Practical, Provably-Correct Interactive Learning in the Realizable Setting: The Power of True Believers

Julian Katz-Samuels
University of Wisconsin, Madison
katzsamuels@wisc.edu

Blake Mason
Rice University
bm63@rice.edu

Kevin Jamieson
University of Washington
jamieson@cs.washington.edu

Robert Nowak
University of Wisconsin, Madison
rdnowak@wisc.edu

Abstract

We consider interactive learning in the realizable setting and develop a general framework to handle problems ranging from best arm identification to active classification. We begin our investigation with the observation that agnostic algorithms *cannot* be minimax-optimal in the realizable setting. Hence, we design novel computationally efficient algorithms for the realizable setting that match the minimax lower bound up to logarithmic factors and are general-purpose, accommodating a wide variety of function classes including kernel methods, Hölder smooth functions, and convex functions. The sample complexities of our algorithms can be quantified in terms of well-known quantities like the extended teaching dimension and haystack dimension. However, unlike algorithms based directly on those combinatorial quantities, our algorithms are computationally efficient. To achieve computational efficiency, our algorithms sample from the version space using Monte Carlo “hit-and-run” algorithms instead of maintaining the version space explicitly. Our approach has two key strengths. First, it is simple, consisting of two unifying, greedy algorithms. Second, our algorithms have the capability to seamlessly leverage prior knowledge that is often available and useful in practice. In addition to our new theoretical results, we demonstrate empirically that our algorithms are competitive with Gaussian process UCB methods.

1 Introduction

In this paper, we study interactive learning where an algorithm makes a decision and observes feedback that it then uses to update its behavior. Interactive learning problems are becoming increasingly widespread in the information era. Examples include A/B/n testing where technology companies perform large-scale experiments to adaptively collect data to optimize their products on platforms like websites or smart phone applications [1]; active classification where learning algorithms adaptively collect data with the hope of learning high-quality predictive models using a much smaller number of labels than is typically required in supervised learning [2]; and environmental monitoring using sensor networks [3].

At a high-level, there are two main algorithmic paradigms for interactive learning: *agnostic* algorithms and *realizability-based* algorithms. Agnostic algorithms may use a model class $\mathcal{F}$ to guide learning, but do not assume that the true data-generating process is well-modeled by $\mathcal{F}$. Because of this, agnostic algorithms tend to have the advantages of being robust to model misspecification and noise. Due to these virtues, agnostic algorithms have received much attention in the literature on interactive learning, e.g., in active classification [4, 5, 6]. By contrast, realizability-based algorithms assume

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

that the model class $\mathcal{F}$ accurately models the real-world and leverages the structure in $\mathcal{F}$ to guide and potentially accelerate learning. Computationally efficient realizability-based algorithms have only been developed for *specific* model classes for problems like best arm identification [7, 8, 9] and regret minimization [10], and the literature lacks a general framework for developing computationally efficient minimax optimal algorithms for generic function classes in the realizable setting.

The starting point of this paper is the basic question: in the realizable setting, can agnostic algorithms compete with realizability-based algorithms? In this paper, we begin by giving a series of negative results that demonstrate that agnostic algorithms pay a significant cost for their robustness to model mispecification. As an example, we show that *any agnostic active classification algorithm* is minimax suboptimal for a class of realizable instances and thus has no hope of competing with realizability-based algorithms in the realizable setting. These results motivate us to develop *a general framework for computationally efficient and sample-efficient algorithms for generic function classes in the realizable setting*. In doing so, we solve an open problem dating back to the work of [11] on the Haystack Dimension, developing the first computationally efficient algorithm for best arm identification with generic function classes that matches the minimax lower bound up to logarithmic factors. Finally, we empirically demonstrate the generality and practicality of our new approach, `GRAILS`, for function classes ranging from vanilla kernel methods to kernel methods with side information and to the class of convex functions.

2 Problem Setup

Let $\mathcal{X}$ denote the input space and $\mathcal{Y} \subset \mathbb{R}$ the output space. We assume that $|\mathcal{Y}| < \infty$, but will relax this assumption later. Let $x_1, \ldots, x_n \in \mathcal{X}$ be a fixed pool of n measurements (or arms) with associated scores $y_1, \ldots, y_n \in \mathcal{Y}$. At each round t, the learner selects (or queries) $I_t \in [n]$ and observes y_{I_t}. We assume that the learner is given a function class $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ where $\mathcal{Y}^{\mathcal{X}}$ denotes the set of all functions mapping $\mathcal{X}$ to $\mathcal{Y}$. We say that *realizability* holds if there exists $f^* \in \mathcal{F}$ such that $f^*(x_i) = y_i$ for all $i \in [n]$. An algorithm is *realizability-based* if it assumes realizability. We focus on the noiseless setting here, but it is straightforward to extend our algorithms to handle the case where y_{I_t} is perturbed by independent, additive noise by simple repeated sampling.

We consider the following three objectives:

- **Best arm identification:** The goal is to identify $\text{argmin}_{i \in [n]} y_i$ using as few queries as possible.
- **Cumulative Loss Minimization:** The goal is to identify $\text{argmin}_{i \in [n]} y_i$ while minimizing the loss $\sum_{t=1}^{T} y_{I_t}$ incurred where T is the round that the agent identifies $\text{argmin}_{i \in [n]} f^*(x_i)$.
- **Active Classification:** The goal is to identify $\text{argmin}_{f \in \mathcal{F}} \sum_{i=1}^{n} \mathbb{1}\{y_i \neq f(x_i)\}$ using as few queries as possible.

Best-arm identification is a well-studied problem with applications ranging from clinical trials to A/B/n testing. Cumulative loss minimization is a new problem. Applications include running a clinical trial to determine which of a collection of drugs is most effective while, due to ethical concerns about giving patients ineffective drugs, minimizing the number of participants with bad outcomes. It is closely related to regret minimization where the goal is instead to minimize $\sum_{t=1}^{T_0} y_{I_t} - \min_{i \in [n]} y_i$ for a fixed time horizon T_0. Finally, active classification is a mature field whose goal is to minimize the number of labels required to learn a high-quality classifier.

Our main focus in this work is on minimax optimality for a fixed $\mathcal{F}$ in the realizable setting:

- **Best arm identification:** $\Lambda_{\textit{best}}(\mathcal{F})$ is the smallest integer q such that there exists some algorithm $\mathcal{A}$ such that for every $f^* \in \mathcal{F}$, $\mathcal{A}$ outputs an element of $\text{argmin}_{i \in [n]} f^*(x_i)$ after at most q queries.
- **Cumulative Loss Minimization:** $\Lambda_{\textit{loss}}(\mathcal{F})$ is the smallest real number q such that there exists some algorithm $\mathcal{A}$ such that for every $f^* \in \mathcal{F}$, $\mathcal{A}$ outputs an element of $\text{argmin}_{i \in [n]} f^*(x_i)$ after incurring a loss of at most q.
- **Active Classification:** $\Lambda_{\textit{class}}(\mathcal{F})$ is the smallest integer q such that there exists some algorithm $\mathcal{A}$ such that for every $f^* \in \mathcal{F}$, $\mathcal{A}$ outputs f^* after at most q queries.

We emphasize that the above notion of minimax optimality is with respect to the class $\mathcal{F}$. Next, we briefly summarize our contributions:

- **Best arm identification:** Assuming we can sample efficiently from a distribution π with support $\mathcal{F}$, we give a greedy and computationally efficient algorithm that obtains a sample complexity of $O(\log(n)\log(\frac{1}{\mathbb{P}_\pi(S_{f^*})})v^*_{best})$ where $\mathbb{P}_\pi(S_{f^*})$ is the probability of sampling f^* and v^*_{best} is a combinatorial quantity related to the extended teaching dimension and is a minimax lower bound. This is the first computationally efficient and nearly minimax optimal algorithm for best arm identification with generic function classes that matches the minimax lower bound up to logarithmic factors.
- **Loss minimization:** We propose a new algorithm that achieves a loss of $O(v^*_{loss}\log(|\mathcal{F}|))$ in the worst case where v^*_{loss} is the minimax lower bound. We show that when applied to the regret minimization setting with general function classes, our algorithm achieves a state-of-the-art regret bound that is always better than the prior state-of-the-art regret bound in [11], can be arbitrarily better, and for a large set of function classes matches the minimax lower bound up to logarithmic factors. Furthermore, using our techniques from our algorithm for best arm identification, we make this algorithm computationally efficient by leveraging a sampling oracle.
- **Active Classification:** We show that there exists a class of realizable instances such that any agnostic algorithm must query $\tilde{\Omega}(\sqrt{n})$ arms to identify the best arm, but there is a realizability-based algorithm that requires $O(\log(n))$ queries. This demonstrates an exponential gap between agnostic active classification and realizable active classification algorithms.

3 Related Work

Best arm identification has received much attention in the literature but predominantly for special classes of functions (e.g., linear) [12, 13, 7, 8, 9]. By contrast, our work concerns best arm identification with *general function classes*, which has received much less attention, and is most closely related to [11]. [11] introduce a combinatorial quantity $\text{HD}(\mathcal{F})$ known as the *haystack dimension*, and propose a greedy algorithm that achieves a sample complexity of $\text{HD}(\mathcal{F})\log(|\mathcal{F}|)$, but is not computationally efficient. We build on this work by designing a greedy algorithm that achieves computational efficiency by appealing to [14] to sample functions from the function class $\mathcal{F}$. A key technical challenge is that the version space under a greedy algorithm is not convex and therefore standard sampling algorithms like "hit and run" cannot be directly applied to it.

