Grinding the Space: Learning to Classify Against Strategic Agents

Yiling Chen  
Harvard University  
yiling@seas.harvard.edu

Yang Liu  
UC Santa Cruz  
yangliu@ucsc.edu

Chara Podimata  
Harvard University  
podimata@g.harvard.edu

June 16, 2019

Abstract

We address the problem of online learning in strategic classification settings from the perspective of the learner, who is repeatedly facing bounded, strategic agents. We model this interplay as a repeated Stackelberg game, where at each timestep the learner deploys a high-dimensional linear classifier first and an agent, after observing the classifier, along with his real feature vector, and according to his bounded utility function, best-responds with a (potentially altered) feature vector. We measure the loss of the learner in terms of Stackelberg regret for her $0-1$ loss function. We prove that in such strategic settings there exist worst-case scenarios, where any sequence of actions providing sublinear external regret might result in linear Stackelberg regret and vice versa. We then provide the Grinding Algorithm, an adaptive discretization algorithm, potentially of independent interest, and prove its data-dependent upper bound on the Stackelberg regret given oracle access, while being computationally efficient. We complement our theoretical analysis with simulation results, which suggest that our algorithm outperforms the benchmarks, even given approximate oracle access.

1 Introduction

As Machine Learning (ML) algorithms are getting more and more involved in real-life decision making, the agents that we normally face are neither stochastic, nor adversarial. Rather, they seem to be strategic. Think about a college that wishes to deploy an ML algorithm to make the admissions decisions. Student candidates might try to manipulate their test scores in order to try to fool the classifier. Importantly, however, these student candidates do want to be admitted; they do not simply want to sabotage the admissions algorithm. And this is precisely what differentiates them from being fully adversarial. Similar situations arise in almost every deployment of ML algorithms in real-life settings. As such, strategic agents present a unique threat, but also a unique opportunity for ML researchers.

Compared to classical adversarial models in ML, strategic models have a unique advantage; namely, if the incentives of the agents are aligned properly (i.e., through the use of payments, or through the use of specifically designed mechanisms that are robust to strategic noise), then it is possible for the learner to obtain a virtually clean dataset. In the language of game theory, such mechanisms render the respective ML tasks that they are applied to strategyproof, and there has been an increasing interest in these types of mechanisms, especially for the tasks of classification, regression, and strategyproofness remains a very hard desideratum to achieve for dynamical, real-life settings, like the one which is the focus of this work: online strategic classification. The more relaxed solution concept that has emerged as the appropriate for these settings is the one of Stackelberg regret, where a learner compares her cumulative loss with the cumulative loss of her best-fixed action in hindsight, had she given the agent the opportunity to best-respond to it.

1 And, indeed, there is ample evidence they would do so.
1.1 Our Results and Techniques

While previous works have assumed more rigid structural properties both for the agents’ utility functions and the learner’s loss function, in this work we focus on a rather large and abstract family of agents’ utility functions, which can be expressed as $\text{value}(\text{misreport}) - \text{cost}(\text{misreport})$. We observe that these utility functions all satisfy a nice property, which we term $\delta$-boundedness. For the learner’s loss, we focus on the $0-1$ loss function. We first prove a strong worst-case incompatibility result between Stackelberg and external regret as well as with their closely related notion of strategic regret. This incompatibility implies that no standard external regret algorithm could be blindly applied and guarantee sublinear Stackelberg regret from the learner’s perspective. The latter, coupled with the fact that as we prove, in this more abstract utility model that we consider, the learner’s loss as a function of her action $\alpha$ is not even Lipschitz, commands for a new algorithm for minimizing Stackelberg regret when the learner’s action set is the continuous interval $[-1, 1]^d$. Previous approaches have dealt with this problem by proving conditions under which the learner’s loss function becomes either linear or convex. Drawing intuition from Kleinberg et al. [22]’s seminal work on the Zooming algorithm, we propose our Grinding Algorithm, an adaptive discretization algorithm for learning in strategic settings, which might be of independent interest in the general online learning literature. Roughly, our Grinding Algorithm partitions appropriately the learner’s action space, $A$, into polytopes. We prove, using an involved analysis of a continuous variant of multiplicative weights update, the following data dependent bound on the Stackelberg regret of the learner, provided that she has access to an in-oracle, which can be queried to return the total in-probability (defined by Alon et al. [1]) of an action: $\mathcal{R}(T) \leq O\left(\sqrt{\max_{t \in [T]} [8 \log(4|\mathcal{A}|) + \lambda(P_{t,GT})] \cdot \log(\frac{|\mathcal{A}|}{T}) \cdot T}\right)$, where by $\lambda(A)$ we denote the Lebesgue measure of a measurable space $A$, by $p$ the polytope with the smallest Lebesgue measure after $T$ timesteps, and by $P_{t,GT}$ a subset of $A$ which depends on the $\delta$-boundedness of the agent’s utility function. Finally, while the requirement of an in-oracle might seem like a hard constraint to satisfy, we show in simulations both on a discrete, predefined set of actions, and a fully continuous one, that this oracle can be approximated by a logistic regression oracle and the resulting algorithm performs much better than the EXP3 Stackelberg regret.

1.2 Related Work

Our work is primarily situated at the interface of three research areas: learning using data from strategic sources, multi-armed bandits, and Stackelberg games. Recently, there has been an increasing body of work studying machine learning algorithms that are trained on data controlled by strategic agents. In this context, the two most studied machine learning tasks for which such algorithms are provided are linear regression [28, 18, 17, 14, 7, 15, 8], and classification. Focusing on the task of classification, most of the existing works are written from the perspective of an offline learner [21, 26, 27], with the exception of [19]. Contrary to our setting, in the models used by Meir et al. [20, 27] the feature vectors of the agents are assumed to be publicly verifiable, while in [21, 19] the feature vectors are the only manipulable piece of information that the agents possess. Our work is most closely related to the work of Dong et al. [19], who find the appropriate conditions guaranteeing that the best-response of the agent, written as a function of the learner’s committed action, is concave. As a result, the learner’s loss function becomes convex, and well-known online convex optimization algorithms could be applied (e.g., [20, 13]), in conjunction with their novel consideration of mixture feedback that the learner receives. In our work, however, we consider less structured utility and loss functions from the perspective of the agent and the learner respectively.

Online learning through multi-armed bandits in both stochastic and adversarial environments with partial feedback has received great attention in the ML literature (see [12, 31, 24] for excellent overviews of the field). Contrary to the above, our strategic classification setting presents a unique form of feedback, which depends on the best-response of the agents to the learner’s committed actions. Connecting learning from strategic sources and online learning for multi-armed bandits, Braverman et al. [10] studied a model where each arm

\footnote{Apart from the intuitive idea of partitioning the action space further, when one thinks it is required, the two algorithms have no other properties in common.}
is controlled by a strategic entity, able to choose the amount of the reward that it will pass to the learner. In our model, the arms are not explicitly controlled by anybody.

Finally, our work is related to the Stackelberg games literature, which has mostly dealt with Stackelberg Security Games (SSGs) (see [23] for an overview), for which there have been a number of recent works related to learning theoretic problems [24, 25, 9, 5]. SSGs impose extra structure both on the agents utility, and on the learner’s loss, which is not present in the setting of strategic classification that we consider.

2 Model

Formally, we model the problem of online learning for classification as the following repeated Stackelberg game:

1. At each timestep $t \in [T]$, the learner draws an action $\alpha_t \sim A \subseteq [-1, 1]^{d+1}$.
2. An agent observes $\alpha_t$ and draws $(x_i, y_t)$, where $x_i \sim X \subseteq ([0, 1]^d, 1)$ and $y_t = \{-1, +1\}$.
3. The agent chooses to report a feature vector $z_t(\alpha_i; x_i)$ which may be different from $x_i$.
4. The learner observes the true label of $z_t(\alpha_i; x_i)$, denoted $\hat{y}_t$, and incurs classification loss $\ell(\alpha_i, z_t(\alpha_i; x_i)) = \mathbb{I}\{\hat{y}_t \cdot (\alpha_i, z_t(\alpha_i; x_i)) \leq -1\}$.

The agent’s real feature vector $x_i$ can be drawn adversarially from $X$, and can never be observed with certainty by the learner. Motivated by the spam emails setting, we assume that agents derive no value if, by manipulating their feature vector from $x_i$ to $z_t(\alpha_i; x_i)$, they change the corresponding true label, that is $y_t \neq \hat{y}_t$. Intuitively, this means that a spammer may want to modify a spamming email to pass the spam filter as long as the email is still a spam, but a spammer has no value if the modification is so dramatic that it essentially changes the email to a legitimate non-spam email. Similarly, a non-spammer derives no value if he changes his email to a spam email. We use $v_t(\alpha_i; z_t(\alpha_i; x_i)) \in [0, 1]$ to denote an agent’s value of reporting $z_t(\alpha_i; x_i)$ when his true feature vector is $x_i$. An example of the valuation function is $v_t(\alpha_i, z_t(\alpha_i; x_i)) = \delta I\{\langle (\alpha_i, z_t(\alpha_i; x_i)) \rangle \geq 0\} I\{\hat{y}_t = y_t\}$, which equals 0 if $\hat{y}_t \neq y_t$ and equals $\delta$ if the agent passes the classifier without making $\hat{y}_t$ to be different from $y_t$. An agent’s utility of reporting $z_t(\alpha_i; x_i)$ is the difference between his value $v_t(\alpha, z_t(\alpha_i; x_i))$ and the cost of manipulation $c_t(\alpha, z_t(\alpha_i; x_i))$. For example, $u_t(\alpha_i, z_t(\alpha_i; x_i)) = \delta \cdot I\{\langle (\alpha_i, z_t(\alpha_i; x_i)) \rangle \geq 0\} I\{\hat{y}_t \neq y_t\} - (x_i - z_t(\alpha_i; x_i))^2$, with $c_t(\alpha, z_t(\alpha_i; x_i)) = (x_i - z_t(\alpha_i; x_i))^2$. The learner does not know the specific value and cost functions that agents have but knows the general form of the utility function (i.e., value - cost).

We assume that agents are myopically rational. That is, they act to maximize their utility in the current round (hence, rational) without trying to mislead the future choices of classifiers (hence, myopic). This means that the agent will report $r_t(\alpha_i; x_i) = \arg\max_{x_i \in X \subseteq [0, 1]^d} u_t(\alpha_i, z_t(\alpha_i; x_i))$ in step 3 of the Stackelberg game. Because the utility function is of the form $u_t(\alpha, z_t(\alpha; x_i)) = v_t(\alpha, z_t(\alpha; x_i)) - c_t(\alpha, z_t(\alpha; x_i))$, where $v_t(\cdot) \in [0, 1]$, and the agents are individually rational, they satisfy a nice $\delta$-boundedness property, for some appropriately tuned $\delta$: when best-responding in step 3, an agent’s report $r_t(\alpha_i; x_i)$ lies in a ball of known radius $\delta$ around their $x_i$, denoted by $B_\delta(x_i)$. We note that the previously considered utility functions in Dong et al. [19] can all be captured by our model. Moreover, in our model, the value function $v_t(\cdot)$ can even be $0 - 1$ and the cost function $c_t(\cdot)$ need not be convex, which address an open question of Dong et al. [19]. We finally remark that because agents derive no value when $\hat{y}_t = y_t$, by observing $\hat{y}_t$, one essentially observes $y_t$. We hence drop the notation $\hat{y}_t$ and only use $y_t$ for the rest of the paper.

---

3 We note here that $A$ corresponds to the $d + 1$ normal vectors that translate into $d$-dimensional hyperplanes.

4 Our results hold even without this assumption, if the learner can observe the true label $y_t$ for $x_t$ in each round, similar to the setting of Dong et al. [19].

5 When clear from context, we will denote the agent’s best-response feature vector to action $\alpha$, when he has true, hidden feature vector $x_t$, as $r_t(\alpha)$, in order to simplify notation.
3 Regret Notions

In this section, we outline the three predominant notions of regret used to evaluate online learning in strategic settings (i.e., external, Stackelberg, and strategic regret). We will present a somewhat surprising result; namely, that despite intuitively thinking that there exists a clear hierarchy between the three notions, there are settings where any two of these are strongly incompatible. For completeness, we first include the definitions of the regret notions in the context of repeated Stackelberg games, and then we proceed with the statement of our results. The proofs of this section can be found in the Supplementary Material. For what follows, let \( \{\alpha_t\}_{t=1}^{T} \) be the sequence of actions chosen by the learner in a repeated Stackelberg game and \( \mathcal{A} \) the allowable action set.

**Definition 3.1** (External Regret). \( R(T) = \sum_{t=1}^{T} \ell(\alpha_t, r_t(\alpha_t)) - \min_{\alpha^*_t \in \mathcal{A}} \sum_{t=1}^{T} \ell(\alpha^*_t, r_t(\alpha_t)). \)

**Definition 3.2** (Stackelberg Regret). \( R(T) = \sum_{t=1}^{T} \ell(\alpha_t, r_t(\alpha_t)) - \min_{\alpha^*_t \in \mathcal{A}} \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha_t)). \)

**Definition 3.3** (Strategic Regret). Let \( r^*_t \) denote the truthful response of the agent at timestep \( t \). Strategic regret is defined as: \( R(T) = \sum_{t=1}^{T} \ell(\alpha_t, r_t(\alpha_t)) - \min_{\alpha^*_t \in \mathcal{A}} \sum_{t=1}^{T} \ell(\alpha^*_t, r^*_t) \).

