in \mathbb{R}^{n \times p}$.

$$[\mathbf{M}_b]_i = f(a_i, b)$$

Strong Stackelberg equilibrium

Tie broken in leader's favor

$$\mathop{\mathbb{E}}_{a\sim\mathbf{x}}[r(s,a,b)] \geq \mathop{\mathbb{E}}_{a\sim\mathbf{x}}[r(s,a,b')]$$

Ensures the best in hindsight policy exists

Ensures the following inequality can be active

$$\mathop{\mathbb{E}}_{a\sim\mathbf{x}}[u(a,b)] \geq \mathop{\mathbb{E}}_{a\sim\mathbf{x}}[u(a,b')]$$

The Learning Scheme

Learning from past observations

After we play a mixed strategy $\mathbf{x} \in \Delta(\mathcal{A})$, we observe a response $b \in \mathcal{B}$. Then, we know that $\forall b' \in \mathcal{B}$,

$$\mathop{\mathbb{E}}_{a \sim \mathbf{x}}[u(a, b)] \geq \mathop{\mathbb{E}}_{a \sim \mathbf{x}}[u(a, b')]$$

Learning from past observations

After we play a mixed strategy $\mathbf{x} \in \Delta(\mathcal{A})$, we observe a response $b \in \mathcal{B}$. Then, we know that $\forall b' \in \mathcal{B}$,

$$\begin{aligned}
\mathop{\mathbb{E}}_{a \sim \mathbf{x}}[u(a,b)] &\geq \mathop{\mathbb{E}}_{a \sim \mathbf{x}}[u(a,b')] \\
\mathop{\mathbb{E}}_{a \sim \mathbf{x}}[\langle f(a,b), \boldsymbol{\theta}^* \rangle] &\geq \mathop{\mathbb{E}}_{a \sim \mathbf{x}}[\langle f(a,b'), \boldsymbol{\theta}^* \rangle] \\
\langle \mathbf{x}^T \mathbf{M}_b, \boldsymbol{\theta}^* \rangle &\geq \langle \mathbf{x}^T \mathbf{M}_{b'}, \boldsymbol{\theta}^* \rangle \\
\mathbf{x}^T \mathbf{M}_b \boldsymbol{\theta}^* &\geq \mathbf{x}^T \mathbf{M}_{b'} \boldsymbol{\theta}^* \\
\mathbf{x}^T (\mathbf{M}_b - \mathbf{M}_{b'}) \boldsymbol{\theta}^* &\geq 0.
\end{aligned}$$

The learning scheme

1. Maintain a *version space* of what $\boldsymbol{\theta}^*$ could be given past observations.

$$\Theta = \{\boldsymbol{\theta} \in \mathbb{R}^p \mid ||\boldsymbol{\theta}|| = 1 \wedge \forall i, b' \; \mathbf{x}_i^T(\mathbf{M}_{b_i} - \mathbf{M}_{b'})\boldsymbol{\theta} \geq 0\}$$

The learning scheme

1. Maintain a *version space* of what $\boldsymbol{\theta}^*$ could be given past observations.

$$\Theta = \{\boldsymbol{\theta} \in \mathbb{R}^p \mid ||\boldsymbol{\theta}|| = 1 \wedge \forall i, b' \ \mathbf{x}_i^T(\mathbf{M}_{b_i} - \mathbf{M}_{b'})\boldsymbol{\theta} \geq 0\}$$