[15] gives a general computationally efficient algorithm that solves many pure exploration problems such as active classification and clustering. Their algorithm requires specifying a distance d between functions in the function class $\mathcal{F}$ and proceeds by shrinking the average diameter of $\mathcal{F}$ as measured by d. Unfortunately, for problems such as best arm or ϵ-good arm identification, it is unclear how to construct the appropriate distance function to achieve optimal performance or even if such a function exists for a given application.

Active Classification: There have been many works on active classification [6, 2], both in the realizable setting (e.g., [16, 17, 18]) and on agnostic algorithms (e.g., [5, 19, 4, 20]). Proposition 1 shows that agnostic algorithms are minimax suboptimal in the realizable setting, suggesting that they may never be able to match the performance of realizability-based algorithms. Our work is closely related to the problem of *Exact learning* where a learning algorithm is required to identify the true f^* with probability 1 [21, 22], in contrast to the PAC requirement where a failure probability of δ is permitted. While this requirement is certainly strong, this framework enables the design of practical algorithms, as suggested by our experiments.

Bayesian Optimization Work: Our work is also related to the vast field of Bayesian optimization. We review a few relevant theoretical results and refer readers to [3] for a thorough survey. [23] propose and analyze `GP-UCB`. Although `GP-UCB` is applied often to best arm identification problems, [23] only give a regret bound and optimality is unclear. Other notable works include [24] whose results are asymptotic and the recent paper [25], which gives a new algorithm for best arm identification for kernel bandits based on experimental design, but its optimality is unclear.

4 No Free Lunch for Agnostic Algorithms

Many agnostic algorithms have been proposed in the active classification literature (e.g. [5, 19, 4]). We say an algorithm $\mathcal{A}$ is δ*-agnostic* if for any labeling of the data $y \in \mathcal{Y}^n$, $\mathcal{A}$ finds the best classifier, argmin$_{f\in\mathcal{F}}\sum_{i=1}^n \mathbb{1}\{y \neq f(x_i)\}$, with probability at least $1-\delta$. Despite considering a much larger class of possible labelings, agnostic algorithms have been shown to achieve the lower bound in the *realizable setting* for the well-studied problem of thresholds where $\mathcal{F} = \{f_1, \ldots, f_n\}$ and $f_i(x_j) = 1$ if $j \leq i$ and $f_i(x_j) = 0$ otherwise [26]. This raises the question: *can agnostic algorithms be optimal over classes of realizable instances in general*? The following result shows that in fact the gap between agnostic algorithm and the minimax lower bound is exponential.

Proposition 1. *Let $\delta \in (0, \min(\frac{1}{20}, \frac{1}{n}))$. Consider the active classification setting. There exists $x_1, \ldots, x_n$ and $\mathcal{F}$ forming a class of instances $\mathcal{I} = \{(x_i, f(x_i))_{i=1}^n : f \in \mathcal{F}\}$ such that*

- *the expected number of samples of any δ-agnostic algorithm is $\Omega\left(\log(1/\delta)\frac{\sqrt{n}}{\log(n/\delta)}\right)$ on one of the instances in $\mathcal{I}$, and*
- *there exists a realizability-based algorithm that solves each instance in $\mathcal{I}$ in $O(\log(n))$ queries.*

It turns out agnostic algorithms are also minimax suboptimal for regret minimization, a problem closely related to loss minimization as we will discuss in more detail in Section 6. Here, we say that a regret minimization algorithm $\mathcal{A}$ is ϵ-*agnostic* if for any $y \in \mathcal{Y}^n$ such that $\min_{j\in[n]\setminus\{i_*\}} y_j - y_{i_*} \geq \epsilon$ where $i_* \in$ argmin$_{i\in[n]} y_i$, $\mathcal{A}$ suffers at most bounded regret independent of the time horizon.

Proposition 2. *There exists $\mathcal{F}$ such that any 1-agnostic algorithm suffers regret at least $O(|\mathcal{F}|)$ for some instance in $\mathcal{F}$ while there exists a realizability-based algorithm suffering regret at most $O(\log(|\mathcal{F}|))$.*

For best-arm identification, a fully agnostic algorithm must consider any $y \in \mathcal{Y}^n$ and it is therefore trivial that it would need to query every x_i. Therefore, we consider a weaker notion of agnostic algorithm. For $k \in \mathbb{N}$ and $\delta \in (0,1)$, we say an algorithm $\mathcal{A}$ is (δ, k)-*agnostic* if for any $y \in \mathcal{Y}^n$ such that $\min_{f\in\mathcal{F}}\sum_{i=1}^n \mathbb{1}\{f(x_i) \neq y_i\} \leq k$, $\mathcal{A}$ identifies argmin$_{i\in[n]} y_i$ with probability at least $1-\delta$. Despite only allowing for small amounts of mispecification, there is still an exponential gap between agnostic algorithms and the minimax lower bound.

Proposition 3. *Let $\delta \in (0, \min(\frac{1}{40}, \frac{1}{n}))$. There exists $\mathcal{F}$ such that for any algorithm $\mathcal{A}$ that is $(\delta, 1)$-agnostic with respect to $\mathcal{F}$, $\mathcal{A}$ takes $\Omega\left(\frac{n}{\log(n/\delta)}\right)$ queries in expectation on some instance in $\mathcal{F}$, while there exists a realizability-based algorithm requiring $O(1)$ samples on all instances in $\mathcal{F}$.*

Techniques: The above results rely on a novel approach for constucting instance-dependent lower bounds for the noiseless setting. The key idea is a reduction of the noiseless setting to a setting where observations are corrupted with a Gaussian random variable, that is, when the arm $i \in [n]$ is queried, the agent observes $y_i + \eta$ where $\eta \sim N(0,1)$, instead of y_i. This reduction enables the application of the transportation Lemma from the multi-armed bandit literature [27] to the noiseless setting and thereby to construct instances where agnostic algorithms *necessarily* perform poorly.

5 Efficient Algorithm for Best Arm Identification

Given the limitations of agnostic algorithms for interactive learning in the realizable setting established in the prior section, we now turn to developing *realizability-based* algorithms that are computationally efficient and match the minimax lower bound up to logarithmic factors. In this section, we examine the best arm identification problem in which the goal is to identify an $i_* \in$ argmin$_{i\in[n]} y_i$ using as few queries as possible. Practitioners may be willing to sacrifice a bit of optimality if that makes it easier to solve a problem and thus here we state some of our results and our algorithm in terms of ϵ-good arm identification, a strict generalization of best arm identification. In this problem, we are given $\epsilon \geq 0$ and the goal is to identify an ϵ-good arm, that is, an $i_* \in [n]$ such that $y_{i_*} \leq \min_{i\in[n]} y_i + \epsilon$. When $\epsilon = 0$, this reduces to best arm identification.

We begin by introducing a new quantity, inspired by the extended teaching dimension, for quantifying the difficulty of identifying an ϵ-good arm in a worst-case sense.

$$v^*_{best,\epsilon}(\mathcal{F}) = \max_{g:\mathcal{X}\mapsto\mathcal{Y}} \min_{I\subset[n]} |I|$$

$$\text{s.t. } \exists j \in [n] : \{f \in \mathcal{F} : f(x_i) = g(x_i)\, \forall i \in I\} \subset \left\{f \in \mathcal{F} : f(x_j) \leq \min_{l\in[n]} f(x_l) + \epsilon\right\}.$$

When $\epsilon = 0$, we simply write $\upsilon^*_{best}(\mathcal{F})$ instead of $\upsilon^*_{best,\epsilon}(\mathcal{F})$. We occasionally write $\upsilon^*_{best,\epsilon}$ instead of $\upsilon^*_{best,\epsilon}(\mathcal{F})$ when the context leaves no ambiguity. In words, υ^*_{best} is the minimum number of samples required so that for any function $g : \mathcal{X} \mapsto \mathcal{Y}$, there is a subset of queries of size υ^*_{best} that can make the best arm $i_* \in [n]$ unambiguous by eliminating all $f \in \mathcal{F}$ that do not put i_* as the best. The following theorem establishes υ^*_{best} as a lower bound to the optimal minimax sample complexity Λ_{best} for best arm identification.

Theorem 1. *For any $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$, $\upsilon^*_{best}(\mathcal{F}) \leq \Lambda_{best}(\mathcal{F})$.*

The setting of best arm identification in general function classes was previously studied in [11] in which the sample complexity results were quantified in terms of the Haystack dimension. Specifically, letting $\mathcal{F}' \subset \mathcal{F}$, define $\mathcal{F}'((x_i, y)) = \{f \in \mathcal{F} : f(x_i) \neq y\}$, the subset of functions in $\mathcal{F}'$ that disagree with the label y on x_i, and $\mathcal{F}'(x_i) = \{f \in \mathcal{F} : i \in \text{argmin}_{j\in[n]} f(x_j)\}$, the set of functions in $\mathcal{F}'$ that are minimized at x_i. Define $\gamma(\mathcal{F}') := \sup_{i\in[n]} \inf_{y\in\mathcal{Y}} \frac{|\mathcal{F}'(x_i)\cup\mathcal{F}'((x_i,y))|}{|\mathcal{F}'|}$. The Haystack dimension is defined as:

$$\text{HD}(\mathcal{F}) := \frac{1}{\inf_{\mathcal{F}'\subset\mathcal{F}} \gamma(\mathcal{F}')}.$$

The following Proposition shows that $\upsilon^*_{best}(\mathcal{F})$ is never significantly less than the Haystack dimension $\text{HD}(\mathcal{F})$ and is never greater than $\text{HD}(\mathcal{F})$ by more than a $O(\log(|\mathcal{F}|))$ factor.

