Learning to Play Sequential Games 
versus Unknown Opponents

Pier Giuseppe Sessa
Ilija Bogunovic
Maryam Kamgarpour
Andreas Krause
ETH Zürich

Abstract

We consider a repeated sequential game between a learner, who plays first, and an opponent who responds to the chosen action. We seek to design strategies for the learner to successfully interact with the opponent. While most previous approaches consider known opponent models, we focus on the setting in which the opponent’s model is unknown. To this end, we use kernel-based regularity assumptions to capture and exploit the structure in the opponent’s response. We propose a novel algorithm for the learner when playing against an adversarial sequence of opponents. The algorithm combines ideas from bilevel optimization and online learning to effectively balance between exploration (learning about the opponent’s model) and exploitation (selecting highly rewarding actions for the learner). Our results include algorithm’s regret guarantees that depend on the regularity of the opponent’s response and scale sublinearly with the number of game rounds. Moreover, we specialize our approach to repeated Stackelberg games, and empirically demonstrate its effectiveness in a traffic routing and wildlife conservation task.

Keywords: Sequential Games, Online Learning, Gaussian Processes

1. Introduction

Several important real-world problems involve sequential interactions between two parties. These problems can often be modeled as two-player games, where the first player chooses a strategy and the second player responds to it. For example, in traffic networks, traffic operators plan routes for a subset of network vehicles (e.g., public transport), while the remaining vehicles (e.g., private cars) can choose their routes in response to that. The goal of the first player in these games is to find the optimal strategy (e.g., traffic operators seek the routing strategy that minimizes the overall network’s congestion, cf., Korilis et al. (1997)). Several algorithms have been previously proposed, successfully deployed, and used in domains such as urban roads (Jain et al., 2011b), airport security (Pita et al., 2009), wildlife protection (Yang et al., 2014), and markets (He et al., 2007), to name a few.

In many applications, complete knowledge of the game is not available, and thus, finding a good strategy for the first player becomes more challenging. The response function of the second player, that is, how the second player responds to strategies of the first player, is typically unknown and can only be inferred by repeatedly playing and observing the responses and game outcomes (Letchford et al., 2009; Blum et al., 2014). Consequently, we refer to the first and second players as learner and opponent, respectively. An additional challenge for the learner in such repeated games lies in facing a potentially different type of opponent at

* This work has been submitted to the Neural Information Processing Systems (NeurIPS) Conference, 2020
every game round. In various domains (e.g., in security applications), the learner can even face an adversarially chosen sequence of opponent/attacker types (Balcan et al., 2015).

Motivated by these important considerations, we study a repeated sequential game against an unknown opponent with multiple types. We propose a novel algorithm for the learner when facing an adversarially chosen sequence of types. No-regret guarantees of our algorithm in these settings ensure that the learner’s performance converges to the optimal one in hindsight (i.e., the idealized scenario in which the types’ sequence and opponent’s response function are known ahead of time). To that end, our algorithm learns the opponent’s response function online, and gradually improves the learner’s strategy throughout the game.

2. Problem Setup

We consider a sequential two-player repeated game between the learner and its opponent. The set of actions that are available to the learner and opponent in every round of the game are denoted by $\mathcal{X}$ and $\mathcal{Y}$, respectively. The learner seeks to maximize its reward function $r(x, y)$ that depends on actions played by both players, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. In every round of the game, the learner can face an opponent of different type $\theta_t \in \Theta$ that is unknown to the learner at the decision time. As the sequence of opponent’s types can be chosen adversarially, we focus on randomized strategies for the learner as explained below. We summarize the protocol of the repeated sequential game as follows.

In every game round $t$:
1. The learner computes a randomized strategy $p_t$, i.e., a probability distribution over $\mathcal{X}$, and samples action $x_t \sim p_t$.
2. The opponent observes $x_t$ and responds by selecting $y_t = b(x_t, \theta_t)$, where $b : \mathcal{X} \times \Theta \to \mathcal{Y}$ represents the opponent’s response function.
3. The learner observes the opponent’s type $\theta_t$ and response $y_t$, and receives reward $r(x_t, y_t)$.