**Theorem 3.1.** There exists a repeated Stackelberg game between a learner and an agent, such that every action sequence with sublinear external regret incurs linear Stackelberg regret, and every action sequence with sublinear Stackelberg regret incurs linear external regret.

At a high level, the proof of this Theorem constructs a particular Stackelberg repeated game with three actions for the learner, where two of the actions are the best fixed ones for the Stackelberg regret, while they yield linear external regret, and the third action is the best fixed one for the external regret but yields linear Stackelberg regret. Using similar techniques, one can prove the following two incompatibility Theorems.

**Theorem 3.2.** There exists a repeated Stackelberg game between a learner and an agent, such that every action sequence with sublinear strategic regret incurs linear Stackelberg regret, and every action sequence with sublinear Stackelberg regret incurs linear strategic regret.

**Theorem 3.3.** There exists a repeated game between a learner and an agent, such that every action sequence with sublinear external regret incurs linear strategic regret, and every action sequence with sublinear strategic regret incurs linear external regret.

Despite these worst-case incompatibility results, we show in the Supplementary Material, that there are some families of repeated Stackelberg games, which we call Pure Stackelberg Games, where there does exist a hierarchy between these regret notions. The challenge, however, rests on the fact that most meaningful repeated Stackelberg games (e.g., strategic classification, Stackelberg Security Games) lie in the space between the worst-case incompatibility instances and Pure Stackelberg Games.

4 The Grinding Algorithm

In this section, we will present and analyze our Grinding Algorithm. For the ease of exposition, we begin with a thought experiment; if the learner was given a priori a well-defined action set, then, combining known tools from the literature in Online Learning with Feedback Graphs, we observe that the learner

---

\(^6\) We refer the interested reader to the Supplementary Material for an extensive discussion on the connections with the related notion of policy regret.

\(^7\) I.e., an action set that was guaranteed to include at least one action which performed on average approximately as well as the best-fixed action in hindsight, had you given the agent the chance to best-respond to it at each timestep.
could at each timestep build a strongly observable feedback graph. Ideally, the learner would want to place in \(N^{\text{out}}(\alpha)\) the actions that are outside of \(B_{2\delta}(x_t)\), but in the absence of knowledge of the agent’s true \(x_t\), updating actions outside of \(B_{2\delta}(r_t(\alpha))\) is a conservative estimate of the \(N^{\text{out}}(\alpha)\). The reason why the feedback graph can be constructed this way and we do not encounter problems similar to the ones mentioned in [16] is the fact that \(\delta\) is assumed to be known in advance. Most often than not, such a well-defined action set is not provided to the learner. Further, the function \(\ell(\alpha, r_t(\alpha))\) is not even Lipschitz, and no black-box external regret minimizing algorithm could be safely applied. Our Grinding Algorithm solves these problems by adaptively discretizing the learner’s action space, according to the agent’s responses.

**Grinding Algorithm Overview.** At a high level, the Grinding Algorithm maintains a sequence of nested polytopes \(P_t\) of the form \([x, y]^{d+1}\) for \(-1 \leq x \leq y \leq 1\). By convention, we will call a single point a point-polytope, and we will denote the set of all polytopes as \(\mathcal{P}\). We start by setting \(P_1 = [-1, 1]^{d+1}\). At round \(t\), the Grinding Algorithm chooses an action for the learner randomly as follows. First, the learner draws a polytope \(p \in P_t\) with probability \(\pi_t(p)\) and then, she draws an action \(\alpha_t \sim \text{Unif}(p)\) uniformly over the polytope \(p\). We denote the resulting distribution by \(\mathcal{A}_t\), and by \(\Pr_{\mathcal{A}_t}\) and \(f_{\mathcal{A}_t}(\alpha)\) the associated probability and probability density function, respectively.

After the learner observes \(r_t(\alpha_t)\), the refinement of her action space is done as follows: First, she computes two hyperplanes: \(\beta^U_t(\alpha_t)\) and \(\beta^L_t(\alpha_t)\), such that: \((\beta^U_t(\alpha_t), r_t(\alpha_t)) = 4\sqrt{d}\delta\), and \((\beta^L_t(\alpha_t), r_t(\alpha_t)) = -4\sqrt{d}\delta\)\(^8\). These two hyperplanes define the new discretization polytope boundaries for our action space by creating three regions: one for which \(\forall w: (w, r_t(\alpha_t)) \geq 4\sqrt{d}\delta\), one for which \(\forall w: (w, r_t(\alpha_t)) \leq -4\sqrt{d}\delta\) and one “in the middle”. Figure 1 shows a depiction of the initial discretization of \([-1, 1]^2\). Formally, these regions are defined as follows.

**Definition 4.1 (Boundary Hyperplanes).** We define the upper and the lower boundary hyperplanes of action \(\alpha \in \mathcal{A}\), denoted as \(\beta^U_t(\alpha) \in [-1, 1]^{d+1}\) and \(\beta^L_t(\alpha) \in [-1, 1]^{d+1}\), the hyperplanes such that \((\beta^U_t(\alpha), r_t(\alpha)) = 4\sqrt{d}\delta\) and \((\beta^L_t(\alpha), r_t(\alpha)) = -4\sqrt{d}\delta\) respectively.

Let \(H^+(\beta), H^-(\beta)\) denote the closed positive and negative halfspaces defined by hyperplane \(\beta\).

**Definition 4.2 (Upper & Lower Polytopes Set).** We define the set of polytopes that belong in the closed positive halfspace defined by hyperplane \(\beta^+_{t}(\alpha)\), denoted as \(P^+_t(\alpha) = \{p \in P_1: p \subseteq H^+(\beta^+_{t}(\alpha))\}\), as the action \(\alpha\)’s upper polytopes set. Similarly, the set of polytopes that belong in the closed negative halfspace defined by hyperplane \(\beta^-_{t}(\alpha)\), denoted as \(P^-_t(\alpha) = \{p \in P_1: p \subseteq H^-(\beta^-_{t}(\alpha))\}\), will be called actions \(\alpha\)’s lower polytopes set.

\(^8\) This is crucial in the worst-case, as any \(B_{\varepsilon}(x'_t)\) for \(\varepsilon < 2\delta\) and \(x'_t \neq r_t(\alpha)\) might create constant bias, thus resulting in linear Stackelberg regret.

\(^9\) Since actions are not distinct, but are instead drawn from a continuous probability distribution, we will use the probability density function in order to construct the unbiased estimates of the actions belonging to \(B_{2\delta}(r_t(\alpha_t))\).

\(^10\) The \(2\sqrt{d}\) factor accounts for the fact that \(\beta^U_t(\alpha_t), \beta^L_t(\alpha_t)\) must be at distance \(2\delta\) from the point \(r_t(\alpha_t)\).
The usefulness of sets $P_t^i(\alpha_i)$ and $\hat{P}_t^i(\alpha_i)$ is that for all the actions contained in them, due to the agent’s $\delta$-boundedness, since $\alpha \in P_t^i(\alpha_i) \cup \hat{P}_t^i(\alpha_i)$, the loss of the learner boils down to: $\ell(\alpha, r_t(\alpha)) = 1 \{y_t = -1\} \{\alpha \in P_t^i(\alpha_i)\} + 1 \{y_t = 1\} \{\alpha \in \hat{P}_t^i(\alpha_i)\}$, without requiring direct observation of $x_t$. Despite the fact that our algorithm does not require the computation of $P_t^i(\alpha), \hat{P}_t^i(\alpha), \forall \alpha \neq \alpha_t \in \mathcal{A}$, we do require access to what we call an In-Oracle, defined formally below. While access to such an omnipotent oracle might seem far-fetched, we provide simulations in Section 5, where we have relaxed this requirement to an easily-computable approximate in-oracle, with positive results.

**Definition 4.3 (In-Oracle).** We define the In-Oracle as a black-box algorithm, which gets as input an action (resp. a polytope) and returns the total in-probability for this action (resp. polytope):

$$\hat{P}_t^i[\alpha] = \int_{A_t^i} \Pr[\{\alpha \in H^+(\beta^i_t(\alpha'))\} \cup \{\alpha \in H^-(\beta^i_t(\alpha'))\} \cup \{\alpha' = \alpha\}] \, d\alpha'$$

and

$$\hat{P}_t^i[p] = \int_{A_t} \Pr[\{p \subseteq H^+(\beta^i_t(\alpha'))\} \cup \{p \subseteq H^-(\beta^i_t(\alpha'))\}] \, d\alpha'.$$

We are now ready to formally define our Grinding Algorithm, where we denote by $\lambda(A)$ the Lebesgue measure of the measurable space $A$. In what follows, we refer to $dq_t(\alpha)$ as the probability density function at point $\alpha$, and to $q_t(B)$ as the cumulative distribution function for a polytope $B$.

**Algorithm 1** Grinding Algorithm for Strategic Classification

1. Let $dq_1(\alpha) = \frac{1}{\lambda(A)}, \forall \alpha \in \mathcal{A}, \mathcal{P} = [-1, 1]^{m+1}, w_1(p) = \lambda(p), p \in \mathcal{P}$ and $\eta, \gamma$ to be specified in the analysis.
2. for $t = 1, \ldots, T$ do
3. Compute $\forall p \in \mathcal{P}_t : \pi_t(p) = (1 - \gamma)q_t(p) + \gamma \frac{\lambda(p)}{\lambda(A)}$.
4. Select polytope $p_t \sim \pi_t$ and then, draw an action $\alpha_t \sim \text{Unif}(p_t)$.
5. Commit to action $\alpha_t$.
6. Observe the attacker’s $r_t(\alpha_t)$ and $y_t$ (i.e., the true label for $r_t(\alpha_t)$).
7. Define set of new polytopes $\mathcal{P}_t = \mathcal{P}_t^i(\alpha_t) \cup \hat{P}_t^i(\alpha_t) \cup \hat{P}_t^i(\alpha_t)$, where:
   - $\mathcal{P}_t^i(\alpha_t) = \{p' | p' \neq 0, p' \in \mathcal{P}_t \}$
   - $\hat{P}_t^i(\alpha_t) = \{p' | p' \neq 0, p' \in \mathcal{P}_t \}$
   - $\hat{P}_t^i(\alpha_t) = \{p' | p' \neq 0, p' \in \mathcal{P}_t \}$
8. Compute $\hat{\ell}(\alpha_t, r_t(\alpha_t)) = \frac{\ell(\alpha_t, r_t(\alpha_t))}{\hat{P}_t^i(\alpha_t)}$.
9. for $p \in \mathcal{P}_t$ do
10. if $p \subseteq H^+(\beta^i_t(\alpha_t))$ or $p \subseteq H^-(\beta^i_t(\alpha_t))$ then
11. Compute $\hat{\ell}(p, r_t(p)) = \frac{\ell(p, r_t(p))}{\hat{P}_t^i(p)}$.
12. end if
13. end for
14. $w_{t+1}(p) = \lambda(p) \exp(-\eta \sum_{t=1}^{T} \hat{\ell}(p, r_t(p)))$, $q_{t+1}(p) = \frac{w_{t+1}(p)}{\sum_{p' \in \mathcal{P}_{t+1}} w_{t+1}(p')}$.
15. end for

The polytopes are defined in such a way that at each timestep the estimated loss within each polytope is constant. Not only that, but also, if a polytope has not been further “grounded” by the algorithm, then the estimated loss that was used to update the polytope has been the same within the actions of the polytope for each timestep. This observation explains the way the weights of the polytopes are updated. Essentially, the learner updates the $dq_t(\alpha)$ and $dw_t(\alpha)$ of all the actions in the space $\mathcal{A}$ as follows $dw_{t+1}(\alpha) = \exp(-\eta \hat{\ell}(\alpha, r_t(\alpha)))dw_t(\alpha)$, and $dq_{t+1}(\alpha) = \frac{dw_{t+1}(\alpha)}{\int_{\alpha' \in \mathcal{A}} \exp(-\eta \hat{\ell}(\alpha', r_t(\alpha')))dw_t(\alpha')}$.
Theorem 4.1. The Stackelberg Regret induced by the Grinding Algorithm using the unbiased estimator \( \hat{\ell}(\alpha, r_i(\alpha)) \), and both when the horizon \( T \) is known and unknown is:

\[
\mathcal{R}(T) \leq O\left( \sqrt{\max_{t \in [T]} \left\{ 4 \log \left( \frac{\lambda(A)}{\lambda(p)} \right) \left| \mathcal{P}_{t,GT}^u \cup \mathcal{P}_{t,GT}^l \right| T \right\} + \lambda \left( \frac{\mathcal{P}_{m,t,GT}^u}{\mathcal{P}_{m,t,GT}^l} \right) \cdot \log \left( \frac{\lambda(A)}{\lambda(p)} \right), T \right) \right)
\]

We defer the proof to the Supplementary Material and below we include a sketch.