Proposition 4. *For any $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$, $HD(\mathcal{F}) - 1 \leq \upsilon^*_{best}(\mathcal{F}) \leq cHD(\mathcal{F}) \log(|\mathcal{F}|)$, where c is a positive universal constant.*

5.1 Sampling Oracles for Efficient Realizable Active Learning

In this section we introduce the concept of a *sampling oracle*, a key tool for achieving computational efficiency. In contrast to prior active methods that enumerate an intractably large version space (e.g., [11, 21]), we instead place a measure over the version space to track its size without explicitly storing it and use sampling to approximate this measure. Let $\mathcal{R} \subset \mathbb{R}^{\mathcal{X}}$ be a set of regression functions where $\mathbb{R}^{\mathcal{X}}$ denotes the set of all functions $r : \mathcal{X} \mapsto \mathbb{R}$. Given $\mathcal{R} \subset \mathbb{R}^{\mathcal{X}}$, we say that we have access to a *sampling oracle* for $\mathcal{R}$ if there exists a distribution π on $\mathcal{R}$.

- such that we can draw $r \sim \pi$, and
- for any $\widetilde{\mathcal{R}} = \mathcal{R} \cap \mathcal{H}$ where $\mathcal{H}$ is the intersection of $\text{poly}(n, d)$ halfspaces of the form $\{x \in \mathbb{R}^n : w^\top x \leq b\}$, we can sample $r \sim \pi_{\widetilde{\mathcal{R}}}$ where for any measurable $A \subset \widetilde{\mathcal{R}}$ $\pi_{\widetilde{\mathcal{R}}}(A) = \frac{\pi(A)}{\pi(\widetilde{\mathcal{R}})}$.

By representing constraints on the version space as a set $\mathcal{H}$ in the second bullet, we can leverage sampling techniques to estimate the measure over the version space efficiently. We show in Appendix B that non-trivial sampling oracles are available for many useful function classes such as linear models, kernel methods, and convex functions. To keep the presentation simple, we assume here that we can compute probabilities exactly and show that approximation suffices in the appendix.

For $z \in \mathbb{R}$, let $\mathcal{D}[z]$ denote the $y \in \mathcal{Y}$ that is closest to z, tiebreaking by rounding down. The set of regressors $\mathcal{R}$, together with $D[\cdot]$, induces a set of discretized functions mapping $\mathcal{X}$ to $\mathcal{Y}$

$$\mathcal{F}_{\mathcal{R}} = \{\mathcal{D}[r] : r \in \mathcal{R}\}.$$

Our algorithmic approach is to use $\mathcal{F}_{\mathcal{R}}$ as our predictors and to leverage the structure of $\mathcal{R}$ to achieve computational efficiency. Our algorithm relies on *greedy volume reduction*, which requires that the $f \in \mathcal{F}$ have nonzero measure and the discretized setting ensures this. In the appendix, we extend our algorithms to the continuous setting but at the cost of a potentially suboptimal sample complexity. We make the realizability assumption that there exists $r^* \in \mathcal{R}$ such that $\mathcal{D}[r^*(x_i)] = y_i$ for all $i \in [n]$.

5.2 The Algorithm

In this section, we present the algorithm for ϵ-good arm identification, a strict generalization of best arm identification. `GRAILS` (see Algorithm 1) implicitly maintains a version space $\mathcal{R}_t$ at each round over $\mathcal{R}$ and thus we define

$$\mathcal{R}_t(x_i, y) = \{r \in \mathcal{R}_t : \mathcal{D}[r(x_i)] \neq y\},\ \mathcal{R}_{t,\epsilon}(x_i) = \{r \in \mathcal{R}_t : \mathcal{D}[r(x_i)] \leq \min_{j\in[n]} \mathcal{D}[r(x_j)] + \epsilon\},$$

```
$P_1 \longleftarrow \pi, \mathcal{R}_1 \longleftarrow \mathcal{R}, k \longleftarrow 1, t_1 = 1, O_1 \longleftarrow \emptyset$;
for $t = 1, 2, \dots$ do
    Let $I_t \in \arg\max_{i \in [n] \setminus O_t} \min_{y \in \mathcal{Y}} \mathbb{P}_{r \sim P_k}(r \in \mathcal{R}_{t,\epsilon}(x_i) \cup \mathcal{R}_t(x_i, y))$;
    Query $x_{I_t}$ and observe $y_{I_t}$ and set $O_{t+1} \longleftarrow O_t \cup \{I_t\}$ ;
    Let $\mathcal{R}_{t+1} \longleftarrow \mathcal{R}_t \setminus (\mathcal{R}_{t,\epsilon}(x_{I_t}) \cup \mathcal{R}_t(x_{I_t}, y_{I_t}))$;
    if $\mathbb{P}_{r \sim P_k}(r \in \mathcal{R}_{t+1}) \leq \frac{1}{2n}$ then
        $k \longleftarrow k + 1$;
        $t_k \longleftarrow t + 1, P_{k+1} \longleftarrow \frac{1}{n - |O_{t_k}|} \sum_{l \in [n] \setminus O_{t_k}} \pi_{C_l(O_{t_k})}$;
    if STOP($\mathcal{F}_{\mathcal{R}}$,$O_t$) then
        return $\text{argmin}_{i \in O_t} y_i$
```

Algorithm 1: GRAILS (GReedy Algorithm for Interactive Learning using Sampling)

defined analogously to the set $\mathcal{F}'(x_i, y)$ and $\mathcal{F}'(x_i)$ above. $\mathcal{R}_t(x_i, y)$ consists of functions in the version space that are inconsistent with the observation (x_i, y) and $\mathcal{R}_{t,\epsilon}(x_i)$ is the set of functions in the version space for which x_i is an ϵ-good arm and therefore no longer need to be considered. When $\epsilon = 0$, we write $\mathcal{R}_t(x_i)$ instead of $\mathcal{R}_{t,\epsilon}(x_i)$.

Algorithm 1 takes the greedy approach. At each round t, our algorithm queries a point $I_t \in [n]$ that maximizes the measure of the functions removed from the version space under a distribution P_k, namely $\mathbb{P}_{r \sim P_k}(r \in \mathcal{R}_t(x_i) \cup \mathcal{R}_{t,\epsilon}((x_i, y))$, for the worst case observation $y \in \mathcal{Y}$. Once the algorithm queries I_t, it adds I_t to O_t, the set of observations up to time t.

Now, we describe our sampling approach. To keep the presentation simple and to capture the main ideas, we assume here $\mathcal{R}$ is convex. Define $\texttt{better}_\epsilon = \max_{y < \min_{i \in O_t} y_i - \epsilon} y$, which is well-defined since $\mathcal{Y}$ is discrete. Note that $r \in \mathcal{R}_t$ only if there exists $l \in [n] \setminus O_t$ such that $\mathcal{D}[r(x_l)] \leq \texttt{better}_\epsilon$. We may decompose the version space $\mathcal{R}_t$ as the union of $O(n)$ sets:

$$\mathcal{R}_t = \cup_{l \in [n] \setminus O_t} \{r \in \mathcal{R} : \mathcal{D}[r(x_i)] = y_i \forall i \in O_t\} \cap \{\mathcal{D}[r(x_l)] \leq \texttt{better}_\epsilon\} = \cup_{l \in [n] \setminus O_t} C_l(O_t)$$

where

$$C_l(O_t) := \{r \in \mathcal{R} : \mathcal{D}[r(x_i)] = y_i \forall i \in O_t\} \cap \{\mathcal{D}[r(x_l)] \leq \texttt{better}_\epsilon\}.$$

Each set $C_l(O_t)$ is an intersection of $\mathcal{R}$ with $O(n)$ halfspaces and is a convex set. Unfortunately, a union of convex sets need not be convex so one cannot hope to directly apply algorithms like hit-and-run to efficiently sample from $\mathcal{R}_t$. To overcome this, the algorithm samples from a mixture with each component supported on $C_l(O_t)$, a convex set. Operating in stages, in stage k, it puts the measure P_k over the remaining functions in the version spaces where

$$P_k := \frac{1}{n - |O_t|} \sum_{l \in [n] \setminus O_t} \pi_{C_l(O_t)}.$$

Sampling from the mixture P_k is a key algorithmic innovation in this work and enables efficient sampling from a non-convex version space.

We make two final remarks. First, STOP($\mathcal{F}_{\mathcal{R}}$,$O_t$) is a subroutine for terminating the algorithm. It essentially checks whether the version space $\mathcal{R}_t$ is empty by checking whether each of the $C_l(O_t)$ sets is feasible. For many function classes of interest such as linear models, kernel methods, and the class of convex functions, this can be formulated as a convex feasibility problem (see the Appendix for a concrete instance). Second, in practice, one may not know the true model class precisely apriori, but it is straightforward to do model selection through the standard doubling technique. For example, for the class of Lipschitz functions, one may not know the true Lipschitz constant. In these situations, one can apply a standard doubling trick on the Lipschitz constant.