2. Solve for an *optimistic* ϵ*-conservative* policy.

value of current state via Bellman equation

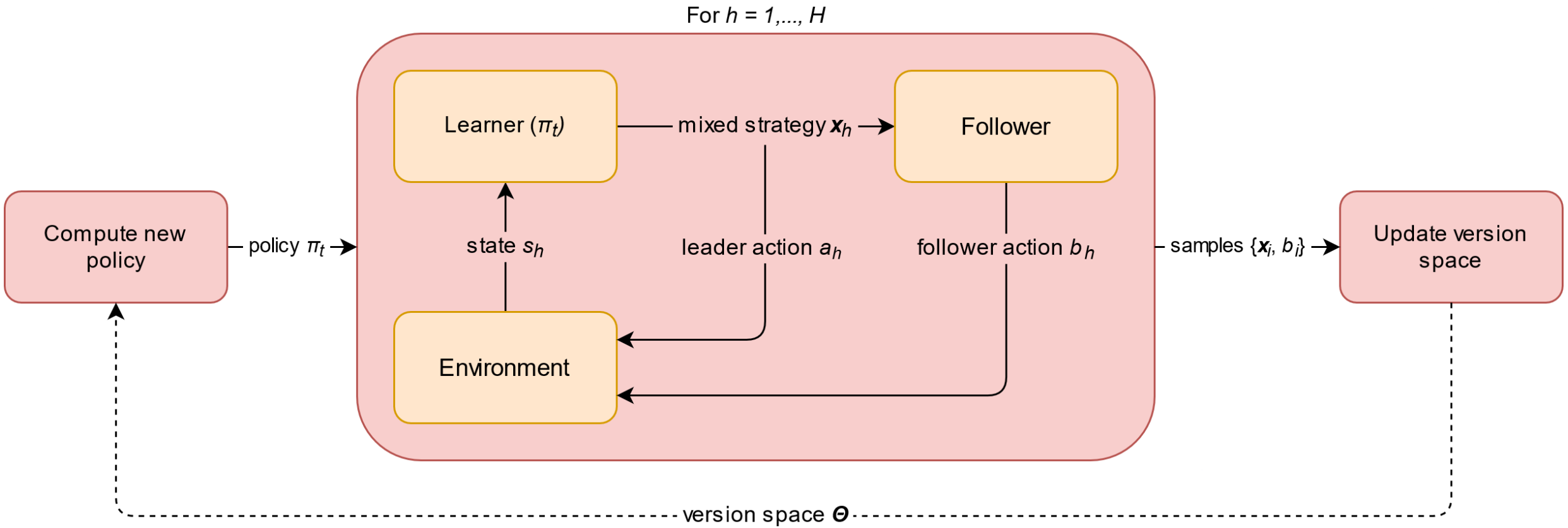
choose θ_s optimistically

mixed strategy over possible actions

choose the policy conservatively

$$\begin{aligned} \max_{\mathbf{x}_s, \boldsymbol{\theta}_s} \quad & \widetilde{V}_t(s) = \mathop{\mathbb{E}}_{a \sim \mathbf{x}_s} \left[r(s,a,b) + \sum_{s'} P(s,a,b,s') \widetilde{V}_t(s') \right] \\ \text{s.t.} \quad & \boldsymbol{\theta}_s \in \Theta \\ & \mathbf{x}_s \in \Delta(\mathcal{A}(s)) \\ & \mathbf{x}_s^T(\mathbf{M}_b - \mathbf{M}_{b'})\boldsymbol{\theta}_s \geq \epsilon, \ \forall b' \in B, b' \neq b \end{aligned}$$

The learning scheme



No-regret learning with high-probability

- *With high probability, the regret is upper bounded such that it is*
  - *Independent of the size of leader's state space (S)*
  - *Sublinear in the size of follower's action space:* $\sqrt{m}$
  - *Linear in the size of leader's action space (n) and episode length (H)*
  - *Depends on the number of rounds (T) and follower's features (p):* $(T)^{1-\frac{1}{p}}$
  - Tight w.r.t. *p* and *T* [Zhao, Zhu, Jiao, Jordan, ICML 2023]

No-regret learning with high-probability

Theorem 1: (high-probability regret bound)

Let $\delta \in (0,1)$. With probability at least $1-\delta$,

$$R_T \leq (T)^{1-\frac{1}{p}} \left(d\sqrt{mn}(1+\sqrt{nH})H + H \right) + H\sqrt{\frac{T}{2}\ln\left(\frac{1}{\delta}\right)}$$

constant

number of episodes

number of features representing *u*

number of follower actions

number of leader actions

size of episodes

Proof overview

Sources of regret:

1. From making *mistakes* (the follower plays an unexpected action) we get $\mathcal{O}(\epsilon^{-p})$ regret.
2. From choosing ϵ-conservative policies we get $\mathcal{O}(\epsilon n\sqrt{m})$ regret.

