Learning to Play Sequential Games versus Unknown Opponents

Pier Giuseppe Sessa
Ilija Bogunovic
Maryam Kamgarpour
Andreas Krause

ETH zürich

NEURAL INFORMATION PROCESSING SYSTEMS

Repeated Sequential Game Setup

For $t = 1, \dots$:

1) **Learner** picks action: x_t

2) **Opponent** responds with:

$$y_t = b(x_t, \theta_t)$$

Response function — **Type** of opponent at round t

3) Learner obtains reward: $r(x_t, y_t)$

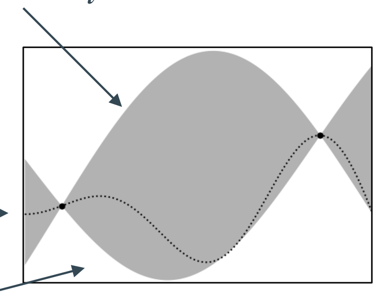
- Response function $b(\cdot,\cdot)$ is **unknown** to the learner. Learner only observes $y_t + \epsilon_t$, with ϵ_t zero-mean sub-Gaussian.
- Opponent's types can be adversarially selected.

Learner's **regret** as a performance indicator:

$$R(T) := \max_{x \in \mathcal{X}} \sum_{t=1}^{T} r(x, b(x, \theta_t)) - \sum_{t=1}^{T} r(x_t, y_t) \,.$$

High-Level Ideas

1. Iteratively learn the opponent response function via **kernel-ridge regression**



2. Use an online learning strategy with **optimistic reward estimates**:

$$\hat{\mathbf{r}}_t[x] = \max_{y} r(x, y) \qquad (1)$$
$$\text{s.t.} \quad y \in [\text{lcb}_t(x, \theta_t), \text{ucb}_t(x, \theta_t)] \,.$$

The **StackelUCB** Algorithm

StackelUCB

Inputs: decision set $\mathcal{X}$ of K actions, kernel k, learning rate η

Initialize mixed strategy $\mathbf{w}_1 = \frac{1}{K}[1, \dots, 1]$

For $t = 1, \dots$:

- Sample action $x_t \sim \mathbf{w}_t$
- Compute reward estimates $\hat{\mathbf{r}}_t[x]$, $\forall x \in \mathcal{X}$ via (1)
- Update $\mathbf{w}_t$ via multiplicative update rule:

$$\mathbf{w}_{t+1}[x] \propto \mathbf{w}_t[x] \cdot \exp\left(\eta \cdot \hat{\mathbf{r}}_t[x]\right) \,, \quad \forall x \in \mathcal{X}$$

- Update lcb$_t(\cdot)$, ucb$_t(\cdot)$ based on observed data

Thm (informal): Assume $\|b(\cdot)\|_k$ is bounded and r is L-Lipschitz. Then, w.h.p.,

$$R(T) \leq \mathcal{O}\left(\sqrt{T \log K}\right) + \mathcal{O}\left(L \gamma_T \sqrt{T}\right) .$$

Kernel-dependent **max info. gain** [2] γ_T :
Maximal uncertainty reduction about $b(\cdot)$ after T noisy observations.
E.g., $\gamma_T = \mathcal{O}((\log T)^{d+1})$ for SE kernels with domain dimension d.

Single Opponent Type (i.e., when $\theta_t = \bar{\theta}\ \forall t$)

Corollary: Consider the strategy: $x_t = \arg\max_{x \in \mathcal{X}} \hat{\mathbf{r}}_t[x]$
Then, w.h.p.,

$$R(T) \leq \mathcal{O}\left(L \gamma_T \sqrt{T}\right) .$$

Stackelberg Games with unknown followers' utilities

$\mathcal{X}$ = n_l-dimensional simplex and $b(\cdot, \theta_t) = \arg\max_y U_f(\cdot, \theta_t)$.

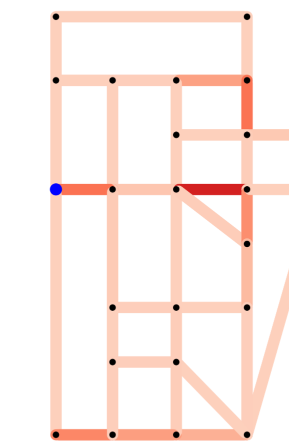
Corollary: When **StackelUCB** is run over a finite discretisation of $\mathcal{X}$, w.h.p.