5.3 The Upper Bound

We now introduce some notation necessary to state our upper bound. Define for every $f \in \mathcal{F}_{\mathcal{R}}$ the set

$$S_f = \{r \in \mathcal{R} : \mathcal{D}[r(x_i)] = f(x_i) \forall i \in [n]\}.$$

The sets S_f induce a partition of $\mathcal{R}$.

Theorem 2. *Fix $\mathcal{R}$. Let $\epsilon \geq 0$. There exists a universal constant $c > 0$ such that if t is greater than*

$$c\upsilon^*_{best,\epsilon}(\mathcal{F}_\mathcal{R}) \log(\frac{1}{\mathbb{P}_\pi(S_{f^*})}) \log(n),$$

then Algorithm 1 has pulled an arm I_s at some $s \leq t$ such that $y_{I_s} \leq \min_{i \in [n]} y_i + \epsilon$.

When $\epsilon = 0$, Algorithm 1 and Theorem 2 together solve the open problem from [11] of developing a computationally efficient optimal algorithm for best arm identification for generic function classes that matches the minimax lower bound up to logarithmic factors.

Comparison to prior work: [11] proposes a *computationally inefficient* algorithm for best arm identification that obtains a sample complexity of $O(\text{HD}(\mathcal{F}_\mathcal{R}) \log(|\mathcal{F}_\mathcal{R}|))$. By Proposition 4, $\upsilon^*_{best}(\mathcal{F}_\mathcal{R})$ is upper bounded by $\text{HD}(\mathcal{F}_\mathcal{R}) \log(|\mathcal{F}_\mathcal{R}|)$, and thus our sample complexity is loose by a factor of $\log(\frac{1}{\mathbb{P}_\pi(S_{f^*})}) \log(n)$. Although our bound is indeed looser than the bound in [11], we note that computationally efficient algorithms for other active learning problems that match the minimax lower bound up to logarithmic factors have a similar logarithmic dependence on the inverse probability of sampling the true function [18, 15]. Thus, it is an important open question whether it is possible to develop computationally efficient and nearly minimax optimal algorithms for active learning that weaken or remove the dependence on $\log(\frac{1}{\mathbb{P}_\pi(S_{f^*})})$ in their sample complexity.

We close this section with a simple instance of best arm identification in a linear function class that provides an easy instantiation of our upper bound.

Proposition 5. *Let $x_1, \ldots, x_n \in \mathbb{R}^{n+1}$ such that $x_i = \frac{i}{n} e_1 + 10 \cdot e_{i+1}$. Let $\mathcal{R}_i = \{r(v) = v_1 - b + v_{i+1} : b \in \frac{1}{n}[i, i+1)\}$ where v_i denotes the ith entry of v and $\mathcal{R} = \cup_{i=1}^n \mathcal{R}_i$. Define $\mathcal{Y} = \{0, 1, 10\}$. Let π be a uniform distribution over $[0, 1]$. Fix $f^* \in \mathcal{F}_\mathcal{R}$. Then, `GRAILS` returns the best arm in $\upsilon^*_{best}(\mathcal{F}_\mathcal{R}) \log(\frac{1}{\mathbb{P}_\pi(S_{f^*})}) \log(n) \leq O(\log(n)^2)$ samples.*

By contrast, any non-adaptive algorithm would require $\Omega(n)$ samples for the above function class. Furthermore, we suspect that the additional logarithmic factor stemming from Theorem 2 is an artifact of the analysis. We discuss other instances in the appendix.

6 An Efficient Algorithm for Cumulative Loss Minimization

In this Section, we consider the task of loss minimization. To interpret each y_i as a loss, we assume $\min_{y \in \mathcal{Y}} y \geq 0$. Recall that in this setting when the learner identifies an element $i^* \in \text{argmin}_{i \in [n]} f^*(x_i)$, she declares that the game is over. We stress that the learner can identify an element $i^* \in \text{argmin}_{i \in [n]} f^*(x_i)$ by eliminating all $f \in \mathcal{F}$ such that $i^* \notin \arg\min_{i \in [n]} f(x_i)$.

To begin, we introduce the following novel quantity for quantifying the worst-case difficulty of cumulative loss minimization, inspired by the extended teaching dimension:

$$\begin{aligned} \upsilon^*_{loss}(\mathcal{F}) := & \max_{g:\mathcal{X} \mapsto \mathcal{Y}} \min_{I \subset [n]} \min_{\widetilde{I} \subset I : |\widetilde{I}| = |I| - 1} \sum_{i \in \widetilde{I}} g(x_i) \\ & \text{s.t. } \exists j \in [n] : \{f \in \mathcal{F} : f(x_i) = g(x_i) \, \forall i \in [I]\} \subset \{f \in \mathcal{F} : j \in \text{argmin}_{i \in [n]} f(x_i)\}. \end{aligned}$$

We occasionally write υ^*_{loss} instead of $\upsilon^*_{loss}(\mathcal{F})$ when the context leaves no ambiguity. In words, υ^*_{loss} is the loss that must be incurred (up to the penultimate round) for some scoring function $g : \mathcal{X} \mapsto \mathcal{Y}$ in order to identify the best arm. Next, we show that υ^*_{loss} is a minimax lower bound.

Theorem 3. *For all $\mathcal{F} \subset \mathcal{Y}^\mathcal{X}$, $\upsilon^*_{loss}(\mathcal{F}) \leq \Lambda_{loss}(\mathcal{F})$.*

The following Proposition gives an upper bound of υ^*_{loss} in terms of υ^*_{best}.

Proposition 6. *For all $\mathcal{F} \subset \mathcal{Y}^\mathcal{X}$, $\upsilon^*_{loss}(\mathcal{F}) \leq \upsilon^*_{best}(\mathcal{F}) max_{y \in \mathcal{Y}} y$.*

In words, this Proposition reflects that one strategy for cumulative loss minimization is to minimize the number of queries to identify the best arm, ignoring the losses.

```
$P_1 \longleftarrow \pi, \mathcal{R}_1 \longleftarrow \mathcal{R}, k \longleftarrow 1, t_1 = 1, O_1 \longleftarrow \emptyset$;
for $t = 1, 2, \ldots$ do
    Let $I_t \in \text{argmin}_{i \in [n] \setminus O_t} \text{max}_{y \in \mathcal{Y}} \frac{y}{\mathbb{P}_{r \sim P_k}(r \in \mathcal{R}_t(x_i) \cup \mathcal{R}_t((x_i, y)))}$;
    Query $x_{I_t}$ and observe $y_{I_t}$ and set $O_{t+1} \longleftarrow O_t \cup \{I_t\}$ ;
    Let $\mathcal{R}_{t+1} \longleftarrow \mathcal{R}_t \setminus (\mathcal{R}_t(x_{I_t}) \cup \mathcal{R}_t((x_{I_t}, y_{I_t})))$;
    if $\mathbb{P}_{r \sim P_k}(r \in \mathcal{R}_{t+1}) \leq \frac{1}{2n}$ then
        $k \longleftarrow k + 1$;
        $t_k \longleftarrow t + 1, P_{k+1} \longleftarrow \frac{1}{n - |O_{t_k}|} \sum_{l \in [n] \setminus O_{t_k}} \pi_{C_l(O_{t_k})}$;
    if STOP($\mathcal{F}_{\mathcal{R}}$,$O_t$) then
        return $\text{argmin}_{i \in O_t} y_i$
```

Algorithm 2: GRAILS for Loss Minimization

6.1 The Algorithm and Upper Bound

Algorithm 2 is similar to Algorithm 1: it is greedy and operates in phases, sampling from the mixture P_k in the kth phase due to the nonconvexity of the version space. The main difference is the objective for selecting I_t: it queries the arm I_t that for the worst case $y \in \mathcal{Y}$ minimizes the ratio of the loss incurred in round t and the volume of the functions removed under the measure P_k. This objective is inspired by the greedy algorithm for weighted set cover [28].

Define $S_{\min} = \text{argmin}_{S_f : f \in \mathcal{F}_{\mathcal{R}}, \mathbb{P}_\pi(S_f) > 0} \mathbb{P}_\pi(S_f)$. $S_{\min}$ is the subset of $\mathcal{F}_{\mathcal{R}}$ in the partition that has the least nonzero probability under π. Recall $\mathcal{R}$ is a set of regression functions.

Theorem 4. *Fix $\mathcal{R}$. Algorithm 2 identifies $\text{argmin}_{i \in [n]} f^*(x_i)$ after incurring a loss of at most*

$$\widetilde{O}(\Lambda_{\mathit{loss}}(\mathcal{F}_{\mathcal{R}})) = 2(v^*_{\mathit{loss}}(\mathcal{F}_{\mathcal{R}}) + \mathit{max}_{i \in [n], f \in \mathcal{F}_{\mathcal{R}}} f(x_i)) \log(n) \log\left(\frac{1}{\mathbb{P}(S_{\mathit{min}})}\right).$$

The above guarantee is optimal up to a multiplicative factor of $\log(n) \log(\frac{1}{\mathbb{P}(S_{\min})})$ and an additive factor of $\max_{i \in [n], f \in \mathcal{F}} f(x_i)$, which is typically of lower order. We also give an algorithm that enumerates the function class and therefore is not efficient when $|\mathcal{F}_{\mathcal{R}}|$ is exponential in problem-dependent parameters, but has a stronger guarantee. Due to space constraints and its similarity to Algorithm 2, we defer its presentation to the supplementary material and provide the result here.