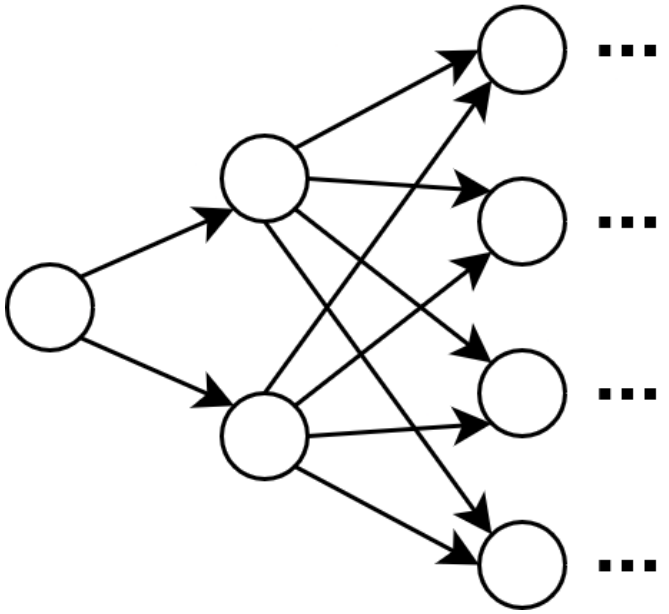
$$\mathbf{x}_s^T(\mathbf{M}_b - \mathbf{M}_{b'})\boldsymbol{\theta}_s > \epsilon,\ \forall b' \in \mathcal{B}$$

Experimental Results

Overview

1. Experimentally verify how regret scales in parameters of the game

2. Comparing against other policies:
   - The optimal policy
   - *SARSA*
   - A random policy



Structure of the state space.

Varying *p*

Parameters

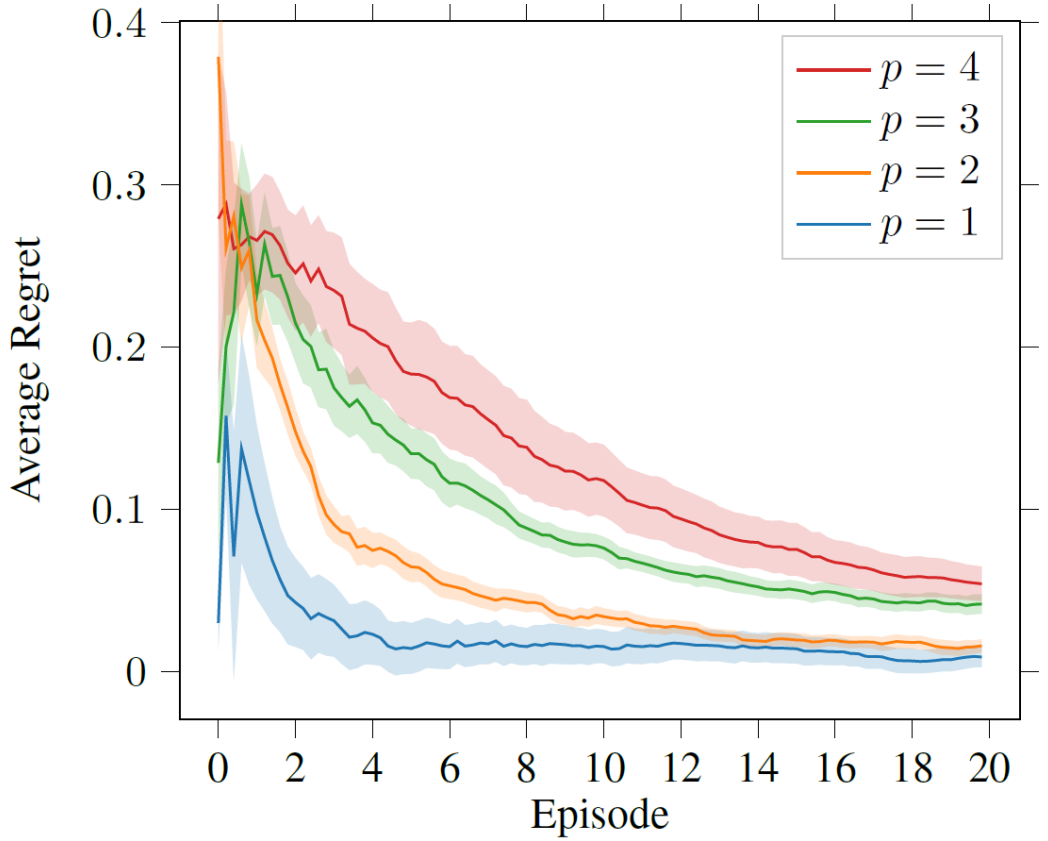
$\mathcal{S}$ = (1,2,4,8,16)