The opponent’s types $\{\theta_i\}_{i=1}^T$ can be chosen by an adaptive adversary, i.e., at round $t$, the type $\theta_t$ can depend on the sequence of randomized strategies $\{p_i\}_{i=1}^t$ of the learner and on the previous realized actions $x_1, \ldots, x_{t-1}$ (but not on the current action $x_t$). The goal of the learner is to maximize the cumulative reward $\sum_{t=1}^T r(x_t, y_t)$ over $T$ rounds of the game. We assume that the learner knows its reward function $r(\cdot, \cdot)$, while the opponent’s response function $b(\cdot, \cdot)$ is unknown. To achieve this goal, the learner has to repeatedly play the game and learn about the opponent’s response function from the received feedback. After $T$ game rounds, the performance of the learner is measured via the cumulative regret:

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

The regret represents the difference between the cumulative reward of a single best action from $\mathcal{X}$ and the sum of the obtained rewards. An algorithm is no-regret if $R(T)/T \to 0$ as $T \to \infty$.

**Regularity assumptions.** Attaining sub-linear regret is not possible in general for arbitrary response functions and domains, and hence, this requires further regularity assumptions. We consider a finite set of actions $\mathcal{X} \subset \mathbb{R}^d$ available to the learner, and a finite set of opponent’s types $\Theta \subset \mathbb{R}^p$. We assume the unknown response function $b(x, \theta)$ is a member of a reproducing kernel Hilbert space $\mathcal{H}_k$ (RKHS), induced by some known positive-definite kernel function $k(x, \theta, x', \theta')$. RKHS $\mathcal{H}_k$ is a Hilbert space of (typically
non-linear) well-behaved functions $b(\cdot, \cdot)$ with inner product $\langle \cdot, \cdot \rangle_k$ and norm $\| \cdot \|_k = \langle \cdot, \cdot \rangle_k^{1/2}$, such that $b(x, \theta) = \langle b, k(\cdot, \cdot, x, \theta) \rangle_k$ for every $x \in \mathcal{X}, \theta \in \Theta$ and $b \in \mathcal{H}_k$. The RKHS norm measures smoothness of $b$ with respect to the kernel function $k$ (it holds $\|b\|_k < \infty$ iff $b \in \mathcal{H}_k$).

We assume a known bound $B > 0$ on the RKHS norm of the unknown response function, i.e., $\|b\|_k \leq B$. This assumption encodes the fact that similar opponent types and strategies of the learner lead to similar responses, where the similarity is measured by the known kernel function $k$.

Most popularly used kernel functions that we also consider are linear, squared-exponential (RBF) and Matérn kernels (Rasmussen, 2003).

Our second regularity assumption is regarding the learner’s reward function $r : \mathcal{X} \times \mathcal{Y} \rightarrow [0, 1]$, which we assume is $L_r$-Lipschitz continuous with respect to $\| \cdot \|_1$.

3. Proposed Approach

The observed opponent’s response can often contain some observational noise, e.g., in wildlife protection (see Section 4), we only get to observe an imprecise/inexact poaching location. Hence, instead of directly observing $b(x_t, \theta_t)$ at every round $t$, the learner receives a noisy response $y_t = b(x_t, \theta_t) + \epsilon_t$. For the sake of clarity, we consider the case of scalar responses, i.e., $y_t \in \mathbb{R}$, but in our full paper we also consider the case of vector-valued responses.

We let $\mathcal{H}_t = \{(x_t, \theta_t, y_t)\}_{t=1}^\infty$, and assume $\mathbb{E}[\epsilon_t | \mathcal{H}_t] = 0$ and $\epsilon_t$ is conditionally $\sigma$-sub-Gaussian, i.e., $\mathbb{E}[\exp(\zeta \epsilon_t) | \mathcal{H}_t] \leq \exp(\zeta^2 \sigma^2 / 2)$ for any $\zeta \in \mathbb{R}$.