Theorem 5. *Fix $\mathcal{F}$. Algorithm 2 identifies $\text{argmin}_{i \in [n]} f^*(x_i)$ after incurring a loss of at most*

$$\widetilde{O}(\Lambda_{\mathit{loss}}(\mathcal{F})) = 2(v^*_{\mathit{loss}}(\mathcal{F}) + \mathit{max}_{i \in [n], f \in \mathcal{F}} f(x_i)) \log(|\mathcal{F}|).$$

Theorems 4 and 5 are the first algorithms for cumulative loss minimization that match the minimax lower bound up to logarithmic factors. As a corollary, we obtain results for *regret minimization*, in which the goal is to identify $\text{argmin}_{i \in [n]} f^*(x_i)$ while minimizing the regret $\sum_{t=1}^{T} f^*(x_{I_t}) - \min_{i \in [n]} f^*(x_i)$ incurred where T is the round that the agent identifies $\text{argmin}_{i \in [n]} f^*(x_i)$.

Remark 1. *If $\min_j f(x_j) = 0$ for all $f \in \mathcal{F}$, then loss minimization is equivalent to regret minimization. In this case, Theorem 5 gives nearly optimal minimax bounds for the regret minimization metric. Furthermore, if $\min_j f(x_j) = \min_j f'(x_j) =:$ opt for all $f, f' \in \mathcal{F}$, then by subtracting opt from each f, we reduce to the setting where $\min_j f(x_j) = 0$ for all $f \in \mathcal{F}$ and obtain a nearly optimal minimax bound.*

To the best of our knowledge, *these are the first regret bounds that match the minimax lower bound up to logarithmic factors for a large and general class of function classes.*

Comparison to [11]. Next, we compare our algorithm in Theorem 5 to the *computationally inefficient* algorithm from [11] for regret minimization in the noiseless setting. Their regret scales as $O(\Delta_{\max} \text{HD}(\mathcal{F}) \ln(|\mathcal{F}|))$, ignoring lower order terms. The following Proposition shows that our cumulative regret bound is never worse than theirs by more than a polylogarithmic factor. Let $y_* = \text{argmin}_{y \in \mathcal{Y}} y$ and $\Delta_{\max} = \max_{y \in \mathcal{Y}} y - y_*$ and $\Delta_{\min} = \min_{y \in \mathcal{Y} : y \neq y_*} y - y_*$.

Proposition 7. *Let $\mathcal{F}$ such that $v^*_{\mathit{best}}(\mathcal{F}) \geq 1$. There exists a universal constant $c > 0$ such that for large enough T_0, if Algorithm 3 is run for T_0 rounds, the regret of Algorithm 3 is bounded above by*

$$c v^*_{\mathit{best}}(\mathcal{F}) \ln(|\mathcal{F}|) \Delta_{\mathit{max}} \leq c \mathit{HD}(\mathcal{F}) \ln(|\mathcal{F}|)^2 \Delta_{\mathit{max}}.$$

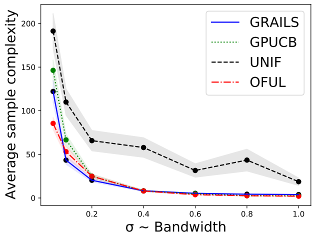

Figure 1: Varying σ

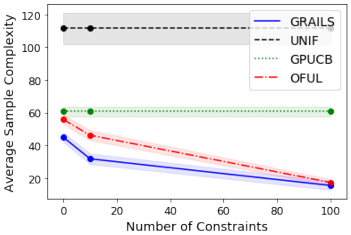

Figure 2: Prior Knowledge

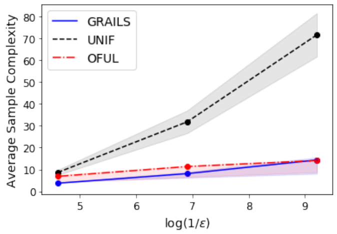

Figure 3: Convex Functions

The above Proposition says that if the horizon is long enough, then the cumulative regret in Theorem 5 is never that much worse than the cumulative regret of the algorithm in [11].

On the other hand, there exists an instance where the regret of our Algorithm in Theorem 5 is significantly better than the regret of the algorithm in [11].

Proposition 8. *For any $\xi > 0$, there exists an instance with $\Delta_{max} \geq \Delta_{\min} + \xi$ where (ignoring logarithmic factors) the regret minimization algorithm from [11] obtains a regret of at least $\Delta_{max} n/2$ while the guarantee in Theorem 5 is $n\Delta_{\min} + \Delta_{max}$.*

Thus, the regret of [11] scales as $\Delta_{\max} n$, while the regret of Algorithm 3 scales as $\Delta_{\min} n$, and the gap between $\Delta_{\max}$ and $\Delta_{\min}$ can be made aribtrarily large. *The key advantage of our algorithm over the work of [11] is that their algorithm is an explore-and-then-commit algorithm, ignoring the cost of information gain, whereas our algorithm Algorithm 3 (see Appendix) is cost-aware weighing the tradeoff between information gain and loss incurred.*

Relation to Regret Minimization. We conclude this section by discussing the relationship between loss minimization and regret minimization. Our first observation is that any minimax-optimal regret-minimizing algorithm is also a minimax-optimal loss-minimizing algorithm, as shown by the following Proposition.

Proposition 9. *Fix $T_0 \in \mathbb{N}$. Let $loss(\mathcal{A}; f; T_0) = \sum_{t=1}^{T_0} f(x_{I_t})$ and $regret(\mathcal{A}; f; T_0) = \sum_{t=1}^{T_0} f(x_{I_t}) - \min_{j\in[n]} f(x_j)$ denote the loss and regret incurred by an algorithm $\mathcal{A}$ over T_0 rounds. Let $\bar{\mathcal{A}}$ be an algorithm such that $max_{f\in\mathcal{F}} regret(\bar{\mathcal{A}}; f; T_0) \leq c \min_{\mathcal{A}} max_{f\in\mathcal{F}} regret(\mathcal{A}; f; T_0)$. Then,*

$$max_{f\in\mathcal{F}} loss(\bar{\mathcal{A}}; f; T_0) \leq (c+1) \min_{\mathcal{A}} max_{f\in\mathcal{F}} loss(\mathcal{A}; f; T_0).$$

On the other hand, a minimax optimal algorithm for loss minimization can have regret that is arbtrarily worse than the minimax lower bound, implying that regret minimization subsumes loss minimization.

Proposition 10. *For any $\xi > 0$, there exists $\mathcal{F}$ such that the minimax regret is 1, but a loss minimizing algorithm obtains a regret of ξ.*

As discussed above, *we provide the first upper bounds for loss minimization that match the minimax lower bound up to logarithmic factors, as well as state-of-the-art results for regret minimization with general function classes.*

7 Experiments

In this Section, we present experiments comparing `GRAILS` to uniform sampling (`UNIF`), `GP-UCB` [23], and `OFUL` [29]. We consider a version of `GRAILS` that is designed for continuous output spaces, and which we present and analyze in the appendix. For `OFUL`, we construct the lower confidence bound by solving a constrained optimization problem based on the prior feedback and the function class structure. Our algorithm outperforms the other algorithms and demonstrates the ability to seamlessly incorporate prior knowledge to accelerate learning, and works for function classes ranging from kernel methods to convex functions. While `OFUL` tends to perform only slightly worse than `GRAILS`, unlike `GRAILS`, *it does not have general theoretical guarantees* that can handle the incorporation of constraints given by prior knowledge or specially structured spaces like the set of convex functions, and therefore we consider it a strong, if heuristic, baseline. *Indeed, in the appendix, we show* *that* `OFUL` *is minimax suboptimal for some function classes.* Due to space constraints, we provide additional implementation details in the appendix.

Varying σ. In Figure 1, we plot the average number of samples necessary to achieve a simple regret of less than 0.01 with the shaded region showing 1 standard error taken over 36 independent trials. We generated random 2d functions living in a RBF Reproducing Kernel Hilbert Space (RKHS) with parameter σ varying. We evaluate the function at 400 points taken in a grid of $[0,1]^2$. As σ decreases and the effective dimension increases, the performance of all methods becomes similar to uniform sampling. Conversely, when σ is large and the dimension decreases, the active methods improve markedly upon uniform sampling. In all cases, `GRAILS` matches or slightly exceeds the performance of the baseline methods.

Kernel Methods with Prior Knowledge: In many practical settings, prior knowledge is available to practitioners, e.g., of the form $f^*(x) \geq f^*(x')$ for some pairs $x, x' \in X$. It is straightforward to incorporate into our algorithm any prior knowledge that can be expressed in terms of a polyhedron in the output space $\mathbb{R}^n$. By contrast, while there are a variety of approaches to incorporate constraints into GP regression methods [30], it is not immediately clear how to use constrained GP models in adaptive sampling methods. In this experiment, we use $x_1, \ldots, x_n \in [0,6]$ equally spaced with $n = 250$. We use the Gaussian RBF kernel with $\sigma = 0.075$. For each of the 60 trials, we draw a random function from the RKHS. We vary the number of random pairwise constraints given to `GRAILS` and `OFUL`. Figure 2 depicts the average number of samples required to achieve a simple regret of 0.005 and shows that as `GRAILS` and `OFUL` obtain more prior knowledge, their performance improves.