Proof Sketch. We first show that the loss estimator \( \hat{\ell}(\alpha, r_i(\alpha)) \) is an unbiased estimator of the true loss for each action \( \ell(\alpha, r_i(\alpha)) \). Subsequently, we show that its second moment for an action \( \alpha \) is upper bounded by the term \( 1/\Pr^{in}[\alpha] \). For the purposes of the analysis, we then define three families of polytope sets as follows. Polytopes including actions that are in signed distance more than \( 2\delta \) away from \( x_t \), belong in the upper ground truth polytope set, \( \mathcal{P}_{t,GT}^u \). Polytopes including actions that are in signed distance less than \( -2\delta \) from \( x_t \), belong in the lower ground truth polytope set, \( \mathcal{P}_{t,GT}^l \). The rest of the polytopes belong in the middle ground truth polytope set, \( \mathcal{P}_{t,GT}^m \). Actions that belong in \( \mathcal{P}_{t,GT}^m \) are actions that potentially the agent could misreport and fool the learner, so each of these actions can safely update only itself. This creates the dependency of our regret to the Lebesgue measure of \( \mathcal{P}_{t,GT}^m \). The next argument makes a novel connection with a graph theoretic lemma, used by the literature in online learning with feedback graphs. Observe that each of the actions belonging in \( \mathcal{P}_{t,GT}^u, \mathcal{P}_{t,GT}^l \) gets updated with probability 1 by any other action in the set \( \mathcal{P}_{t,GT}^u \). This is because for any of the actions in \( \mathcal{P}_{t,GT}^u \), the agent could not have possibly misreported. So, for all actions \( \alpha \in \mathcal{P}_{t,GT}^u \cup \mathcal{P}_{t,GT}^l \) we have that: \( \Pr^{in}[\alpha] \geq \sum_{\mathcal{P}_{t,GT}^u \cup \mathcal{P}_{t,GT}^l \pi_t(p)} \). As a result, we can instead think about the set of polytopes that belong in \( \mathcal{P}_{t,GT}^u \) as forming a fully connected feedback graph. The latter, coupled with the fact that our exploration term makes sure that each polytope \( p \) is chosen with probability at least \( \lambda(p)/\lambda(A) \) gives us the result. We note here that an effort of a straightforward application of the graph theoretic lemma on the action set, rather than the polytopes’ one, gives vacuous regret upper bounds, due to the logarithmic dependence in the number of nodes of the feedback graph.

5 Simulations

In this section, we present our experimental results. Since all real datasets for classification that are currently available are not collected taking the strategic considerations of the agents into account, they cannot be used for evaluating our algorithm. Indeed, one cannot be certain whether the currently reported feature vectors stem from an altered original feature vector or not. In order to evaluate empirically the Grinding algorithm’s performance against other algorithms, one needs to know \( x_t \). In the Supplementary Material, we include an extended discussion on the simulations, and we provide the plots for a different approximation oracle than the one that we have below. 

\[\text{Since they are outside of his } \delta \text{-bounded region.}\]
Setup. We fix the dimension $d = 2$, but the code can be generalized to higher dimensions as well. To make our model more realistic, and drawing intuition from Dong et al. [10] we assume that we do not just face strategic agents (called spammers with $y_t = -1, x_t \sim \text{Unif}[0.0, 0.6]^2$), but also, with probability $p$, agents that do not wish to try and fool us (mixture feedback) (called non-spammers with $y_t = 1, x_t \sim \text{Unif}[0.4, 1]^2$). A careful reader will notice that in this case, our theoretical analysis remains unchanged, except for the part that computes the variance of the estimator. In the first set of simulations, we assumed a predefined action space of size 100 that both the Grinding and the EXP3 algorithm use, and the second one corresponds the fully continuous Grinding algorithm versus an EXP3 algorithm on a fixed discretization of the space. Each instance was run for $T = 2000$ timesteps, with 30 repetitions for each timestep. The solid plots correspond to the empirical mean of the regret in the predefined action space simulations, and to the cumulative loss in the continuous space ones, while the opaque bands correspond to the 10th and the 90th percentile.

For the predefined action space, we report the results using in-probability oracle as we assumed in previous sections under the name “dgrind”. Under the name “dgrind\_regression” we report our results for an approximation oracle, which uses past spammer data to train a logistic regression model for each $\Pr^m[\cdot]$. In the Supplementary Material, we explain more thoroughly how this logistic regression is run. In summary, we see that both with the use of an omnipotent oracle, and with the use of an approximation one, for every $\delta$, and for every $p$ our algorithm performs better than the EXP3 algorithm. Interestingly, the approximation oracle follows very closely the performance of the omnipotent one.

For the continuous action space, we implemented agnostic versions of both EXP3, and the Grinding algorithm. Since it is not generally possible to compute the best-fixed action in hindsight from the infimum of actions that $[-1, 1]^3$ contains, we resort to depicting the cumulative losses of the two algorithms. We use the logistic regression oracle for each of the polytopes again using only spammers’ data. As we see from the results in Figure 3 for the more realistic cases of $p = 0.6, 0.8$, the Grinding algorithm’s regret clearly outperforms that of EXP3. Furthermore, we have just run the Grinding algorithm with a relatively simple logistic regression oracle; for more advanced, better oracles, the results would be strengthened. The latter is also the reason that we see that in some cases with a majority of attackers ($p = 0.4$), we perform comparably to EXP3.

6 Discussion and Open Questions

In this paper, we have studied online learning against $\delta$-bounded, strategic agents in classification settings, for which we provided the Grinding algorithm. We complemented our theoretical analysis with simulations, showing first the benefits of our algorithm with an omnipotent oracle, and second, that there are approximation oracles that perform comparably well.

There is a number of interesting questions stemming from our work. On a technical level, we strongly conjecture that there is a matching lower bound for the Stackelberg regret in online classification settings. Further, an interesting research direction is to provide both theoretical results regarding the Stackelberg regret experienced by the learner for the case that she uses approximation oracles, similar to the ones that we used in the simulations, and experimental ones for different variants of best-response oracles. In many natural settings (mostly the ones that are considered in the fairness community, as mentioned by Bechavod et al. [6]), the learner can only observe the labels of some of the agents. This extra partial observability imposes an extra challenge, and cannot be handled by our framework currently. Moreover, our current analysis is based on the fact that the learner knows $\delta$ a priori. Additionally, no studies exist currently about an estimation of $\delta$. We believe that this can only be achieved through experiments with human subjects in as realistic conditions as possible. Finally, we do believe that the agents’ real feature vectors are neither fully stochastic, nor fully adversarial. This model formulation is similar to the one considered in the recent line of works in the online learning literature on the best-of-both-worlds [11, 30, 4, 29] regret bounds. We think it would be very interesting to provide theoretical bounds for the Stackelberg regret in such strategic classification settings.

\[12\] We cannot empirically verify this close connection in the continuous case, since, without being able to predict $r_t(\alpha)$ for each $\alpha$ given $x_t$, the number of potential actions that the omnipotent oracle would need in order to run is infinite.
Figure 2: Grinding vs. EXP3 for Predefined Action Set: both with the use of an omnipotent, and an approximation oracle, Grinding regret converges faster than that of EXP3.
Figure 3: Continuous Grinding vs. EXP3: Continuous Grinding using only approximation oracles converges clearly faster or comparably with EXP3i.
References


Appendix A  Supplementary Material for Section 3

A.1 Policy Regret

Policy regret (introduced by Arora et al. [2] and further studied in [3]) captures the fact that more powerful adversaries, which they term \(m\)-memory bounded adversaries, can adapt their responses to the last \(m\) actions taken by the learner. For example, according to their definition, external regret corresponds to \(m = 0\)-memory bounded adversaries. The key difference between Stackelberg and policy regret is that in the latter, the adversaries can adapt their response according to the memory frame, in any way they want. However, in the former their report is always the best-response to the current action that the learner has committed to. As a result, it might originally seem to be the case that the Policy Regret is more powerful in the sense that if an algorithm yields no-policy regret, then it should also yield no-Stackelberg regret. Contrary to this intuition, we prove below that the two notions of regret are worst-case incompatible.

Theorem A.1. There exists a repeated Stackelberg game between a learner and an agent, such that every action sequence with sublinear Stackelberg regret incurs linear policy regret, and every action sequence with sublinear policy regret incurs linear Stackelberg regret.

Proof. We present our results for an agent with \(m = 2\) - bounded memory. Let \(A = \{\alpha_1, \alpha_2\}\) denote the action space and let us define a General Stackelberg Game as follows:

- \(\ell(\alpha_1, r_t(\alpha_1)) = 1/2, t \in [T]\), if the agent bases \(r_t(\alpha_1)\) only on the current timestep \(t\)
- \(\ell(\alpha_1, r_t(\alpha_1)) = 1, t \in [T]\), if the past two actions taken by the learner were equal to \(\alpha_1\) and the agent bases \(r_t(\alpha_1)\) on the last two timesteps
- \(\ell(\alpha_2, r_t(\alpha_2)) = 1, t \in [T]\), if the agent bases \(r_t(\alpha_2)\) only on the current timestep \(t\)
- \(\ell(\alpha_2, r_t(\alpha_2)) = 1/2, t \in [T]\), if the past two actions taken by the learner were equal to \(\alpha_2\) and the agent bases \(r_t(\alpha_2)\) on the last two timesteps

We will first prove that any sequence with sublinear Stackelberg regret will have linear policy regret. Observe that for the Stackelberg regret, the best fixed action in hindsight is action \(\alpha_1\) with cumulative loss \(T/2\). Therefore, any sequence of \(T\) actions that the agent can base her best response only on the current timestep that yields sublinear Stackelberg regret must have cumulative loss \(T/2 + o(T)\). Therefore, action \(\alpha_1\) must be played a total of \(T - o(T)\) times, while action \(\alpha_2\) a total of at most \(o(T)\) times. For the \(m = 2\) bounded memory adversary, the best fixed action in hindsight is action \(\alpha_2\) with a total loss \(o(T)\) of \(T/2\).

Moving forward, we will prove that any sequence with sublinear policy regret, will have linear Stackelberg regret. As we mentioned above, the best strategy, when playing against the aforementioned \(m = 2\) memory bounded adversary, is to play action \(\alpha_2\) for the total of \(T\) timesteps, yielding a loss of \(T/2\). Hence, any sequence with sublinear policy regret, should include at least \(T - o(T)\) times the action \(\alpha_2\) and \(o(T)\) times the action \(\alpha_1\). This, however, implies that we have constructed a sequence with linear Stackelberg regret.

A.2 Missing Proofs from Section 3

Proof of Theorem 3.1. Let an action space \(A = \{\alpha_1, \alpha_2, \alpha_3\}\) and a repeated Stackelberg Game defined in Table 4 for all \(t \in [T]\), where each entry \((i, j)\) corresponds to the loss \(\ell(i, j)\) incurred by the learner when she commits to action \(i\), while the agent commits to action \(j\).

Let \(\{\alpha_i\}_{i=1}^T\) define the sequence of actions that the learner chooses in \(T\) timesteps. We will first prove that any sequence with sublinear Stackelberg regret, will have linear external regret. Observe that for the Stackelberg regret, the best fixed action in hindsight is action \(\alpha_1\) and action \(\alpha_2\), with cumulative loss \(\frac{T}{2}\).

\[\text{This loss corresponds to the } m\text{-bounded loss of the best action in hindsight.}\]
Therefore, any action sequence that yields sublinear Stackelberg regret must have cumulative loss $\frac{T}{2} + o(T)$, meaning that action $\alpha^3$ is played at most $o(T)$ times, while actions $\alpha^1$ and $\alpha^2$ combined are played at least $(T - o(T))$ times. Let $w_1$ be the fraction of the times that action $\alpha^1$ is played, and $1-w_1$ the fraction of the times that $\alpha^2$ is played among the total of $T - o(T)$ times. Let $\{\alpha_t\}_{t=1}^T$ define the sequence of actions that the learner chooses in $T$ timesteps. We will first prove that any sequence with sublinear Stackelberg regret, will have linear external regret. Observe that for the Stackelberg regret, the best fixed action in hindsight is action $\alpha^1$ and action $\alpha^2$, with cumulative loss $\frac{T}{2}$. Therefore, any action sequence that yields sublinear Stackelberg regret must have cumulative loss $\frac{T}{2} + o(T)$, meaning that action $\alpha^3$ is played at most $o(T)$ times, while actions $\alpha^1$ and $\alpha^2$ combined are played at least $(T - o(T))$ times. Let $w_1$ be the fraction of the times that action $\alpha^1$ is played, and $1-w_1$ the fraction of the times that $\alpha^2$ is played among the total of $T - o(T)$ times.