At every round $t$, by using the previously collected data $\{(x_i, \theta_i, y_i)\}_{i=1}^t$, we can compute a mean estimate of the opponent’s response function via standard kernel ridge regression. This can be obtained in closed-form as:

$$
\mu_t(x, \theta) = k_t(x, \theta)^T (K_t + \lambda I_t)^{-1} y_t,
$$

where $y_t = [y_1, \ldots, y_t]^T$ is the vector of observations, $\lambda > 0$ is a regularization parameter, $k_t(x, \theta) = [k(x, \theta, x_1, \theta_1), \ldots, k(x, \theta, x_t, \theta_t)]^T$ and $[K_t]_{i,j} = k(x_i, \theta_i, x_j, \theta_j)$ is the kernel matrix.

The variance of the proposed estimator can be obtained as:

$$
\sigma^2_t(x, \theta) = k_t(x, \theta, x, \theta) - k_t(x, \theta)^T (K_t + \lambda I_t)^{-1} k_t(x, \theta).
$$

Moreover, we can use (2) and (3) to construct upper and lower confidence bound functions:

$$
\text{ucb}_t(x, \theta) := \mu_t(x, \theta) + \beta_t \sigma_t(x, \theta), \quad \text{lcb}_t(x, \theta) := \mu_t(x, \theta) - \beta_t \sigma_t(x, \theta),
$$

respectively, for every $x \in \mathcal{X}, \theta \in \Theta$, where $\beta_t$ is a confidence parameter. A standard result from Abbasi-Yadkori (2013); Srinivas et al. (2010) shows that under our regularity assumptions, $\beta_t$ can be set such that, with high probability, response $b(x, \theta) \in [\text{lcb}_t(x, \theta), \text{ucb}_t(x, \theta)]$ for every $(x, \theta) \in \mathcal{X} \times \Theta$ and $t \geq 1$.

Before moving to our main results, we define a sample complexity parameter that quantifies the maximum information gain about the unknown function from noisy observations:

$$
\gamma_t := \max_{\{(x_i, \theta_i)\}_{i=1}^t} 0.5 \log \det(I_t + K_t / \lambda).
$$

It has been introduced by Srinivas et al. (2010) and later on used in various theoretical works on Bayesian optimization. Analytical bounds that are sublinear in $t$ are known for popularly used kernels (Srinivas et al., 2010), e.g., when $\mathcal{X} \times \Theta \subset \mathbb{R}^d$, we have $\gamma_t \leq \mathcal{O}(\log(t)^{d+1})$ and $\gamma_t \leq \mathcal{O}(d \log(t))$ for squared exponential and linear kernels, respectively. This quantity characterizes the regret bounds obtained in the next sections.
The StackelUCB algorithm

Input: Finite action set $\mathcal{X} \subset \mathbb{R}^d$, kernel $k(\cdot, \cdot)$, param. $\lambda, \{\beta_t\}_{t \geq 1}, \eta$

1. Initialize: Uniform strategy $p_1 = \frac{1}{|\mathcal{X}|} \mathbf{1}_{|\mathcal{X}|}$
2. for $t = 1, 2, \ldots, T$
   3. Sample action $x_t \sim p_t$ // Opponent $\theta_t$ observes $x_t$ and computes $b(x_t, \theta_t)$
   4. Observe $\theta_t$ and noisy response $y_t = b(x_t, \theta_t) + \epsilon_t$
   5. Compute optimistic reward estimates:
      \[ \tilde{r}_t(x, \theta_t) := \max_y r(x, y), \quad \text{s.t.} \quad y \in [\text{lcb}_t(x, \theta_t), \text{ucb}_t(x, \theta_t)] \]
   6. Perform strategy update: $\forall x \in \mathcal{X}: p_{t+1}[x] \propto p_t[x] \cdot \exp(\eta \cdot \tilde{r}_t(x, \theta_t))$
   7. Update: $\mu_{t+1}, \sigma_{t+1}$ with $(x_t, \theta_t, y_t)$ (via (2), (3)), and $\text{lcb}_{t+1}, \text{ucb}_{t+1}$ (via (4))