Convex Functions: In this experiment, the algorithms use the function class $\mathcal{F}$ consisting of all convex functions. We generate 300 points in an equally spaced grid on the interval $[0,1]$. $f^*(x) = 5(x - x_{\min})^2$ where for each of the 30 trials $x_{\min}$ is drawn uniformly at random from $[0,1]$. We only consider `GRAILS`, `OFUL`, and `UNIF`. Figure 3 depicts the average number of samples that each algorithm uses to obtain an ϵ-accurate solution, showing that `GRAILS` does slightly better than `OFUL`, while, as expected, the sample complexity of `UNIF` blows up as ϵ decreases.

8 Conclusion

Our work leaves many open questions. First, while `GRAILS` is computationally efficient as evidenced by our experiments, it becomes computationally burdensome as n grows very large. Further work is required to scale `GRAILS` to the regime where n is extremely large (e.g., $n \approx 10{,}000$). Second, we have designed computationally efficient algorithms that match the minimax lower bound up to logarithmic factors in the *noiseless setting* and it remains an open question how to achieve near minimax optimality in the noisy setting. Finally, it is an important future direction to develop computationally efficient and near minimax optimal algorithms in the continuous output setting.

Acknowledgments and Disclosure of Funding

Kevin Jamieson was supported in part by the NSF TRIPODS II grant DMS 2023166. This work was partially supported by AFOSR grant FA9550-18-1-0166.

References

[1] Kevin Jamieson and Lalit Jain. A bandit approach to multiple testing with false discovery control. *arXiv preprint arXiv:1809.02235*, 2018.

[2] Burr Settles. Active learning literature survey. 2009.

[3] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[4] Sanjoy Dasgupta, Daniel J Hsu, and Claire Monteleoni. *A general agnostic active learning algorithm*. Citeseer, 2007.

[5] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89, 2009.

[6] Steve Hanneke et al. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309, 2014.

[7] Marta Soare, Alessandro Lazaric, and Rémi Munos. Best-arm identification in linear bandits. In *Advances in Neural Information Processing Systems*, pages 828–836, 2014.

[8] Tanner Fiez, Lalit Jain, Kevin G Jamieson, and Lillian Ratliff. Sequential experimental design for transductive linear bandits. In *Advances in Neural Information Processing Systems*, pages 10666–10676, 2019.

[9] Julian Katz-Samuels, Lalit Jain, Kevin G Jamieson, et al. An empirical process approach to the union bound: Practical algorithms for combinatorial and linear bandits. *Advances in Neural Information Processing Systems*, 33, 2020.

[10] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

[11] Kareem Amin, Michael Kearns, and Umar Syed. Bandits, query learning, and the haystack dimension. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 87–106. JMLR Workshop and Conference Proceedings, 2011.

[12] Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil'ucb: An optimal exploration algorithm for multi-armed bandits. In *Conference on Learning Theory*, pages 423–439, 2014.

[13] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1):1–42, 2016.

[14] László Lovász and Santosh Vempala. Fast algorithms for logconcave functions: Sampling, rounding, integration and optimization. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 57–68. IEEE, 2006.

[15] Christopher Tosh and Daniel Hsu. Diameter-based interactive structure discovery. In *International Conference on Artificial Intelligence and Statistics*, pages 580–590. PMLR, 2020.

[16] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine learning*, 15(2):201–221, 1994.

[17] Sanjoy Dasgupta. Coarse sample complexity bounds for active learning. In *NIPS*, volume 18, pages 235–242, 2005.

[18] Christopher Tosh and Sanjoy Dasgupta. Diameter-based active learning. In *International Conference on Machine Learning*, pages 3444–3452. PMLR, 2017.

[19] Alina Beygelzimer, Daniel Hsu, John Langford, and Tong Zhang. Agnostic active learning without constraints. *arXiv preprint arXiv:1006.2588*, 2010.

[20] Julian Katz-Samuels, Jifan Zhang, Lalit Jain, and Kevin Jamieson. Improved algorithms for agnostic pool-based active classification. *International Conference on Machine Learning*, 2021.

[21] Tibor Hegedus. Generalized teaching dimensions and the query complexity of learning. In *Proceedings of the eighth annual conference on Computational learning theory*, pages 108–117, 1995.

[22] José L Balcázar, Jorge Castro, and David Guijarro. A new abstract combinatorial dimension for exact learning via queries. *Journal of Computer and System Sciences*, 64(1):2–21, 2002.

[23] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[24] Adam D Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(10), 2011.

[25] Romain Camilleri, Julian Katz-Samuels, and Kevin Jamieson. High-dimensional experimental design and kernel bandits. *International Conference on Machine Learning*, 2021.

[26] Lalit Jain and Kevin G Jamieson. A new perspective on pool-based active classification and false-discovery control. In *Advances in Neural Information Processing Systems*, pages 13992–14003, 2019.

[27] Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory*, pages 998–1027, 2016.

[28] David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.

[29] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *NIPS*, volume 11, pages 2312–2320, 2011.

[30] Laura P Swiler, Mamikon Gulian, Ari L Frankel, Cosmin Safta, and John D Jakeman. A survey of constrained gaussian process regression: Approaches and implementation challenges. *Journal of Machine Learning for Modeling and Computing*, 1(2), 2020.

[31] Tor Lattimore and Csaba Szepesvari. The end of optimism? an asymptotic analysis of finite-armed linear bandits. In *Artificial Intelligence and Statistics*, pages 728–737. PMLR, 2017.

[32] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer, 2001.

[33] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[34] Julian Katz-Samuels and Kevin Jamieson. The true sample complexity of identifying good arms. In *International Conference on Artificial Intelligence and Statistics*, pages 1781–1791. PMLR, 2020.

[35] Julian Katz-Samuels and Clay Scott. Feasible arm identification. In *International Conference on Machine Learning*, pages 2535–2543. PMLR, 2018.

[36] Julian Katz-Samuels and Clayton Scott. Top feasible arm identification. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1593–1601. PMLR, 2019.

Contents

A Suboptimality of Agnostic Algorithms Proofs

A.1 Active Classification

Proof of Proposition 1. Let $m \in \mathbb{N}$ and $\ell = m + 1$. Note $n = m\ell + m$. Define for all $i \in [m]$

$$f_i(x_j) = \begin{cases} 1 & j \in \{\ell(i-1)+1, \dots, i\ell\} \cup \{m\ell+1, \dots, m\ell+i\} \\ -1 & \text{o/w} \end{cases}$$

Further, define $f_0(x_i) = -1$ for all $i \in [n]$. Let $\mathcal{F} = \{f_0, f_1, \dots, f_m\}$. This instance essentially couples disjoint sets and thresholds.

Suppose that $y_i = f_0(x_i)$ for all i, so that $f_0 = f^*$ (and note that the realizability assumption holds).

Note that if an algorithm assumes realizability, then it suffices to perform binary search on $x_{ml+1}, \dots, x_{m(l+1)}$ (which is a thresholds instance) and this procedure terminates and outputs f_0 after $O(\log(m))$ queries.

Next, we provide a lower bound for any δ-agnostic algorithm wrt active classification on this instance. By corollary 1, any δ-agnostic algorithm wrt active classification requires

$$\frac{\log(1/\delta)}{\log(n/\delta)}\phi^*$$

samples in expectation, so it suffices to lower bound ϕ^*. Define the alternative instances for all $i \in [m]$.

$$\bar{y}_j^{(i)} = \begin{cases} 1 & \text{if } j \in \{\ell(i-1)+1, \dots, i\ell\} \\ y_j & \text{if otherwise} \end{cases}$$

Note that since $\ell = m + 1$, we have that

$$\sum_{i=1}^{n} \mathbb{1}\{f_j(x_i) \neq \bar{y}_i^{(j)}\} = j < m + 1 = \sum_{i=1}^{n} \mathbb{1}\{f_0(x_i) \neq \bar{y}_i^{(j)}\}$$

and thus $\bar{y}^{(j)}$ is an alternative instance. Therefore, we have that

$$\begin{aligned} \phi^* &= \min_{\lambda} \max\nolimits_{\tilde{y} \in \{-1,1\}^n : \text{argmin}_{f \in \mathcal{F}} \sum_{i=1}^n \mathbb{1}\{\tilde{y}_i \neq f(x_i)} \frac{1}{\sum_{i=1}^n \lambda_i \mathbb{1}\{y_i \neq \tilde{y}_i\}} \\ &\geq \min_{\lambda} \max\nolimits_{i \in [m]} \frac{1}{\sum_{j \in \{\ell(i-1)+1, \dots, i\ell\}} \lambda_j} \end{aligned}$$

The minimizing λ puts $\frac{1}{\sum_{j \in \{\ell(i-1)+1, \dots, i\ell\}} \lambda_j} = C$ for all $i \in [m]$ for some constant $C \in \mathbb{R}$. We also have that

$$1 = \sum_{i=1}^{m\ell} \lambda_i = \sum_{i=1}^{m} \sum_{j \in \{\ell(i-1)+1, \dots, i\ell\}} \lambda_j = \sum_{i=1}^{m} \frac{1}{C} = \frac{m}{C}$$

Therefore, we have shown that $\phi^* \geq m = \Omega(\sqrt{n})$. This completes the proof.

□

Recall that we say an algorithm $\mathcal{A}$ is δ-agnostic wrt $\mathcal{F}$ if for any $y \in \mathcal{Y}^n$, $\mathcal{A}$ outputs $\text{argmin}_{f \in \mathcal{F}} \sum_{i=1}^n \mathbb{1}\{y_i \neq f(x_i)\}$ with probability at least $1 - \delta$.