We proceed by identifying the best fixed action for the external regret in any action such sequence $\{\alpha_t\}_{t=1}^T$. We distinguish the following three cases:

1. Assume that action $\alpha^1$ was the fixed action in hindsight for the sequence $\{\alpha_t\}_{t=1}^T$. Then, the cumulative loss incurred by playing $\alpha^1$, denoted by $\sum_{t=1}^T \ell(\alpha^1, r_t(\alpha_t))$:

$$
\frac{T - o(T)}{2} + \frac{(1-w_1)}{6} (T - o(T)) + \frac{1 \cdot o(T)}{3} \leq \frac{T}{2} + o(T)
$$

2. Assume that action $\alpha^2$ was the fixed action in hindsight for the aforementioned action sequence. Then, the cumulative loss incurred by playing $\alpha^2$, denoted by $\sum_{t=1}^T \ell(\alpha^2, r_t(\alpha_t))$ is equal to

$$
\frac{T - o(T)}{2} + \frac{(1-w_1)}{2} (T - o(T)) + \frac{1 \cdot o(T)}{2} \leq \frac{T}{2} + o(T)
$$

3. Finally, assume that action $\alpha^3$ was the fixed action in hindsight for the aforementioned action sequence. Then, the cumulative loss incurred by playing $\alpha^3$, denoted by $\sum_{t=1}^T \ell(\alpha^3, r_t(\alpha_t))$

$$
\frac{T - o(T)}{3} + \frac{(1-w_1)}{4} (T - o(T)) + \frac{1 \cdot o(T)}{4} \leq \frac{T}{3} + o(T)
$$

Hence, we have that the best fixed action in hindsight for the external regret for the sequence $\{\alpha_t\}_{t=1}^T$ is action $\alpha^3$. This means, however, that for the sequence $\{\alpha_t\}_{t=1}^T$, which guaranteed sublinear Stackelberg regret, the external regret is linear in $T$:

$$
R(T) \geq \frac{T - o(T)}{2} + o(T) - \frac{T}{3} - o(T) \geq \frac{T}{6} + o(T)
$$

<table>
<thead>
<tr>
<th>Learner</th>
<th>$r_1(\alpha^1)$</th>
<th>$r_1(\alpha^2)$</th>
<th>$r_1(\alpha^3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha^1$</td>
<td>1/2</td>
<td>1/6</td>
<td>1</td>
</tr>
<tr>
<td>$\alpha^2$</td>
<td>1/2</td>
<td>1/2</td>
<td>1</td>
</tr>
<tr>
<td>$\alpha^3$</td>
<td>1/3</td>
<td>1/4</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1: Example Instance
Similarly to the analysis above, we distinguish the following cases:

1. Assume that action $\alpha^1$ was the fixed action in hindsight for the aforementioned action sequence. Then, the cumulative loss incurred by playing $\alpha^1$, denoted by $\sum_{t=1}^{T} \ell (\alpha^1, r_t(\alpha_t))$, is:
   \[
   w_1 \frac{o(T)}{2} + \left(1 - w_1\right) \frac{o(T)}{6} + T - o(T)
   \]

2. Assume that action $\alpha^2$ was the fixed action in hindsight for the aforementioned action sequence. Then, the cumulative loss incurred by playing $\alpha^2$, denoted by $\sum_{t=1}^{T} \ell (\alpha^2, r_t(\alpha_t))$, is equal to:
   \[
   w_1 \frac{o(T)}{2} + o(T)(1 - w_1)\frac{1}{2} + T - o(T)
   \]

3. Finally, assume that action $\alpha^3$ was the fixed action in hindsight for the aforementioned action sequence. Then, the cumulative loss incurred by playing $\alpha^3$, denoted by $\sum_{t=1}^{T} \ell (\alpha^3, r_t(\alpha_t))$, is:
   \[
   \frac{w_1}{3} o(T) + \left(1 - \frac{w_1}{2}\right) + T - o(T)
   \]

Hence, no matter which one is the best fixed action in hindsight (a decision which will depend on the constants of the term $o(T)$), we can guarantee that in this case, the external regret will be $o(T)$, i.e., sublinear. This concludes our proof.

We proceed by identifying the best fixed action for the external regret in any action sequence $\{\alpha_t\}_{t=1}^{T}$. We distinguish the following three cases:

1. Assume that action $\alpha^1$ was the fixed action in hindsight for the sequence $\{\alpha_t\}_{t=1}^{T}$. Then, the cumulative loss incurred by playing $\alpha^1$, denoted by $\sum_{t=1}^{T} \ell (\alpha^1, r_t(\alpha_t))$:
   \[
   \frac{w_1}{2} \cdot \frac{T - o(T)}{2} + \frac{(1 - w_1)}{6} (T - o(T)) + \frac{1}{2} \cdot o(T) = \frac{T}{3} \left( w_1 + \frac{1}{2} \right) + o(T)
   \]

2. Assume that action $\alpha^2$ was the fixed action in hindsight for the aforementioned action sequence. Then, the cumulative loss incurred by playing $\alpha^2$, denoted by $\sum_{t=1}^{T} \ell (\alpha^2, r_t(\alpha_t))$ is equal to
   \[
   \frac{w_1}{2} \cdot \frac{T - o(T)}{2} + \frac{(1 - w_1)}{2} (T - o(T)) + \frac{1}{2} \cdot o(T) = \frac{T}{2} + o(T)
   \]
3. Finally, assume that action $\alpha^3$ was the fixed action in hindsight for the aforementioned action sequence. Then, the cumulative loss incurred by playing $\alpha^3$, denoted by $\sum_{t=1}^{T} \ell(\alpha^3, r_t(\alpha_t))$

\[
\frac{w_1 T - o(T)}{3} + \frac{(1 - w_1) (T - o(T))}{4} + \frac{1 \cdot o(T)}{4} \leq \frac{T}{4} \left(\frac{w_1}{3} + 1\right) + o(T)
\]

Hence, we have that the best fixed action in hindsight for the external regret for the sequence $\{\alpha_t\}_{t=1}^{T}$ is action $\alpha^3$. This means, however, that for the sequence $\{\alpha_t\}_{t=1}^{T}$, which guaranteed sublinear Stackelberg regret, the external regret is linear in $T$:

\[
R(T) = o(T) + o(T) - T = \frac{T}{6} + o(T)
\]

Moving forward, we will now prove that any sequence with sublinear External regret will have linear Stackelberg regret. Since we previously proved that any action sequence $\{\alpha_t\}_{t=1}^{T}$ with sublinear Stackelberg regret plays at least $T - o(T)$ times actions $\alpha^1$ and $\alpha^2$ and this resulted in having linear External regret, we only need to consider sequences where action $\alpha^3$ is played $T - o(T)$ times, while actions $\alpha^1$ and $\alpha^2$ are played for a combined number of $o(T)$ times. Again, let $w_1$ denote the fraction of the times that action $\alpha^1$ was played and $1 - w_1$ denote the fraction of the times that $\alpha^2$ was played, among the total of $o(T)$ times.

For any such action sequence, $\{\alpha_t\}_{t=1}^{T}$, it suffices to show that the External regret yielded will be sublinear, since, clearly, for any such sequence the Stackelberg regret will be linear:

\[
\mathcal{R}(T) = \frac{o(T)}{2} + \frac{o(T)}{2} + T - o(T) - \frac{T}{2} = \frac{T}{2}
\]

Similarly to the analysis above, we distinguish the following cases:

1. Assume that action $\alpha^1$ was the fixed action in hindsight for the aforementioned action sequence. Then, the cumulative loss incurred by playing $\alpha^1$, denoted by $\sum_{t=1}^{T} \ell(\alpha^1, r_t(\alpha_t))$, is:

\[
\frac{w_1 o(T)}{2} + (1 - w_1) \frac{o(T)}{6} + T - o(T)
\]

2. Assume that action $\alpha^2$ was the fixed action in hindsight for the aforementioned action sequence. Then, the cumulative loss incurred by playing $\alpha^2$, denoted by $\sum_{t=1}^{T} \ell(\alpha^2, r_t(\alpha_t))$, is equal to:

\[
\frac{w_1 o(T)}{2} + o(T)(1 - w_1) \frac{1}{2} + T - o(T)
\]

3. Finally, assume that action $\alpha^3$ was the fixed action in hindsight for the aforementioned action sequence. Then, the cumulative loss incurred by playing $\alpha^3$, denoted by $\sum_{t=1}^{T} \ell(\alpha^3, r_t(\alpha_t))$, is:

\[
\frac{w_1 o(T)}{3} + \frac{(1 - w_1)}{4} o(T) + T - o(T)
\]

Hence, no matter which one is the best fixed action in hindsight (a decision which will depend on the constants of the term $o(T)$), we can guarantee that in this case, the external regret will be $o(T)$, i.e., sublinear. This concludes our proof.

\[\Box\]

\textit{Proof of Theorem 3.2.} Let an action space $A = \{\alpha^1, \alpha^2\}$ and a repeated Stackelberg Game instance defined in Table \[\], where each entry $(i, j)$ corresponds to the loss $\ell(i, j)$ incurred by the learner when she commits to action $i$, while the agent commits to action $j$. Loss functions $\ell_1, \ell_2$ are defined as follows:
• \( \ell_1(t) = 0, t \in \{1, \ldots, T/2\} \), \( \ell_1(t) = 1/2, t \in \{T/2 + 1, \ldots, T\} \), if in the first \( T/2 \) rounds the number of times that action \( \alpha_1 \) is played is larger than the number of times that action \( \alpha_2 \) is played and \( \ell_1(t) = 2, t \in \{T/2 + 1, \ldots, T\} \), otherwise.

• \( \ell_2(t) = 0, t \in \{1, \ldots, T/2\} \), \( \ell_2(t) = 2/3, t \in \{T/2 + 1, \ldots, T\} \), if in the first \( T/2 \) rounds the number of times that action \( \alpha_1 \) is played is larger than the number of times that action \( \alpha_2 \) is played and \( \ell_2(t) = 2, t \in \{T/2 + 1, \ldots, T\} \), otherwise.

Let \( \{\alpha_i\}_{t=1}^T \) the action sequence in \( T \) rounds. We will first prove that any sequence with sublinear Stackelberg regret, will have linear strategic regret. Observe that for the Stackelberg Regret, the best fixed action in hindsight is action \( \alpha_1 \) with cumulative loss \( T/2 \). Therefore, any \( \{\alpha_i\}_{t=1}^T \) that yields sublinear Stackelberg regret must have cumulative loss \( T/2 + o(T) \), meaning that action \( \alpha_2 \) is played at most \( o(T) \) times, while action \( \alpha_1 \) is played at least \( T - o(T) \) times.

We proceed by identifying the best fixed action in hindsight for the strategic regret in any such sequence \( \{\alpha_i\}_{t=1}^T \). Since \( \alpha_1 \) is played for a total of at least \( T - o(T) \), we can derive that it was played more times than \( \alpha_2 \) in the first \( T/2 \) timesteps. As a result, we fixed action \( \alpha_1 \) in hindsight we would incur loss equal to \( T/4 \), while if we fixed action \( \alpha_2 \) in hindsight, we would incur loss equal to \( T \). Thus, the best fixed action in hindsight is action \( \alpha_1 \), which, however, yields strategic regret equal to \( T/4 + o(T) \) (i.e., linear in \( T \)).

Moving forward, we will prove that any sequence with sublinear strategic regret will have linear Stackelberg regret. It suffices to study the case where action \( \alpha_2 \) is played \( T - o(T) \) times, while action \( \alpha_1 \) is played \( o(T) \) times. Observe that in that case, the best fixed action in hindsight is action \( \alpha_2 \) which yields a total strategic regret of \( o(T) \). However, the Stackelberg regret now has become linear in \( T \).

**Proof of Theorem 3.3** The proof is identical to the proof of Theorem 3.2 with the only difference that \( \ell(\alpha_1) \) (resp. \( \ell(\alpha_2) \)) is always defined to be 1/2 (resp. 1), while the loss if the agent was truthful remains \( \ell_1(t) \) (resp. \( \ell_2(t) \)), as defined by the proof of Theorem 3.2.

### A.3 Purely Adversarial and Cooperative Stackelberg Games

Despite the worst-case incompatibility results that we have shown for the different regret notions, there are families for which there is a clear hierarchy in the strength of these regret notions. In this subsection, we will study two of the most important ones; the family of Purely Adversarial, and the family of Purely Cooperative Stackelberg Games.

**Definition A.1** (Purely Adversarial Stackelberg Game (PASGs)). We will call a Stackelberg Game, where a learner commits to an action \( \alpha \in \mathcal{A} \) and an agent best-responds to it with \( r(\alpha) \), Purely Adversarial, if for all actions \( \alpha' \in \mathcal{A} \) for the loss of the learner it holds that: \( \ell(\alpha, r(\alpha)) \geq \ell(\alpha, r(\alpha')) \), i.e., the agent inflicts the higher loss to the learner, when best responding to the action to which she committed.

**Definition A.2** (Purely Cooperative Stackelberg Game (PCSGs)). We will call a Stackelberg Game, where a learner commits to an action \( \alpha \in \mathcal{A} \) and an agent best-responds to it with \( r(\alpha) \), Purely Cooperative if for all actions \( \alpha' \in \mathcal{A} \) for the loss of the learner it holds that: \( \ell(\alpha, r(\alpha)) \leq \ell(\alpha, r(\alpha')) \), i.e., the agent inflicts the lowest loss to the learner, when best responding to the action to which she committed.