3.1 The StackelUCB Algorithm

The considered problem (Section 2) can be seen as an instance of adversarial online learning (Cesa-Bianchi and Lugosi, 2006) in which an adversary chooses a reward function $r_t(\cdot)$ in every round $t$, while the learner (without knowing the reward function) selects action $x_t$ and subsequently receives reward $r_t(x_t)$. To achieve no-regret, the learner needs to maintain a probability distribution $p_t$ over the set $\mathcal{X}$ of available actions and play randomly according to it. Multiplicative Weights (MW) (Littlestone and Warmuth, 1994) algorithms such as EXP3 (Auer et al., 2003) and HEDGE (Freund and Schapire, 1997) are popular no-regret methods for updating $p_t$, depending on the feedback available to the learner in every round. The former only needs observing reward of the played action $r_t(x_t)$ (bandit feedback), while the latter requires access to the entire reward function $r_t(\cdot)$ at every $t$ (full-information feedback).

The considered game setup corresponds (from the learner’s perspective) to the particular online learning problem in which $r_t(\cdot) := r(\cdot, b(\cdot, \theta_t))$, type $\theta_t$ is revealed, and the bandit observation $y_t$ is observed by the learner. Full-information feedback, however, is not available as $b(\cdot, \theta_t)$ is unknown. To alleviate this, similarly to Sessa et al. (2019), we compute "optimistic" reward estimates to emulate the full-information feedback. Based on previously observed data, we establish upper and lower confidence bounds $\text{ucb}_t(\cdot)$ and $\text{lcb}_t(\cdot)$, of the opponent’s response function via (4). These are then used to estimate the optimistic rewards of the learner for any $x \in \mathcal{X}$ at round $t$ as:

\[ \tilde{r}_t(x, \theta_t) := \max_y r(x, y) \quad \text{s.t.} \quad y \in [\text{lcb}_t(x, \theta_t), \text{ucb}_t(x, \theta_t)] \]  

Optimistic rewards allow the learner to control the maximum incurred regret, while Lipschitzness of $r(\cdot)$ ensures that learning the opponent’s response function (via (2) and (3)) translates to more accurate reward estimates. The proposed approach is summarized in our novel StackelUCB algorithm (see Algorithm 1) which provides the following guarantee.

**Theorem 1** For any $\delta \in (0, 1)$, the regret of StackelUCB when used with $\lambda \geq 1$, $\beta_t = \sigma \lambda^{-1} \sqrt{2 \log \left( \frac{1}{\delta} \right) + \log(\det(I_t + K_t/\lambda))} + \lambda^{-1/2}B$, and learning step $\eta = \sqrt{8 \log(|\mathcal{X}|)} / T$, is bounded, with probability at least $1 - 2\delta$, by

\[ R(T) \leq \sqrt{\frac{1}{2} T \log |\mathcal{X}|} + \sqrt{\frac{1}{2} T \log \left( \frac{1}{\delta} \right)} + 4 L\rho \beta_T \sqrt{T\lambda_T}, \]

where $B \geq \|b\|_{\mathcal{H}_k}$ and $\gamma_T$ is the maximum information gain defined in (5).

The obtained regret bound scales sublinearly with $T$, and depends on the regret obtained from playing HEDGE (first two terms) and learning of the opponent’s response function (last
Figure 1: **Left:** Time-averaged regret of the operator using different routing strategies. **StackelUCB** (polynomial kernels of degree 3 or 4) leads to a smaller regret compared to the considered baselines and performs comparably to the idealized **Hedge** algorithm. **Right:** Edges’ congestion (color intensity proportional to the time-averaged congestion) when the operator at each round: (left) Routes 100% of the units via the shortest route, (middle) Routes 0% of units, and (right) Uses **StackelUCB**.

term in the regret bound). We note that **EXP3** attains $O\left(\sqrt{T\log |X|}\right)$ while **HEDGE** attains improved $O\left(\sqrt{T\log |X|}\right)$ regret bound which scales favourably with the number of available actions $|X|$. The same holds for our algorithm, but crucially – unlike **HEDGE** – our algorithm uses the bandit feedback only.