We define the *Gaussian Active Classification problem* as follows: when arm i is pulled, $y_i + \eta$ where $\eta \sim N(0, 1)$ is observed. The goal is to identify $\text{argmin}_{f \in \mathcal{F}} \sum_{i=1}^n \mathbb{1}\{y_i \neq f(x_i)\}$ We call a pair $(y, \mathcal{F})$ where $y \in \{-1, 1\}^n$ an *instance*.

Proposition 11. *Let $\delta \in (0, \frac{1}{n})$. Fix $\mathcal{Y} = \{-1, 1\}$, $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$, and $y \in \{-1, 1\}$. If there exists a δ-agnostic algorithm wrt $\mathcal{F}$ for noiseless active classification that achieves an expected sample complexity of τ on the instance $(y, \mathcal{F})$, then there exists a 2δ-agnostic algorithm wrt $\mathcal{F}$ for Gaussian active classification that achieves an expected sample complexity of $c\log(n/\delta)\tau$ samples on the instance $(y, \mathcal{F})$, where $c > 0$ is a universal constant.*

Proof. We perform a reduction by constructing a new algorithm $\mathcal{A}'$ for the Gaussian active classification problem instantiated by $\mathcal{F}$. Define $\mathcal{A}'$ as follows: at round t, pull the arm I_t that $\mathcal{A}$ would choose to pull in the noiseless active classification problem $O(\log(n/\delta))$ times. If the resulting mean is positive, then tell $\mathcal{A}$ that $\widehat{y}_{I_t} = 1$ otherwise tell $\mathcal{A}$ that $\widehat{y}_{I_t} = -1$. If the algorithm $\mathcal{A}$ terminates and outputs an $f \in \mathcal{F}$, then output f and terminate.

Define $f^* := \text{argmin}_{f\in\mathcal{F}} \sum_{i=1}^n \mathbb{1}\{y_i \neq f(x_i)\}$ and let $\widehat{f}$ be the classifier returned by $\mathcal{A}'$. Note that

$$\begin{aligned}\mathbb{P}(\widehat{f} \neq f^*) &\leq \mathbb{P}(\{\exists i \in [n] : \widehat{y}_i \neq y_i\} \cup \widehat{f} \neq f^*)\\ &\leq 2\delta\end{aligned}$$

using standard subGaussian bounds and the assumption that $\mathcal{A}$ is δ-agnostic. Thus, $\mathcal{A}'$ is 2δ-agnostic algorithm wrt $\mathcal{F}$ for Gaussian active classification.

Let T denote the (random) number of queries taken by $\mathcal{A}'$ on the instance $(y, \mathcal{F})$. Define the event $\Sigma = \{\widehat{y}_i \neq y_i \forall i \in [n]\}$. We have that

$$\mathbb{E}[T] = \mathbb{E}[T|\Sigma]\mathbb{P}(\Sigma) + \mathbb{E}[T|\Sigma^c]\mathbb{P}(\Sigma^c) \leq \tau \log(n/\delta) + n\log(n/\delta)\frac{1}{n} \leq 2\tau\log(n/\delta)$$

□

Corollary 1. *Let $y \in \{-1,1\}^n$, and $\mathcal{F} \subset \{-1,1\}^{\mathcal{X}}$, and define $f^* := argmin_{f\in\mathcal{F}} \sum_{i=1}^n \mathbb{1}\{y_i \neq f(x_i)\}$. Let $\mathcal{A}$ be a δ-agnostic algorithm wrt $\mathcal{F}$ on the active classification problem. Suppose $\mathcal{A}$ an expected sample complexity of $\mathbb{E}[\tau_y]$. Then,*

$$\mathbb{E}[\tau_y] \geq c\frac{\log(1/4.8\delta)}{\log(n/\delta)}\phi^*$$

where

$$\phi^* = \min_\lambda max_{\tilde{y}\in\{-1,1\}^n : argmin_{f\in\mathcal{F}} \sum_{i=1}^n \mathbb{1}\{\tilde{y}_i \neq f(x_i)\} \neq f^*} \frac{1}{\sum_{i=1}^n \lambda_i \mathbb{1}\{y_i \neq \tilde{y}_i\}}$$

Proof. Since $\mathcal{A}$ is δ-agnostic wrt $\mathcal{F}$ for active classification, by Proposition 11, we have that there exists a 2δ-agnostic algorithm $\mathcal{A}'$ wrt $\mathcal{F}$ for Gaussian active classification and that it has expected sample complexity of at most $c\log(n/\delta)\mathbb{E}[\tau_\theta]$. Define $\nu_{\tilde{y}_i} = N(\tilde{y}_i, 1)$. Using the Transportation Lemma from [13], since $\mathcal{A}'$ is 2δ-agnostic wrt $\mathcal{F}$, we have that for any $\tilde{y} \in \{-1,1\}^n$ such that $\text{argmin}_{f\in\mathcal{F}} \sum_{i=1}^n \mathbb{1}\{\tilde{y}_i \neq f(x_i)\} \neq f^*$,

$$\sum_{i=1}^n \mathbb{E}[T_i]\, \mathbf{KL}(\nu_{y_i}, \nu_{\tilde{y}_i}) \geq \log(\frac{1}{4.8\delta})$$

where T_i denotes the number of times that arm i is pulled. Defining $\lambda_i = \frac{\mathbb{E}[T_i]}{\sum_j \mathbb{E}[T_j]}$ and noting that

$$\mathbf{KL}(\nu_{y_i}, \nu_{\tilde{y}_i}) = \begin{cases} 2 & y_i = y_i \\ 0 & \text{otherwise}\end{cases}$$

this implies that

$$\sum_{i=1}^n \mathbb{E}[T_i] \geq \frac{1}{2}\log(1/4.8\delta)\frac{1}{\sum_{i=1}^n \lambda_i \mathbb{1}\{y_i \neq \tilde{y}_i\}}$$

Maximizing over $\tilde{y} \in \{-1,1\}^n$ such that $\text{argmin}_{f\in\mathcal{F}} \sum_{i=1}^n \mathbb{1}\{\tilde{y}_i \neq f(x_i)\} \neq f^*$ and then minimizing over $\lambda \in \mathbf{\Delta}_n$, we have that

$$\begin{aligned} &c\mathbb{E}[\tau_y]\log(n/\delta)\\ &\quad\geq \sum_{i=1}^n \mathbb{E}[T_i]\\ &\quad\geq \frac{1}{2}\log(1/4.8\delta)\min_\lambda \max_{\tilde{y}\in\{-1,1\}^n : \text{argmin}_{f\in\mathcal{F}} \sum_{i=1}^n \mathbb{1}\{\tilde{y}_i \neq f(x_i)\} \neq f^*} \frac{1}{\sum_{i=1}^n \lambda_i \mathbb{1}\{y_i \neq \tilde{y}_i\}}.\end{aligned}$$

Rearranging the above bound give the result. □

A.2 Best Arm Identification

Recall that for $k \in \mathbb{N}$ and $\delta \in (0,1)$, we say an algorithm $\mathcal{A}$ is (δ, k)-agnostic if for any $y \in \mathcal{Y}^n$ such that $\min_{f \in \mathcal{F}} \sum_{i=1}^n \mathbb{1}\{f(x_i) \neq y_i\} \leq k$, $\mathcal{A}$ identifies $\text{argmin}_{i \in [n]} y_i$ with probability at least $1 - \delta$.

We define the *Gaussian Best Arm Identification problem* as follows: when arm i is pulled, $y_i + \eta$ where $\eta \sim N(0,1)$ is observed. The goal is to identify $\text{argmin}_{i \in [n]} y_i$. We call a pair $(y, \mathcal{F})$ where $y \in \{-1,1\}^n$ an *instance*.

Proposition 12. *Let $\delta \in (0, \frac{1}{n})$. Fix $\mathcal{Y} = \{-1,1\}$, $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$, and $y \in \{-1,1\}$. Let $\mathcal{A}$ be a (δ, k)-agnostic algorithm wrt $\mathcal{F}$ for noiseless best arm identification that achieves an expected sample complexity of τ on the instance $(y, \mathcal{F})$ and has a probability p_i of sampling $i \in [n]$. Then there exists a $(2\delta, k)$-agnostic algorithm wrt $\mathcal{F}$ for Gaussian best arm identification that achieves an expected sample complexity of $c \log(n/\delta)\tau$ samples on the instance $(y, \mathcal{F})$, where $c > 0$ is a universal constant, and has a probability p_i of sampling $i \in [n]$.*

Proof. The proof is identical to the proof of Proposition 11. The probability p_i is the same for Gaussian best arm identification setting by the construction of $\mathcal{A}'$ in the proof of Proposition 11. □

Proof of Proposition 3. Let $\mathcal{Y} = \{w_1, \ldots, w_L\}$ with $w_1 < w_2 < \ldots < w_L$. Let $\mathcal{F} = \{f^*\}$ with $f^* : \mathcal{X} \mapsto \{w_1, \ldots, w_L\}$ such that for all $i \in [n]$ $f^*(x_i) \in \{w_2, \ldots, w_{L-1}\}$ and $\text{argmin}_{i \in [n]} f^*(x_i) = \{i_*\}$.