We remark here that despite their similarities, PASGs and PCSGs are not equivalent to zero-sum games; in fact, it is easy to see that every zero-sum game is either a PASG or a PCSG, but the converse is not true.

<table>
<thead>
<tr>
<th>Learner</th>
<th>( r_1(\alpha_1) )</th>
<th>( r_1(\alpha_2) )</th>
<th>( r'_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_1 )</td>
<td>1/2</td>
<td>-</td>
<td>( \ell_1(t) )</td>
</tr>
<tr>
<td>( \alpha_2 )</td>
<td>-</td>
<td>1</td>
<td>( \ell_2(t) )</td>
</tr>
</tbody>
</table>

Table 2: Example Instance
Lemma A.1. In repeated PASGs, Stackelberg regret is upper bounded by external regret, i.e., $R(T) \leq R(T)$.

Proof. Let $\hat{\alpha} = \arg\min_{\alpha \in A} \sum_{t=1}^{T} \ell(\alpha, r_t(\alpha))$ (i.e., best fixed action in hindsight, if the agents do not change their strategies to best-respond to the action $\hat{\alpha}$) and $\alpha^* = \arg\min_{\alpha \in A} \sum_{t=1}^{T} \ell(\alpha, r_t(\alpha))$ (i.e., best fixed action in hindsight, if the agents do change their strategies to best-respond to the action $\alpha^*$). Then, we have:

$$R(T) = \sum_{t=1}^{T} \ell(\alpha_t, r_t(\alpha_t)) - \sum_{t=1}^{T} \ell(\hat{\alpha}, r_t(\alpha_t))$$ (definition of external regret)

$$\geq \sum_{t=1}^{T} \ell(\alpha_t, r_t(\alpha_t)) - \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha_t))$$ (definition of $\hat{\alpha}$)

$$\geq \sum_{t=1}^{T} \ell(\alpha_t, r_t(\alpha_t)) - \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha^*)) \geq (\ell(\alpha^*, r_t(\alpha_t)) \leq \ell(\alpha^*, r_t(\alpha^*))$$

$$= R(T)$$

Hence, in repeated PASGs any no-external regret sequence of actions is also a no-Stackelberg regret sequence.

Lemma A.2. In repeated PASGs, strategic regret is lower bounded by Stackelberg regret, i.e., $R(T) \geq R(T)$.

Proof. For $\alpha^* = \arg\min_{\alpha \in A} \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha^*))$ (i.e., best fixed action in hindsight, if the agents do change their strategies to best-respond to the action $\alpha^*$)

$$\mathcal{R}(T) = \sum_{t=1}^{T} \ell(\alpha_t, r_t(\alpha_t)) - \min_{\alpha \in A} \sum_{t=1}^{T} \ell(\alpha, r_t(\alpha))$$

$$= \sum_{t=1}^{T} \ell(\alpha_t, r_t(\alpha_t)) - \min_{\alpha \in A} \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha^*)) - \min_{\alpha \in A} \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha^*)) + \min_{\alpha \in A} \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha^*))$$

$$= R(T) + \left[ -\min_{\alpha \in A} \sum_{t=1}^{T} \ell(\alpha, r_t^*) + \min_{\alpha^* \in A} \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha^*)) \right]$$

$$\geq R(T)$$

where the inequality holds since when agents are best responding, then they can even lie, thus incurring higher loss to the agent compared to when they report truthfully.}

Hence, in repeated PASGs any no-Strategic regret sequence of actions is also a no-Stackelberg regret sequence.

Lemma A.3. In repeated PASGs, strategic regret is lower bounded by external regret, i.e., $\mathcal{R}(T) \geq R(T)$. 

Table 3: Example of a PASG that is not zero-sum.

(see e.g., the example loss matrix given in Table 3 where the first coordinate of tuple $(i,j)$ corresponds to the loss of the learner, and the second to the loss of the agent).
Proof. Let $\Delta = R(T) - \mathcal{R}(T)$, $\hat{\alpha} = \arg\min_{\alpha \in \mathcal{A}} \sum_{t=1}^{T} \ell(\alpha, r^*_t)$ and $\alpha^* = \arg\min_{\alpha \in \mathcal{A}} \ell(\alpha, r_t(\alpha_t))$. Expanding $\Delta$:

$$
\Delta = \sum_{t=1}^{T} \ell(\hat{\alpha}, r^*_t) - \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha_t)) \\
\leq \sum_{t=1}^{T} \ell(\alpha_t, r^*_t) - \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha_t)) \\
\leq \sum_{t=1}^{T} \ell(\alpha_t, r_t(\alpha_t)) - \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha_t)) \\
\leq 0
$$

where the last inequality comes from the definition of PASGs.

Hence, in repeated PASGs any no-strategic regret sequence of actions is also a no-external regret sequence. Based on Lemmas A.1, A.2, A.3 we get the following.

**Corollary A.1.** In PASGs the hierarchy of regret notions is: $\mathcal{R}(T) \leq R(T) \leq \mathcal{R}(T)$.

Let us now move on to repeated PCSGs.

**Lemma A.4.** In repeated PCSGs, Stackelberg regret is lower bounded by external regret, i.e., $\mathcal{R}(T) \geq R(T)$.

Proof. Let $\hat{\alpha} = \arg\min_{\alpha \in \mathcal{A}} \sum_{t=1}^{T} \ell(\alpha, r_t(\alpha_t))$ (i.e., best fixed action in hindsight, if the agents do not change their strategies to best-respond to the action $\hat{\alpha}$) and $\alpha^* = \arg\min_{\alpha \in \mathcal{A}} \sum_{t=1}^{T} \ell(\alpha, r_t(\alpha))$ (i.e., best fixed action in hindsight, if the agents do change their strategies to best-respond to the action $\alpha^*$). Then, we have:

$$
R(T) = \sum_{t=1}^{T} \ell(\alpha_t, r_t(\alpha_t)) - \sum_{t=1}^{T} \ell(\hat{\alpha}, r_t(\alpha_t)) \\
= \sum_{t=1}^{T} \ell(\alpha_t, r_t(\alpha_t)) - \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha^*)) - \sum_{t=1}^{T} \ell(\hat{\alpha}, r_t(\alpha_t)) + \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha^*)) \\
\leq \mathcal{R}(T) + \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha_t)) - \sum_{t=1}^{T} \ell(\hat{\alpha}, r_t(\alpha_t)) \\
(\ell(\alpha^*, r_t(\alpha^*)) \leq \ell(\alpha^*, r_t(\alpha_t))) \\
\leq \mathcal{R}(T) \\
(\hat{\alpha} = \arg\min_{\alpha \in \mathcal{A}} \sum_{t=1}^{T} \ell(\alpha, r_t(\alpha_t)))
$$

where the first inequality comes from the definition of Purely Cooperative Stackelberg Games.

Hence, in repeated PCSGs any no-Stackelberg regret sequence of actions is also a no-External regret sequence.

**Lemma A.5.** In repeated PCSGs, strategic regret is upper bounded by Stackelberg regret, i.e., $\mathcal{R}(T) \leq \mathcal{R}(T)$.

Proof.

$$
\mathcal{R}(T) = \sum_{t=1}^{T} \ell(\alpha_t, r_t(\alpha_t)) - \min_{\alpha \in \mathcal{A}} \sum_{t=1}^{T} \ell(\alpha, r^*_t) \\
= \sum_{t=1}^{T} \ell(\alpha_t, r_t(\alpha_t)) - \min_{\alpha^* \in \mathcal{A}} \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha^*)) - \min_{\alpha \in \mathcal{A}} \sum_{t=1}^{T} \ell(\alpha, r^*_t) + \min_{\alpha^* \in \mathcal{A}} \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha^*)) \\
= \mathcal{R}(T) + \left[ -\min_{\alpha \in \mathcal{A}} \sum_{t=1}^{T} \ell(\alpha, r^*_t) + \min_{\alpha^* \in \mathcal{A}} \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha^*)) \right] \\
\leq \mathcal{R}(T)
$$

19
where the inequality comes from the definition of PCSGs and that when we constraint the agent to reporting truthfully, \( r^*_t \neq r_t(\alpha) \) potentially.

Hence, in repeated PCSGs any no-Stackelberg regret sequence of actions is also a no-strategic regret sequence.

**Lemma A.6.** In repeated PCSGs, strategic regret is upper bounded by external regret, i.e., \( R(T) \leq R(T) \).

**Proof.** Let \( \Delta = R(T) - R(T) \), \( \hat{a} = \arg\min_{a \in A} \sum_{t=1}^{T} \ell(\alpha, r^*_t) \) and \( \alpha^* = \arg\min_{a \in A} \sum_{t=1}^{T} \ell(\alpha, r_t(\alpha_t)) \). Expanding \( \Delta \):

\[
\Delta = \sum_{t=1}^{T} \ell(\hat{a}, r^*_t) - \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha_t)) \\
\geq \sum_{t=1}^{T} \ell(\alpha_t, r^*_t) - \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha_t)) \\
\geq \sum_{t=1}^{T} \ell(\alpha_t, r_t(\alpha_t)) - \sum_{t=1}^{T} \ell(\alpha, r_t(\alpha_t)) \\
\geq 0
\]

where the last inequality comes from the definition of a PCSG.

Hence, in repeated PCSGs any no-external regret sequence of actions is also a no-strategic regret sequence. Based on Lemmas A.4, A.5, A.6 we get the following.

**Corollary A.2.** In PCSGs the hierarchy of regret notions is: \( R(T) \leq R(T) \leq R(T) \).

### A.4 The Function \( \ell(\alpha, r_t(\alpha)) \)

In this subsection we will show that even in the case where the learner’s loss function is Lipschitz with respect to both its first and second argument, this does not guarantee that we will have a Lipschitz loss function if we give the agent the opportunity to best-respond. The following Lemma formalizes an argument first made by Balcan et al. [3].

**Lemma A.7.** Let \( \ell(x, y) \) denote the learner’s loss function in a Stackelberg game, such that \( \ell \) is \( L_1 \)-Lipschitz with respect to the first argument, and \( L_2 \)-Lipschitz with respect to the second. Then, for the learner’s loss between any two actions \( \alpha, \alpha' \in A \) it holds that:

\[
|\ell(\alpha, r_t(\alpha)) - \ell(\alpha', r_t(\alpha'))| \leq \max\{L_1 \cdot \|\alpha' - \alpha\|, L_2 \cdot \|r_t(\alpha) - r_t(\alpha')\|\}
\]

**Proof.** We distinguish the set of actions \( A \) into pairs \((\alpha, \alpha')\) with the following properties:

1. For the pair \((\alpha, \alpha')\) it holds that: \( \ell(\alpha, r_t(\alpha)) \geq \ell(\alpha, r_t(\alpha')) \) and \( \ell(\alpha', r_t(\alpha')) \geq \ell(\alpha', r_t(\alpha)) \). Observe that, given that \( \ell \) is \( L_1 \)-Lipschitz in its first argument, we have that:

\[
\ell(\alpha', r_t(\alpha')) - \ell(\alpha, r_t(\alpha')) \geq \ell(\alpha', r_t(\alpha)) - \ell(\alpha, r_t(\alpha)) \geq -L_1 \|\alpha' - \alpha\|
\]

and

\[
\ell(\alpha', r_t(\alpha')) - \ell(\alpha, r_t(\alpha)) \leq \ell(\alpha', r_t(\alpha')) - \ell(\alpha, r_t(\alpha')) \leq L_1 \|\alpha' - \alpha\|
\]

Therefore, for these pairs of actions \((\alpha, \alpha')\), the function \( \ell(\alpha, r_t(\alpha)) \) is \( L_1 \)-Lipschitz with respect to \( \alpha \).

2. For the pair \((\alpha, \alpha')\) it holds that: \( \ell(\alpha, r_t(\alpha)) \leq \ell(\alpha, r_t(\alpha')) \) and \( \ell(\alpha', r_t(\alpha')) \leq \ell(\alpha', r_t(\alpha)) \). Similarly to Case [1] it is easy to see that on these pairs of actions, function \( \ell(\alpha, r_t(\alpha)) \) is again \( L_1 \)-Lipschitz with respect to \( \alpha \).
3. For the pair \((\alpha, \alpha')\) it holds that
\[
\ell(\alpha, r_\ell(\alpha)) \geq \ell(\alpha, r_\ell(\alpha')) \tag{1}
\]
and
\[
\ell(\alpha', r_\ell(\alpha')) \leq \ell(\alpha', r_\ell(\alpha)) \tag{2}
\]
From Equations (1) and (2) we have that
\[
\ell(\alpha', r_\ell(\alpha')) - \ell(\alpha, r_\ell(\alpha)) \leq L_1 \|\alpha' - \alpha\| \tag{3}
\]
Let us now further distinguish the following cases:

(a) \(\ell(\alpha, r_\ell(\alpha)) = \ell(\alpha', r_\ell(\alpha))\). Then, trivially it holds that \(|\ell(\alpha, r_\ell(\alpha)) - \ell(\alpha', r_\ell(\alpha'))| \leq L_1 \cdot \|\alpha' - \alpha\|\).