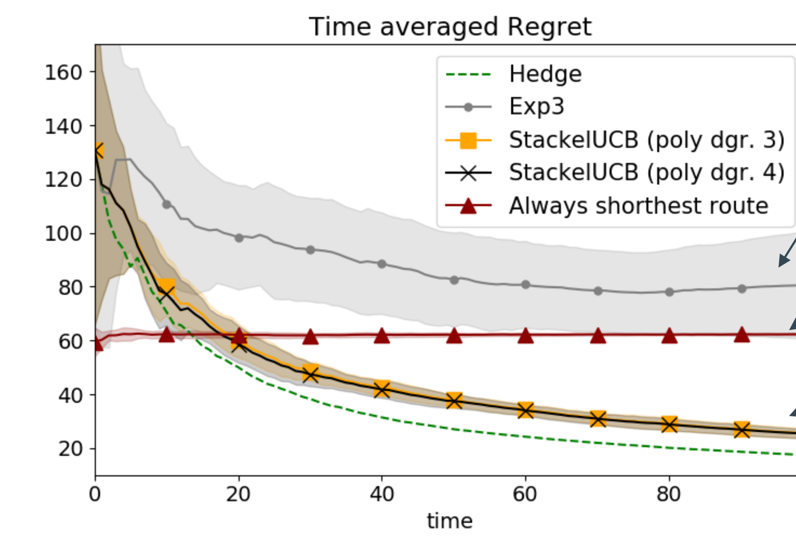
$$R(T) \leq \mathcal{O}\left(\sqrt{T n_l \log\left(L\sqrt{n_l T}\right)}\right) + \mathcal{O}\left(L \gamma_T \sqrt{T}\right) .$$

Routing Vehicles in Congested Traffic Network

1. Network operator chooses routes x_t for fleet of vehicles
2. Based on x_t, users in the network pick their routes and cause congestion: $y_t = b(x_t, \theta_t)$

users' demand profile (i.e., users origins and destinations)





Bandit **Exp3** [3] algorithm which does not learn $b(\cdot)$

Route fleet through the shortest route

StackelUCB with polynomial kernel of degree 3 or 4

Hedge [1] (idealised benchmark) assumes congestion model $b(\cdot)$ is known.

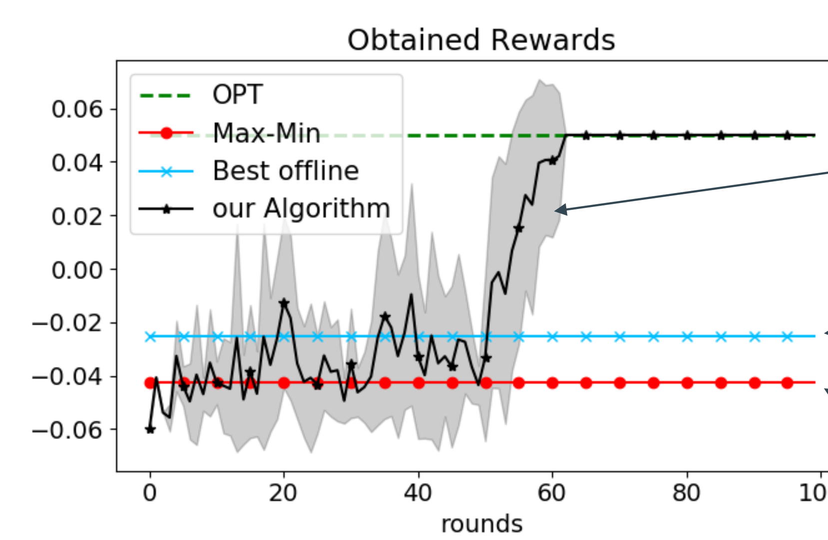
- **StackelUCB** leads to low regret and reduces the network's congestion

Wildlife Protection against Poaching Activities

1. Park rangers choose a patrol strategy x_t (i.e., $x_t[i]$ = probability of patrolling area i in the park)
2. Poachers choose poaching location $y_t = b(x_t, \bar{\theta})$

park animal density

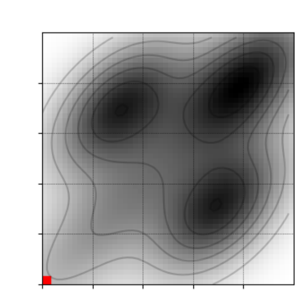
poachers' preference model



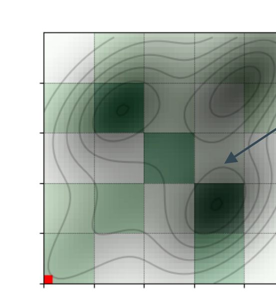
Our algorithm starts with no information and discovers the optimal strategy in ~60 rounds

$\max_x r(x, \hat{b}(x, \bar{\theta}))$, with $\hat{b}$ = estimate over 1000 offline random datapoints.

$\max_x \min_y r(x, y)$



Park animal density



Rangers patrol strategy.

High prob. assigned to areas with high animal density and closer to poachers' base location (unknown to the algorithm)

References

[1] Y. Freund and R. E. Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In *J. Comput. Syst. Sci.*, 1997.

[2] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger. "Gaussian process optimization in the bandit setting: No regret and experimental design". In *ICML*, 2010.

[3] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. "The Nonstochastic Multiarmed Bandit Problem". In *SIAM J. Comput.*, 2003.