n = 4

m = 4

Average Regret

$$\frac{1}{tH}\sum_{i=1}^{t}\sum_{h=1}^{H} R_{i,h}^{\pi^*} - R_{i,h}^{\pi_i}$$



The rate of convergence of the algorithm depends on the number of features representing the follower's utility function.

Varying $|\mathcal{S}|$

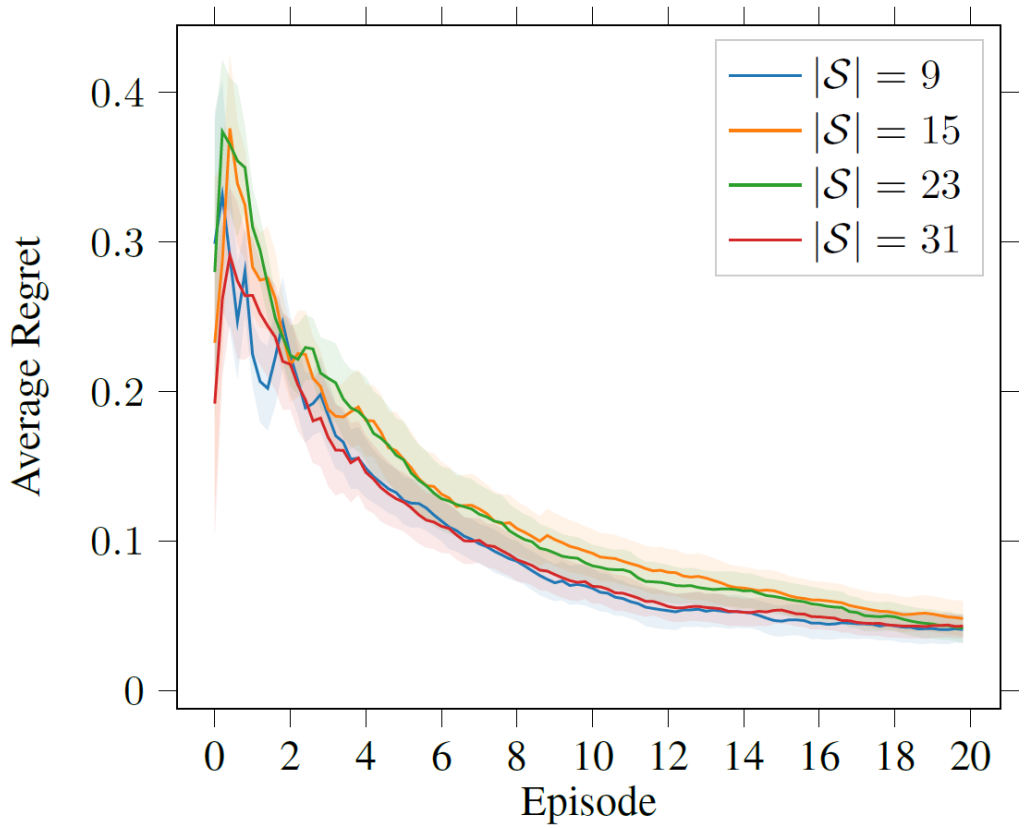
Parameters

p = 4

n = 4

m = 4

	$\mathcal{S}_0$	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{S}_4$
●	1	2	2	2	2
●	1	2	4	4	4
●	1	2	4	8	8
●	1	2	4	8	16



The regret of the algorithm is independent of the size of the state space.