(b) \(\ell(\alpha, r_\ell(\alpha)) \leq \ell(\alpha', r_\ell(\alpha))\). Combining this with Equation (3), we get: \(|\ell(\alpha, r_\ell(\alpha)) - \ell(\alpha', r_\ell(\alpha'))| \leq L_1 \cdot \|\alpha' - \alpha\|\).

(c) \(\ell(\alpha, r_\ell(\alpha)) \geq \ell(\alpha', r_\ell(\alpha'))\) Observe now that if \(\ell(\alpha, r_\ell(\alpha)) \geq \ell(\alpha', r_\ell(\alpha))\), then from Equation (2) the latter is lower bounded by \(\ell(\alpha', r_\ell(\alpha'))\), which leads to a contradiction. Hence, it has to be the case that \(\ell(\alpha, r_\ell(\alpha)) \leq \ell(\alpha', r_\ell(\alpha))\). The latter, combined with the assumption that \(\ell\) is \(L_2\) - Lipschitz with respect to its second argument, implies that \(\ell(\alpha', r_\ell(\alpha')) - \ell(\alpha, r_\ell(\alpha)) \geq -L_2 \cdot \|r_\ell(\alpha') - r_\ell(\alpha)\|\).

4. For the pair \((\alpha, \alpha')\) it holds that \(\ell(\alpha, r_\ell(\alpha)) \leq \ell(\alpha, r_\ell(\alpha'))\) and \(\ell(\alpha', r_\ell(\alpha)) \geq \ell(\alpha', r_\ell(\alpha))\). The case is analogous to Case 3.

\[\text{▲}\]

To summarize, in PASGs and PCSGs the loss function written in terms of the action of the agent is \textit{Lipschitz}, i.e., \(|\ell(\alpha, r_\ell(\alpha)) - \ell(\alpha', r_\ell(\alpha'))| \leq L_1 \cdot \|\alpha' - \alpha\|\). However, in General Stackelberg Games it holds that
\[
|\ell(\alpha, r_\ell(\alpha)) - \ell(\alpha', r_\ell(\alpha'))| \leq \max \{L_1 \cdot \|\alpha' - \alpha\|, L_2 \cdot \|r_\ell(\alpha') - r_\ell(\alpha)\|\} \tag{4}
\]

Despite the fact that the latter means that \(\ell(\alpha, r_\ell(\alpha))\) is not generally Lipschitz, there are some Stackelberg settings where \(\|r_\ell(\alpha') - r_\ell(\alpha)\|\) can be upper bounded by the \(\|\alpha' - \alpha\|\) multiplied by a constant. For example, from well known results in convex optimization (for completeness see Lemma A.8), we can see that this is exactly the case in settings where the agent’s utility function, \(u_\ell(\alpha, r)\) is \textit{strongly} concave in \(r\), and quasilinear\footnote{Quasilinearity in \(\alpha\) establishes that \(L_{f,g}\) which is used by Lemma A.8 will be linear in \(\|\alpha' - \alpha\|\).} in \(\alpha\).

\textbf{Lemma A.8 (Closeness of Maxima of Strongly Concave Functions (folklore))}. Let functions \(f : \mathcal{X} \rightarrow \mathbb{R}, g : \mathcal{X} \rightarrow \mathbb{R}\) be two multidimensional, \(1/\eta_c\)-strongly concave functions with respect to some norm \(\|\cdot\|\). Let \(h(x) = f(x) - g(x), x \in \mathcal{X}\) be \(L_{f,g}\)-Lipschitz\footnote{We use the subscript \(f,g\) in the Lipschitzness constant to denote the fact that it depends on the two functions \(f, g\).} with respect to the same norm \(\|\cdot\|\). Then, for the maxima of the two functions: \(\mu_f = \arg \max_{x \in \mathcal{X}} f(x)\) and \(\mu_g = \arg \max_{x \in \mathcal{X}} g(x)\) it holds that:
\[
\|\mu_f - \mu_g\| \leq L_{f,g} \cdot \eta_c \tag{5}
\]

\textit{Proof.} First, we take the Taylor expansion of \(f\) around its maximum, \(\mu_f\) and use the strong concavity condition:
\[
f(x) \leq f(\mu_f) + \langle \nabla f(\mu_f), x - \mu_f \rangle - \frac{1}{2\eta} \|\mu_f - x\|^2 \tag{strong concavity}
\]
\[
f(\mu_f) - \frac{1}{2\eta} \|\mu_f - x\|^2 \tag{\nabla f(\mu_f) = 0, since \(\mu_f\) is the maximum}
\]
Similarly, by taking the Taylor expansion of $g$ around its maximum and using the strong concavity condition:

$$g(x) \leq g(\mu_g) - \frac{1}{2\eta}||\mu_g - x||^2$$

Using the $L_{f,g}$-Lipschitzness of $h(x)$ we get:

$$L_{f,g} \cdot ||\mu_g - \mu_f|| \geq |h(\mu_g) - h(\mu_f)| \geq h(\mu_g) - h(\mu_f)$$

$$\geq f(\mu_g) - f(\mu_f) + f(\mu_f) - g(\mu_g)$$

$$\geq \frac{1}{2\eta}||\mu_f - \mu_g||^2 + \frac{1}{2\eta}||\mu_f - \mu_g||^2$$

(from the Taylor expansions)

$$\geq \frac{1}{\eta}||\mu_f - \mu_g||^2$$

Dividing both sides with $||\mu_g - \mu_f||$ concludes the proof. ▲

An example of such a utility function in the context of strategic classification (similar to the family of utility functions used in [19]) is presented below.

Example. Let $u_t(\alpha, y(\alpha; x)) = \langle \alpha, y(\alpha; x) \rangle - (x - y(\alpha; x))^2$ be the utility function of the agent, where $y(\alpha; x)$ denotes the response of the agent when the learner commits to action $\alpha$, and the agent has real feature vector $x$. Then, we would like to compute an upper bound on the difference between $||r(\alpha) - r(\alpha')||$, where $r(\alpha) = \arg \max_{x' \in X} u_t(\alpha, y)$ and $r(\alpha') = \arg \max_{x' \in X} u_t(\alpha', y)$. Following Lemma A.8, we can define functions $f(y) = u_t(\alpha, y)$ and $g(y) = u_t(\alpha', y)$. Now, observe that function $h(y) = f(y) - g(y)$ is indeed $||\alpha - \alpha'||$-Lipschitz (i.e., the Lipschitzness constant depends on the specific actions:

$$|f(y) - g(y) - f(z) + g(z)| = ||\alpha - \alpha', y - z|| \leq ||\alpha - \alpha'|| \cdot ||y - z||$$

where the last inequality comes from the Cauchy-Schwartz inequality. Furthermore, observe that both $f(\cdot)$ and $g(\cdot)$ are $\frac{1}{2}$-strongly concave. Therefore, from Lemma A.8 we get that:

$$||r(\alpha) - r(\alpha')|| \leq \frac{||\alpha - \alpha'||}{2}$$

In this work, we focus on more general utility functions from the perspective of the agent.

 Appendix B Supplementary Material for Section 4

The proof of Theorem 4.1 follows from a sequence of lemmas and claims presented below.

Claim B.1. Let $A_t$ denote the two-stage induced probability distribution $A_t$, with $f_{A_t}$ denoting its probability density function. Then, $A_t$ is equivalent to a one-stage probability distribution of drawing directly an action from density $d\pi_t(\cdot)$.

Proof. The one-stage probability distribution that draws an action from $\pi_t$ is equivalent to choosing an action $\alpha \in A$ from probability density function: $d\pi_t(\alpha) = (1 - \gamma)d\pi_t(\alpha) + \frac{\gamma}{\lambda(A)}$. The two-stage probability is: $d\pi_{A_t}(\alpha) = \frac{1}{\lambda(p)} \left((1 - \gamma)q_t(p) + \frac{\gamma \lambda(p)}{\lambda(A)}\right)$. Since $q_t(p) = \lambda(p)d\pi_t(\alpha), \forall \alpha \in A$, we get the result. ▲

Moving forward we will first analyze the first and the second moment of the loss $\hat{\ell}(\alpha, r_t(\alpha))$ for each action $\alpha$, based on the induced probability distribution $A_t$, assuming that we are getting oracle access to $Pr^\pi_{A_t}[\alpha]$.

Lemma B.1 (First Moment). The estimated loss $\hat{\ell}(\alpha, r_t(\alpha))$ is an unbiased estimate of the true loss $\ell(\alpha, r_t(\alpha))$, when actions are drawn from the induced probability distribution, $A_t$. 

22
\textbf{Proof.} For all the actions $\alpha \in A$, given Claim \[B.1\] it holds that:
\[
\mathbb{E}_{\alpha \sim A_t} \left[ \hat{\ell}(\alpha, r_t(\alpha)) \right] = \int_{\alpha' \in A} f_{\mathcal{A}_t}(\alpha') \frac{\ell(\alpha, r_t(\alpha)) \mathbb{I}\{\alpha \in N^{out}(\alpha')\}}{Pr_{\mathcal{A}_t}^m[\alpha]} \, d\alpha' = \ell(\alpha, r_t(\alpha)) \tag{$\triangleright$}
\]

\textbf{Lemma B.2 (Second Moment).} For the second moment of the estimated loss $\hat{\ell}(\alpha, r_t(\alpha))$ with respect to the induced probability distribution $A_t$ it holds that:
\[
\mathbb{E}_{\alpha \sim A_t} \left[ \left( \hat{\ell}(\alpha, r_t(\alpha)) \right)^2 \right] = \frac{\ell(\alpha, r_t(\alpha))^2}{Pr_{\mathcal{A}_t}^m[\alpha]} \leq \frac{1}{Pr_{\mathcal{A}_t}^m[\alpha]} \tag{$\triangleright$}
\]

\textbf{Proof.} For all the actions $\alpha \in A$, given Claim \[B.1\] it holds that:
\[
\mathbb{E}_{\alpha \sim A_t} \left[ \left( \hat{\ell}(\alpha, r_t(\alpha)) \right)^2 \right] = \int_{\alpha' \in A} f_{\mathcal{A}_t}(\alpha') \frac{\ell(\alpha, r_t(\alpha))^2 \mathbb{I}\{\alpha \in N^{out}(\alpha')\}}{Pr_{\mathcal{A}_t}^m[\alpha]^2} \, d\alpha' = \frac{\ell(\alpha, r_t(\alpha))^2}{Pr_{\mathcal{A}_t}^m[\alpha]} \leq \frac{1}{Pr_{\mathcal{A}_t}^m[\alpha]} \tag{$\triangleright$}
\]

A technical lemma follows, which will be used in our proof for the Stackelberg regret, in order to bound the variance of the estimated losses computed by the Grinding Algorithm. Before we proceed to it, we find it useful to define the upper, lower, and middle ground truth polytope sets. These will only be used in the analysis of our algorithm.

\textbf{Definition B.1.} In what follows, we denote by $\text{dist}(\alpha, x_t)$ the signed distance of point $x_t$ from hyperplane $\alpha$.

1. We define the middle ground truth polytope set, denoted by $\mathcal{P}_{t,GT}^m$, as the set containing the following polytopes: $p \in \mathcal{P}_{t,GT}^m : \left| \text{dist}(\alpha, x_t) \right| \leq 2\delta, \forall \alpha \in p$.

2. We define the upper ground truth polytope set, denoted by $\mathcal{P}_{t,GT}^u$, as the set containing the following polytopes: $p \in \mathcal{P}_{t,GT}^u : \text{dist}(\alpha, r_t) \geq 2\delta, \forall \alpha \in p$.

3. We define the lower ground truth polytope set, denoted by $\mathcal{P}_{t,GT}^l$, as the set containing the following polytopes: $p \in \mathcal{P}_{t,GT}^l : \text{dist}(\alpha, r_t) \leq -2\delta, \forall \alpha \in p$.