### 3.2 Single Opponent Type

In case the learner is playing against an opponent of a single type $\bar{\theta}$, in our full paper we show that a simpler version of **StackelUCB** achieves an improved regret bound of $4L_r\beta_\gamma T\sqrt{T\lambda_\gamma \gamma}$. The strategy consists, at each time $t$, of selecting $x_t = \arg\max_{x \in X} \tilde{r}_t(x, \bar{\theta})$ and is reminiscent of the **GP-UCB** algorithm used in standard Bayesian optimization (Srinivas et al., 2010).

### 3.3 Learning in Repeated Stackelberg Games

Repeated Stackelberg games (von Stackelberg, 1934) are sequential games between a leader (learner) and a follower (opponent). They can be mapped to our setup of Section 2 by letting $X = \Delta_{n_l}$ be the leader decision set, where $\Delta_{n_l}$ stands for $n_l$-dimensional simplex. Moreover, the opponent’s response function in a Stackelberg game assumes the specific best-response form $b(x_t, \theta_t) = \arg\max_{y \in Y} U_{\theta_t}(x_t, y)$, where $U_{\theta_t}(x, y)$ represents the expected utility of the follower of type $\theta_t$ under the leader’s strategy $x$. Balcan et al. (2015) shows that a regret bound of $O\left(\sqrt{T \cdot \text{poly}(n_l, n_f, k_f)}\right)$ ($n_f$ and $k_f$ are the numbers of actions available to the follower and possible follower types, respectively) can be achieved assuming a finite set of followers with known utilities. In this work, we show that **StackelUCB** leads to a regret of $O\left(\sqrt{T n_l \log \left(L_r L_\delta \sqrt{n_l T}\right)} + L_r \beta_\gamma T \sqrt{T\lambda_\gamma \gamma}\right)$ in the more challenging setting where such utilities are unknown (also, potentially with an infinite number of types).

### 4. Experiments

We evaluate the proposed algorithms in traffic routing and wildlife conservation tasks.

**Routing Vehicles in Congested Traffic Networks.** We consider a traffic routing task in the network of Sioux-Falls (LeBlanc et al., 1975), in which the goal of the network operator is to route 300 units (e.g., a fleet of autonomous vehicles) between the two nodes of the network (depicted as blue and green nodes in Figure 1). At the same time, the goal of the operator is to avoid the network becoming overly congested. We model this problem as a repeated sequential game (as defined in Section 2) between the network operator (learner)
Wildlife Protection against Poaching Activity. We consider a wildlife conservation task where the goal of park rangers is to protect animals from poaching activities. We model this problem as a sequential game between the rangers, who commit to a patrol strategy $x$ (i.e., covering each park area with some probability), and the poachers that observe the rangers’ strategy to decide upon a poaching location $y = b(x)$. We use the game model of Kar et al. (2015) to define poachers’ model $b(\cdot)$ and rangers’ reward function $r(\cdot)$. We study the repeated version of this game in which the rangers start with no information about the poachers’ model and use the algorithm discussed in Section 3.2 to discover the best patrol strategy online.

In Figure 2 (left plot), we compare the performance of our algorithm with the ones achieved by: 1) Optimal strategy (OPT) $x^* = \arg \max_{x \in X} r(x, b(x))$ with known poachers’ model, 2) Max-Min, i.e., $x_m = \arg \max_{x \in X} \min_y r(x, y)$, which assumes the worst possible poaching location, and 3) Best-offline, that is, $x_o = \arg \max_{x \in X} r(x, \mu_o(x))$, where $\mu_o(\cdot)$ is the mean estimate of $b(\cdot)$ computed offline as in (2) by using 1’000 random data points. Our algorithm outperforms the considered baselines and discovers the optimal patrol strategy after $\sim 60$ rounds. We depict the discovered strategy in Figure 2 (rightmost plot). We observe that the cells covered with higher probabilities are the ones with a high animal density near to the poachers’ starting location (despite the latter is unknown to the algorithm).
References