Comparing policies

Parameters

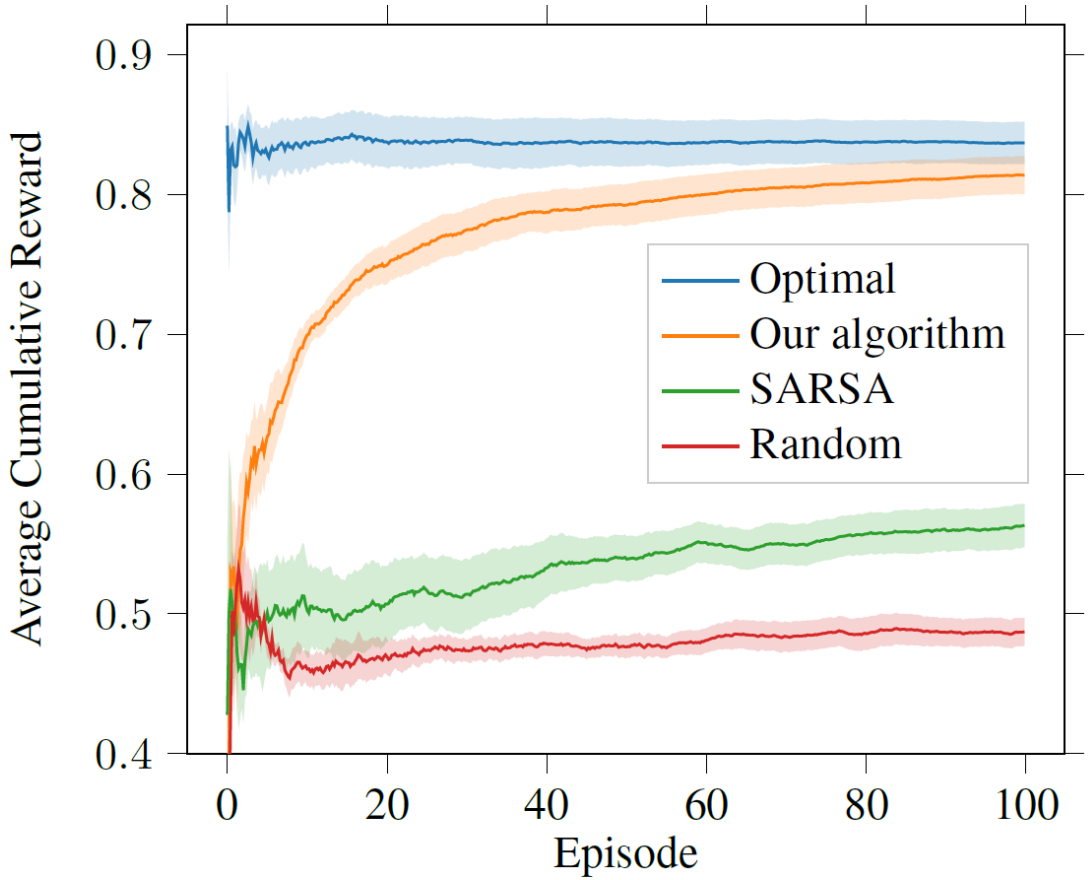
$\mathcal{S}$ = (1,2,2,2,2)

n = 4

m = 4

Average Cumulative Reward

$$\frac{1}{tH}\sum_{i=1}^{t}\sum_{h=1}^{H}R_{i,h}^{\pi_i}$$



The algorithm outperforms SARSA, even on small state spaces.

Conclusion

- Introduced discrete-time dynamic Stackelberg games
- Developed a novel learning algorithm based on optimistically building ϵ-conservative policies
- Established a no-regret learning bound with high probability