\textbf{Lemma B.3.}
\[
\mathbb{E}_{\alpha \sim A_t} \left[ \frac{1}{Pr_{\mathcal{A}_t}^m[\alpha]} \right] \leq 4 \log \left( \frac{4\lambda(A) \cdot |\mathcal{P}_{t,GT}^u \cup \mathcal{P}_{t,GT}^l|}{\gamma\lambda(p)} \right) + \lambda \left( \mathcal{P}_{t,GT}^m \right) \tag{7}
\]

\textbf{Proof.} We first analyze the term: $\mathbb{E}_{\alpha \sim A_t} \left[ \frac{1}{Pr_{\mathcal{A}_t}^m[\alpha]} \right]$ as follows:
\[
\mathbb{E}_{\alpha \sim A_t} \left[ \frac{1}{Pr_{\mathcal{A}_t}^m[\alpha]} \right] = \int_{\alpha \in A} \frac{f_{\mathcal{A}_t}(\alpha)}{Pr_{\mathcal{A}_t}^m[\alpha]} \, d\alpha = \int_{\alpha \in \mathcal{P}_{t,GT}^m} \frac{f_{\mathcal{A}_t}(\alpha)}{Pr_{\mathcal{A}_t}^m[\alpha]} \, d\alpha + \int_{\alpha \in \mathcal{P}_{t,GT}^u} \frac{f_{\mathcal{A}_t}(\alpha)}{Pr_{\mathcal{A}_t}^m[\alpha]} \, d\alpha \tag{7}
\]

where by $\alpha \in \mathcal{P}_{t,GT}^m$ we denote the integral over all actions that belong in some polytope from the set $\mathcal{P}_{t,GT}^m$. In the RHS of Equation (7), the term $\int_{\alpha \in \mathcal{P}_{t,GT}^u} \frac{f_{\mathcal{A}_t}(\alpha)}{Pr_{\mathcal{A}_t}^m[\alpha]} \, d\alpha$ is relatively easier to analyze. Due to the conservative estimates of the true middle space (i.e., the actions such that $\text{dist}(\alpha, x_t) \leq \delta$), the set of polytopes $\mathcal{P}_{t,GT}^m$ contains all the actions that belong in the true middle space, plus some other actions for which the agent could not have misreported, due to their $\delta$-boundedness. Now, for all the actions that
belong in the true middle space, it holds that they only get information (i.e., get updated) when they are chosen by the algorithm, while for the rest of the actions that have ended up in our middle space, they could have been updated by other actions as well. Thus, it holds that:

$$\forall \alpha \in \bigcup P_{l,GT}^m : \Pr_{A_t}^m[\alpha] \geq f_{A_t}(\alpha)$$

As a result:

$$\int_{\alpha \in \bigcup P_{l,GT}^m} \frac{f_{A_t}(\alpha)}{\Pr_{A_t}^m[\alpha]} d\alpha \leq \int_{\alpha \in \bigcup P_{l,GT}^m} \frac{f_{A_t}(\alpha)}{f_{A_t}(\alpha)} d\alpha = \lambda \left( P_{l,GT}^m \right)$$

(8)

Moving forward, we turn our attention to term $$\int_{\alpha \in \bigcup (P_{l,GT}^u \cup P_{l,GT}^l)} \frac{f_{A_t}(\alpha)}{\Pr_{A_t}^m[\alpha]} d\alpha$$. Assume now that an action $$\alpha$$ belongs in a polytope $$p_\alpha$$. Then, there are (weakly) more actions that can potentially update action $$\alpha$$, than the whole polytope in which it belongs, $$p_\alpha$$; indeed, in order to update the polytope, one must make sure that every action within it is updateable. As a result, $$\Pr_{A_t}^m[\alpha] \geq \Pr_{A_t}^m[p_\alpha]$$. Using this in Equation (7) we get that the first term of the right hand side of the variance is upper bounded by:

$$\sum_{p \in P_{l,GT}^u \cup P_{l,GT}^l} \int_{\alpha \in P_{l,GT}^m[p]} \frac{f_{A_t}(\alpha)}{\Pr_{A_t}^m[p]} d\alpha$$

(9)

Further, let $$\Pr_{A_t}^m[p]_{u,l}$$ be the part of $$\Pr_{A_t}^m[p]$$ that depends only in the updates that stem from actions in either the upper or the lower polytopes sets. As such: $$\Pr_{A_t}^m[p]_{u,l} \leq \Pr_{A_t}^m[p]$$ and the term in Equation (9) can be upper bounded by:

$$\sum_{p \in P_{l,GT}^u \cup P_{l,GT}^l} \frac{1}{\Pr_{A_t}^m[p]_{u,l}} \int_{\alpha \in P_{l,GT}^m[p]} f_{A_t}(\alpha) d\alpha$$

(10)

where we have also used the fact that we gain oracle access to quantity $$\Pr_{A_t}^m[p]_{u,l}$$ and therefore, we treat it as a constant in the integral. Observe now that the term $$\int_{\alpha \in P_{l,GT}^m[p]} f_{A_t}(\alpha) d\alpha$$ corresponds to the total probability that the action $$\alpha_t$$, which is chosen from the induced probability distribution $$A_t$$, belongs to polytope $$p$$, i.e., it is equal to $$\pi_t(p)$$. Hence, the term in Equation (10) can be rewritten as:

$$\sum_{p \in P_{l,GT}^u \cup P_{l,GT}^l} \frac{\pi_t(p)}{\Pr_{A_t}^m[p]_{u,l}}$$

(11)

As we have explained before, $$\pi_t(p) = 0$$, for $$p \in P_t$$ and as a result, we can disregard point-polytopes from our consideration for the rest of this proof.

We will now upper bound this term by using a graph-theoretic lemma of Alon et al. [1]. Observe now that all the actions within the upper and the lower polytopes set form a Feedback Graph as follows: each node corresponds to a polytope from one of the sets $$P_{l,GT}^u, P_{l,GT}^l$$. So the total number of nodes is at most $$|P_{l,GT}^u \cup P_{l,GT}^l|$$. Each edge $$(i, j)$$ will correspond to information passing from node $$i$$ to node $$j$$, i.e., the directed edge $$(i, j)$$ exists when polytope $$j$$ gets updated by just observing actions from the polytope $$i$$. However, for each action drawn among the upper and the lower polytopes sets, we know that the agent could not possibly misreport, and as a result, all the actions within the upper and the lower polytopes sets would be updated! As a result, the independence number of this feedback graph is $$\alpha^G = 1$$. Using the fact that each polytope $$p$$ is chosen with probability at least $$\pi_t(p) \geq \gamma \lambda(p)/\lambda(A) \geq \gamma \lambda(p)/\lambda(A)$$, where by $$\lambda(p)$$ we denote the Lebesgue measure of the smallest polytope at timestep $$t$$, the variance is upper bounded by:

$$\mathbb{E}_{\alpha_t \sim A_t} \left[ \frac{1}{\Pr_{A_t}^m[\alpha_t]} \right] \leq 4 \log \left( \frac{4 \lambda(A) \cdot |P_{l,GT}^u \cup P_{l,GT}^l|}{\lambda(p) \cdot \gamma} \right) + \lambda \left( P_{l,GT}^m \right)$$

which concludes our proof.
Let \( p = \arg \min_{\alpha \in \mathcal{A}} \lambda(p) \). Since \( \lambda(A)/\lambda(p) \) corresponds to an upper bound in the number of polytopes that generally exist, we get the following corollary.

**Corollary B.1.**

\[
\mathbb{E}_{\alpha \sim A} \left[ \frac{1}{P_{A \| \alpha}^{\mathcal{P}_T}} \right] \leq 8 \log \left( \frac{4\lambda(A)}{\gamma \lambda(p)} \right) + \lambda \left( \mathcal{P}_{t,GT}^m \right)
\]

**Remark B.1.** We remark here that the added benefit of expressing the first term of the previous proof in terms of polytopes rather than actions is that if we followed the graphs-theoretic upper bound derived by Alon et al. [1], we would have a logarithmic term in the number of nodes (i.e., the number of actions), which are infinite.

**Lemma B.4** (Second Order Regret Bound). Let \( q_1, \ldots, q_T \) be the probability distribution over the polytopes defined by Algorithm 4 for the estimated losses \( \hat{\ell}(\alpha, r_t(\alpha)), t \in [T] \). Then, the second order regret bound induced by the Grinding Algorithm is:

\[
\sum_{t=1}^{T} \sum_{p \in \mathcal{P}_{t+1}} q_t(p)\hat{\ell}(p, r_t(p)) - \sum_{t=1}^{T} \hat{\ell}(\alpha^*, r_t(\alpha^*)) \leq \sum_{t=1}^{T} \frac{\eta_t}{2} \sum_{p \in \mathcal{P}_{t+1}} q_t(p)\hat{\ell}(p, r_t(p))^2 + \frac{1}{\eta_T} \log \left( \frac{\lambda(A)}{\lambda(p)} \right)
\]

where \( p \) we denote the polytope with the smallest Lebesgue measure among all polytopes excluding the point-polytopes in the finest segmentation (i.e., after \( T \) timesteps).

Additionally, for a non-increasing sequence of \( \{\eta_t\}_{t=1}^{T} \), it holds that:

\[
\sum_{t=1}^{T} \sum_{p \in \mathcal{P}_{t+1}} q_t(p)\hat{\ell}(p, r_t(p)) - \sum_{t=1}^{T} \hat{\ell}(\alpha^*, r_t(\alpha^*)) \leq \sum_{t=1}^{T} \frac{\eta_t}{2} \sum_{p \in \mathcal{P}_{t+1}} q_t(p)\hat{\ell}(p, r_t(p))^2 + \frac{1}{\eta_T} \log \left( \frac{\lambda(A)}{\lambda(p)} \right)
\]

**Proof.** We will prove that:

\[
\sum_{t=1}^{T} \sum_{p \in \mathcal{P}_{t+1}} q_t(p)\hat{\ell}(p, r_t(p)) - \sum_{t=1}^{T} \hat{\ell}(\alpha^*, r_t(\alpha^*)) \leq \sum_{t=1}^{T} \frac{\eta_t}{2} \sum_{p \in \mathcal{P}_{t+1}} q_t(p)\hat{\ell}(p, r_t(p))^2 + \frac{1}{\eta_T} \log \left( \frac{\lambda(A)}{\lambda(p)} \right)
\]

from which Equation (12) follows naturally.

Let \( W_t = \sum_{p \in \mathcal{P}_t} w_t(p) \). We will upper and lower bound the quantity \( Q = \sum_{t=1}^{T} \log(W_{t+1}/W_t) \). First, we focus on the lower bound:

\[
Q = \sum_{t=1}^{T} \log \left( \frac{W_{t+1}}{W_t} \right) = \log \left( \frac{W_T}{W_1} \right)
\]

Observe now that in \( t = 1 \) there only exists one polytope (the whole \([-1, 1]^{d+1}\) space), and has both a total weight of \( \lambda(A) \) and a probability of 1. In other words, all the actions within this polytope have the same weight, which is equal to 1 (uniformly weighted). As a result, \( \log W_1 = \log \left( \sum_{p \in \mathcal{P}_1} \int_{\mathcal{A}} 1d\alpha \right) = \log(\lambda(A)) \).

Let us analyze the term \( \log W_T \) now:

\[
\log W_T = \log \left( \sum_{p \in \mathcal{P}_T} w_T(p) \right) = \log \left( \int_{\mathcal{A}} w_T(\alpha)d\alpha \right)
\]

\[
= \log \left( \sum_{p \in \mathcal{P}_T \setminus \mathcal{P}_T} \lambda(p) \exp \left( -\sum_{t=1}^{T} \eta_t \hat{\ell}(p, r_t(p)) \right) + \int_{\mathcal{A} \setminus \mathcal{P}_T} \exp \left( -\sum_{t=1}^{T} \eta_t \hat{\ell}(\alpha, r_t(\alpha)) \right) d\alpha \right)
\]

where the last equality is due to the fact that not further grinded polytopes have maintained the same estimated loss, \( \hat{\ell} \), for all their containing points at each timestep \( t \) and we denote by \( \mathcal{P}_T \) the set of point-polytopes contained in \( \mathcal{P}_T \).
Since the set $\mathcal{P}_T$ is essentially a set of points, thus, it has a Lebesgue measure of 0, we have that:

$$\int_{\alpha \in \bigcup \mathcal{P}_T} \exp \left( - \sum_{t=1}^T \eta_t \hat{\ell}(\alpha, r_t(\alpha)) \right) d\alpha = 0$$

Let $\alpha^* = \arg \min_{\alpha \in \mathcal{A}} \sum_{t=1}^T \hat{\ell}(\alpha, r_t(\alpha))$ (i.e., the best fixed action in hindsight among the all actions after $T$ timesteps, irrespective of whether it belongs to $\bigcup \mathcal{P}_T$ or $\bigcup \mathcal{P}_T \setminus \mathcal{P}_T$) and $p \in \mathcal{P}_T \setminus \mathcal{P}_T$ be the polytope with the smallest Lebesgue measure in $\mathcal{P}_T \setminus \mathcal{P}_T$ (i.e., excluding point-polytopes). Then, the Equation (14) becomes:

$$\log W_T \geq \log \left( \lambda(p) \exp \left( - \sum_{t=1}^T \eta_t \hat{\ell}(\alpha^*, r_t(\alpha^*)) \right) \right) = \log \left( \lambda(p) \right) - \sum_{t=1}^T \eta_t \hat{\ell}(\alpha^*, r_t(\alpha^*))$$

As a result:

$$Q = \log W_T - \log W_1 \geq \log \left( \frac{\lambda(p)}{\lambda(A)} \right) - \sum_{t=1}^T \eta_t \hat{\ell}(\alpha^*, r_t(\alpha^*)) \geq \log \left( \frac{\lambda(p)}{\lambda(A)} \right) - \sum_{t=1}^T \eta_t \hat{\ell}(\alpha, r_t(\alpha^*)) \geq \log \left( \frac{\lambda(p)}{\lambda(A)} \right) - \sum_{t=1}^T \eta_t \hat{\ell}(\alpha, r_t(\alpha))$$

since $\left\{ \eta_t \right\}_{t=1}^T$ is non-increasing.

We move on to the upper bound of $Q$ now. First, we will analyze the quantity $\log(W_{t+1}/W_t)$.

$$\log \left( \frac{W_{t+1}}{W_t} \right) = \log \left( \frac{\int_A w_t(\alpha) \exp \left( - \eta_t \hat{\ell}(\alpha, r_t(\alpha)) \right) d\alpha}{W_t} \right)$$

$$= \log \left( \frac{\int_A q_t(\alpha) \exp \left( - \eta_t \hat{\ell}(\alpha, r_t(\alpha)) \right) d\alpha}{W_t} \right)$$

$$\leq \log \left( \int_A q_t(\alpha) \left( 1 - \eta_t \hat{\ell}(\alpha, r_t(\alpha)) + \frac{\eta_t^2}{2} \hat{\ell}(\alpha, r_t(\alpha))^2 \right) d\alpha \right)$$

$$\leq \log \left( 1 - \eta_t \hat{\ell}(\alpha, r_t(\alpha))d\alpha + \int_A q_t(\alpha) \hat{\ell}(\alpha, r_t(\alpha))^2 d\alpha \right)$$

$$\leq -\eta_t \int_A q_t(\alpha) \hat{\ell}(\alpha, r_t(\alpha))d\alpha + \frac{\eta_t^2}{2} \int_A q_t(\alpha) \hat{\ell}(\alpha, r_t(\alpha))^2 d\alpha$$

Summing up for the $T$ timesteps the latter becomes:

$$\sum_{t=1}^T \log \left( \frac{W_{t+1}}{W_t} \right) \leq -\sum_{t=1}^T \eta_t \int_A q_t(\alpha) \hat{\ell}(\alpha, r_t(\alpha))d\alpha + \sum_{t=1}^T \frac{\eta_t^2}{2} \int_A q_t(\alpha) \hat{\ell}(\alpha, r_t(\alpha))^2 d\alpha$$

Combining Equations (16) and (17) we get that:

$$\sum_{t=1}^T \int_A q_t(\alpha) \hat{\ell}(\alpha, r_t(\alpha))d\alpha - \sum_{t=1}^T \hat{\ell}(\alpha^*, r_t(\alpha^*)) \leq \sum_{t=1}^T \frac{\eta_t}{2} \int_A q_t(\alpha) \hat{\ell}(\alpha, r_t(\alpha))^2 d\alpha + \frac{1}{\eta_T} \log \left( \frac{\lambda(A)}{\lambda(p)} \right)$$

We are now ready for the core proof of Theorem 11.

**Proof of Theorem 11.** We will prove the case of the unknown horizon, from which the known horizon case should be straightforward. By taking the expectation $\mathbb{E}_{\alpha \sim \mathcal{A}_t}$, in Lemma B.3.4 we get that:

$$\sum_{t=1}^T \int_A q_t(\alpha) \mathbb{E}_{\mathcal{A}_t} [\hat{\ell}(\alpha, r_t(\alpha)) ] d\alpha - \sum_{t=1}^T \int_A q_t(\alpha) \mathbb{E}_{\mathcal{A}_t} [\hat{\ell}(\alpha^*, r_t(\alpha^*)) ] \leq \sum_{t=1}^T \frac{\eta_t}{2} \int_A q_t(\alpha) \mathbb{E}_{\mathcal{A}_t} [\hat{\ell}(\alpha, r_t(\alpha))^2 ] d\alpha + \frac{1}{\eta_T} \log \left( \frac{\lambda(A)}{\lambda(p)} \right)$$

26
Combining Lemmas B.1, B.2 with the latter we get:

\[
\sum_{t=1}^{T} \int_{\mathcal{A}} q_t(\alpha) \ell(\alpha, r_t(\alpha)) \, d\alpha - \sum_{t=1}^{T} \ell(\alpha^*, r_t(\alpha^*))
\]

\[
\leq \frac{\eta}{2} \sum_{t=1}^{T} \int_{\mathcal{A}} \frac{q_t(\alpha)}{\Pr_{\mathcal{A}}[\alpha]} \, d\alpha + \frac{1}{\eta t} \log \left( \frac{\lambda(A)}{\lambda(p)} \right)
\]

\[
= \sum_{t=1}^{T} \eta_t \int_{\mathcal{A}} \frac{\pi_t(\alpha)}{\Pr_{\mathcal{A}}[\alpha]} \, d\alpha + \frac{1}{\eta t} \log \left( \frac{\lambda(A)}{\lambda(p)} \right)
\]

\[
\leq \sum_{t=1}^{T} \eta_t \left( 4 \log \left( \frac{4 \lambda(A) |\mathcal{P}_t^m \cup \mathcal{P}_t^l|}{\eta_t \cdot \lambda(p)} \right) + \lambda(\mathcal{P}_t^m) \right) + \frac{1}{\eta t} \log \left( \frac{\lambda(A)}{\lambda(p)} \right)
\]

(Lemma B.3)

Using the fact that \(\int_{\mathcal{A}} \pi_t(\alpha) \, d\alpha \leq \int_{\mathcal{A}} q_t(\alpha) \, d\alpha + \gamma_t\), the latter becomes:

\[
\mathcal{R}(T) \leq \sum_{t=1}^{T} \gamma_t + \sum_{t=1}^{T} \eta_t \left( 4 \log \left( \frac{4 \lambda(A) |\mathcal{P}_t^m \cup \mathcal{P}_t^l|}{\eta_t \cdot \lambda(p)} \right) + \lambda(\mathcal{P}_t^m) \right) + \frac{1}{\eta t} \log \left( \frac{\lambda(A)}{\lambda(p)} \right)
\]

Setting \(\gamma_t = \eta_t\):

\[
\mathcal{R}(T) \leq \sum_{t=1}^{T} \eta_t \left( 1 + 4 \log \left( \frac{4 \lambda(A) |\mathcal{P}_t^m \cup \mathcal{P}_t^l|}{\eta_t \cdot \lambda(p)} \right) + \lambda(\mathcal{P}_t^m) \right) + \frac{1}{\eta t} \log \left( \frac{\lambda(A)}{\lambda(p)} \right)
\]

which can be upper bounded as follows, since \(\eta_t \in (0, 1)\):

\[
\mathcal{R}(T) \leq \sum_{t=1}^{T} \eta_t \left( 3 \log \left( \frac{4 \lambda(A) |\mathcal{P}_t^m \cup \mathcal{P}_t^l|}{\lambda(p)} \right) + \lambda(\mathcal{P}_t^m) \right) + \frac{1}{\eta t} \log \left( \frac{\lambda(A)}{\lambda(p)} \right)
\]

(18)

Then, setting \(\eta_t\) in Equation (18) as

\[
\eta_t = \sqrt{\frac{\log \left( \frac{\lambda(A)}{\lambda(p)} \right)}{t \cdot \left( \log \left( \frac{4 \lambda(A) |\mathcal{P}_t^m \cup \mathcal{P}_t^l|}{\lambda(p)} \right) + \lambda(\mathcal{P}_t^m) \right)}}
\]

using the fact that \(\lambda(p) \leq \lambda(p_t)\), \(\forall t \in [T]\), and the property from calculus that:

\[
\sum_{t=1}^{T} \sqrt{\frac{\log x}{x}} \leq \int_{0}^{T} \sqrt{\frac{\log x}{x}} \, dx \leq \sqrt{T \log T}
\]

we get that the Stackelberg Regret is upper bounded by:

\[
\mathcal{R}(T) \leq O \left( \left\lceil \max_{t \in [T]} \left\{ 4 \log \left( \frac{4 \lambda(A) |\mathcal{P}_t^m \cup \mathcal{P}_t^l|}{\lambda(p)} \right) + \lambda(\mathcal{P}_t^m) \right\} \right\rceil \cdot \log \left( \frac{\lambda(A)}{\lambda(p)} \right) \cdot T \right)
\]

▲
Using again the fact that $\lambda(A)/\lambda(p)$ corresponds to an upper bound on the number of polytopes that generally exist, we get the following corollary for the Stackelberg regret.

**Corollary B.2.**

$$R(T) \leq O \left( \max_{t \in [T]} \left\{ 8 \log \left( \frac{4 \lambda(A)}{\lambda(p)} \right) + \lambda \left( \mathcal{P}_{t,GT}^u \right) \right\} \cdot \log \left( \frac{\lambda(A)}{\lambda(p)} \right) \cdot T \right)$$

We remark here that the data-dependent quantities of the regret’s upper bound (i.e., the Lebesgue measure of the smallest polytope $\lambda(p)$, the Lebesgue measure of the ground truth middle space $\lambda(\mathcal{P}_{t,GT}^m)$, as well as the number of polytopes in the upper and lower polytopes sets $\mathcal{P}_{t,GT}^u, \mathcal{P}_{t,GT}^l$ depend on $\delta$. We have chosen to omit this from the notation for clarity of exposition.

**Appendix C Supplementary Material for Section 5**

### C.1 Implementing the Grinding Algorithm for Continuous Action Space

In order to implement our Grinding algorithm, we used the polytope library\(^{16}\) which is part of the TuLiP python package. Other than some rounding-error fixes, we did not intervene with the core methods of the package.

In order to implement the 2-stage action draw method, we first chose a polytope (according to the probability function prescribed by the Grinding algorithm) and then, by using rejection sampling from the bounding box around the polytope, we end up choosing the action associated with it. Note that this is equivalent to the theoretical 2-stage draw.

As was also the case with the EXP3 algorithm, we lower bound the draw probabilities of each polytope by $10^{-7}$. In order to speed up our algorithm’s performance, we also used the heuristic of bounding the allowable volume of any polytope to be greater than or equal to 0.01, but in all the simulations that we ran, we saw comparable regret results even without the heuristic.

### C.2 Logistic Regression on Spammers

In this subsection, we will outline our implementation of the logistic regression algorithm on the spammers’ past data, which serves as an estimate of the in-probability for each action. For the ease of exposition, we will provide the description of the oracle for the case of a predefined action set, and subsequently, we will outline the way it generalizes to the continuous implementation.

Before we embark on this, allow us first to observe that we already have a very crude (but potentially useful) lower bound for every action $j \in A$. Indeed, each action always updates itself, and actions that always belong in the upper and lower polytope sets are always updated by all actions within these sets. The latter is due to the fact that for any hyperplane chosen within these sets, there is no possible manipulation from the perspective of the agent. Let us denote this crude lower bound for each action $j \in A$ by $c^j$.

Let us first define the notion of an admissible timestep; admissible timesteps are the ones during which the learner encounters a spammer. Everything that we mention in this paragraph uses only admissible timesteps in order to build the training set. Labels are defined as $l^j_i = 0$ if action $j$ was not updated at timestep $i$\(^{17}\) and 1 otherwise. As a first step, this oracle computes for each action $j \in A$ the probability that each action from $A$ updates $j$, by using a logistic regression\(^{18}\) with feature vectors the set $H_{1:t}$, and $L^j_{1:t}$ as the labels. Let $p^j_i, i \in A$ correspond to the output probabilities, i.e., $p^j_i$ encodes the probability that action $j$ will be

---

16\(https://github.com/tulip-control/polytope/tree/master/requirements\)

17In other words, it was within a ball of radius $2\delta$ around the action deployed at timestep $i$.

18Technically, we run a different logistic regression for every action in $A$. 

---

28
updated by action $i$. The in-probability of action $j$ is ultimately defined as:

$$\text{Pr}^i[j] = \max \left\{ \sum_{i \in A} p^i_j \pi_t[i], c^j \right\}$$

At a high-level, it is not hard to see how this can generalize to the continuous grinding case; instead of actions, one now uses whole polytopes. The implementation is significantly messier, and we leave this to the more careful readers of our publicly available code.

C.3 Different Approximation Oracle Results

In this subsection, we will present the results for our Grinding algorithm with a different approximation oracle against EXP3. The only difference between this oracle and the previously defined one is that now the admissible timesteps are all timesteps $t \in [T]$.

Using the new oracle, we see in Figure 4 that the regret performance of the Grinding algorithm becomes much better under the (more realistic) cases of $p = 0.6$ and $p = 0.8$, but we perform worse than the oracle which uses only the spammers’ data in the case that the majority of the agents that we face are strategic ($p = 0.4$). The extent to which this oracle performs worse than the oracle of Section 5 is decreased in the continuous space simulations as shown in Figure 5, potentially due to the fact that the algorithm has the opportunity to grind the space differently, and recover from some errors that the logistic regression creates.

An explanation for the fact that this oracle performs so poorly for the case that $p = 0.4$, but also, performs better than the omnipotent oracle for some of the cases with $p = 0.6, 0.8$ is that in order to achieve the best possible upper bound, the step size $\eta$ should incorporate information about the variance of the logistic regression with respect to the true in-probability function, which is currently missing.
Figure 4: Grinding vs. EXP3 for Predefined Action Set
Figure 5: Continuous Grinding vs. EXP3