Step 0: there exists an arm that is not queried with constant probability. Let $\theta \in \{w_1, \ldots, w_L\}^n$ such that $\theta_i = f^*(x_i)$ for all $i \in [n]$. Let T_i denote the number of times that arm i is queried (a random variable). Towards a contradiction, suppose that $\mathbb{E}_\theta[\sum_{i=1}^n T_i] < n/8$. Since

$$n/8 > \mathbb{E}_\theta[\sum_{i=1}^n T_i] \geq \min_{i \in n} n \mathbb{E}_\theta[T_i].$$

Thus, there exists $i_0 \in [n]$ such that $1/8 > \mathbb{E}_\theta[T_{i_0}] \geq \mathbb{P}_\theta(T_{i_0} > 0)$.

Step 1: Reduction

By Proposition 12, since $\delta < 1/n$, there exists a $(2\delta, 1)$-agnostic algorithm $\mathcal{A}'$ wrt $\mathcal{F}$ for Gaussian best arm identification that achieves an expected sample complexity of $c \log(n/\delta)\tau$ samples on the instance $(y, \mathcal{F})$ and the probability of querying i_0 is at most $1/8$.

Step 2.1: Constructing a bad alternative Let $\widehat{i}$ denote the arm returned at the end of the game. Suppose $i_0 \notin \text{argmin}_{j \in [n]} f^*(x_j)$. Then, since $\delta < \frac{1}{40}$ and $\mathcal{A}'$ is $(2\delta, 1)$-agnostic wrt $\mathcal{F}$ for Gaussian best arm identification

$$\mathbb{P}_\theta(\widehat{i} \neq i_0) \geq 19/20.$$

Define $B = \{T_{i_0} = 0\} \cap \{\widehat{i} \neq i_0\}$. Then,

$$\mathbb{P}_\theta(B^c) \leq \mathbb{P}_\theta(T_{i_0} > 0) + \mathbb{P}_\theta(\widehat{i} = i_0) \leq 1/8 + 1/20 \leq 1/5.$$

Define $\tilde{\theta} \in \{w_1, \ldots, w_L\}^n$ by

$$\tilde{\theta}_j = \begin{cases} f^*(x_j) & j \neq i_0 \\ w_1 & j = i_0 \end{cases}.$$

Note that since by assumption $f^*(x_i) \in \{w_2, \ldots, w_{L-1}\}$ for all $i \in [n]$, under $\tilde{\theta}$, i_0 is the unique minimizer.

Define

$$\widehat{\text{kl}}_{i,T_i} = \sum_{s=1}^{T_i} \log(f_\theta(Z_s)/f_{\tilde{\theta}}(Z_s))$$

where Z_s is the observation on the sth pull of arm i, $f_{\bar{\theta}}$ denotes the density of the distribution associated with arm i under $\bar{\theta}$, specifically $N(\bar{\theta}_i, 1)$. Then, by the change of measure identity (Lemma 18) from [13],

$$\begin{aligned}
\mathbb{P}_{\tilde{\theta}}(\widehat{i} \neq i_0) &\geq \mathbb{P}_{\tilde{\theta}}(B) \\
&= \mathbb{E}_{\theta}[\mathbb{1}\{B\} \exp(-T_i \widehat{\mathrm{kl}}_{i,T_i})] \\
&= \mathbb{P}_{\theta}(B) \\
&\geq \frac{4}{5}.
\end{aligned}$$

where we used the fact that the only difference between problem θ and problem $\tilde{\theta}$ is the mean of ith arm and on the event A, $T_i = 0$. But, this is a contradiction since under $\tilde{\theta}$, i_0 is the minimizer, $\sum_{i=1}^n \mathbb{1}\{\theta_i \neq \tilde{\theta}_i\} = 1$, and the algorithm $\mathcal{A}$ is $(2\delta, 1)$-agnostic with $\delta \in (0, 1/20)$.

Step 2.2:

Suppose that $i_0 = i_*$. Define $\bar{\theta} \in \{w_1, \dots, w_L\}^n$ by

$$\bar{\theta}_j = \begin{cases} f^*(x_j) & j \neq i_* \\ w_L & j = i_* \end{cases}.$$

Define the events $\Sigma_1 = \{T_{i_*} = 0\}, \Sigma_2 = \{i_* = \widehat{i}\}, \Sigma = \Sigma_1 \cap \Sigma_2$. Note that

$$\begin{aligned}
\mathbb{P}_{\theta}(\Sigma^c) &\leq \mathbb{P}_{\theta}(\Sigma_1^c) + \mathbb{P}_{\theta}(\Sigma_2^c) \\
&= \mathbb{P}_{\theta}(T_{i_*} > 0) + \mathbb{P}_{\theta}(\widehat{i_*} \neq \widehat{i}) \\
&\leq 1/8 + 2\delta \\
&\leq 1/5
\end{aligned}$$

This implies by a similar argument to the one made in Step 2.1 that

$$\begin{aligned}
\mathbb{P}_{\bar{\theta}}(\Sigma_2) \geq \mathbb{P}_{\bar{\theta}}(\Sigma) &= \mathbb{E}_{\theta}[\mathbb{1}\{\Sigma\} \exp(-T_{i_*} \widehat{\mathrm{kl}}_{i,T_{i_*}})] \\
&= \mathbb{P}_{\theta}(\Sigma) \\
&\geq 4/5.
\end{aligned}$$

But, this contradicts the assumption that $\mathcal{A}$ is $(2\delta, 1)$-agnostic since $\sum_{i=1}^n \mathbb{1}\{\theta_i \neq \bar{\theta}_i\} = 1$ and i_* is suboptimal in the instance given by $\bar{\theta}$ and $\mathcal{A}$ therefore makes a mistake with probability at least $4/5 > \delta$, a contradiction.

□

A.3 Regret Minimization

Proof of Proposition 2. Consider an agnostic algorithm $\mathcal{A}$ for noiseless regret minimization. We assume there exist n arms $x_1, \cdots, x_n =: \mathcal{X}$ and an unknown true function f^* in a class $\mathcal{F} : \mathcal{X} \to \mathbb{R}$. In the noiseless regime, we assume that pulling x_i returns $f^*(x_i)$. In the noisy regime, we assume that pulling x_i returns $f^*(x_i) + \eta$ where η is a 1 sub-Gaussian random variable.

In the noiseless regime, we assume that $\mathcal{A}$ achieves a bounded regret $R_{\mathcal{A}}(f^*, \mathcal{X}) < \infty$ independent of the horizon. This is a mild assumption as noise is not present in the responses. Indeed, the simple algorithm that samples each x_i once and then chooses the x^* achieving minimal loss $f(x)$ thereafter has bounded regret. Additionally, we may assume without loss of generality that $\mathcal{A}$ never re-samples a suboptimal arm as its value is known.

Step 1: Reduction from regret minimization in the noiseless regime to the noisy regime.

By [31], an algorithm $\mathcal{A}'$ for the noisy regime is said to be *consistent* if its regret at time T, $R_{f^*}^{\mathcal{A}'}(T)$ obeys

$$\limsup_{T \to \infty} \frac{\log(R_{f^*}^{\mathcal{A}'}(T))}{\log(T)} \leq 0.$$

We say that f^* is ϵ-separated on $x_1, \cdots, x_n$ if $\min_{i,j: f^*(x_i) \neq f^*(x_j)} |f^*(x_i) - f^*(x_j)| \geq \epsilon$. Next, we show how to use algorithm $\mathcal{A}$ in the noiseless regime to form a consistent algorithm $\mathcal{A}'$ for 1-separated instances. For simplicity, we assume that f^* takes values over the integers. Let $T > 0$ denote the time horizon and assume that $\max|f(x_i) - f(x_j)| \leq B$. At each time t, $\mathcal{A}'$

1. queries $\mathcal{A}$ for an arm x_i to sample next
2. samples x_i $c\log(nT)$ times for sufficiently large constant c.
3. Average the losses and round the number to the nearest integer.
4. Pass the rounded average to $\mathcal{A}$.

Next, we argue that $\mathcal{A}'$ is consistent. For sufficiently large c and a union bound over the n arms, with probability $1 - 1/T$, we have that $f^*(x_i) = R\left(\frac{1}{c\log(nT)} \sum_{j=1}^{c\log(nT)} f(x_i) + \eta_j\right)$ where η_j is the noise on the j^{th} sample of x_i and $R(\cdot)$ is a rounding function that rounds to the nearest integer value. In this case, since $\mathcal{A}$ has bounded regret and never re-pulls suboptimal arms, the total regret is $cR_A(f^*, \mathcal{X})\log(nT)$. Otherwise, with probability $1/T$, $\mathcal{A}'$ receives regret no worse than BT. Hence the total regret is bounded by

$$R_{f^*}^{\mathcal{A}'}(T) \leq cR_A(f^*, \mathcal{X})\log(nT) + B.$$

Therefore,

$$\begin{aligned}\limsup_{T\to\infty} \frac{\log(R_{f^*}^{\mathcal{A}'}(T))}{\log(T)} &= \limsup_{T\to\infty} \frac{\log\left(cR_A(f^*, \mathcal{X})\log(nT) + B\right)}{\log(T)} \\ &= \limsup_{T\to\infty} \frac{\log\left(cR_A(f^*, \mathcal{X})\log(nT)\right)}{\log(T)} \\ &= 0\end{aligned}$$

which implies that $\mathcal{A}'$ is consistent.

Step 2: Instance dependent regret bounds for consistent algorithms.

For any consistent algorithm $\mathcal{A}'$, by Corollary 2 of [31]

$$C(f^*, \mathcal{X}) \leq \limsup_{n\to\infty} \frac{R_{f^*}^{\mathcal{A}'}(T))}{\log(n)} = \limsup_{n\to\infty} \frac{cR_A(f^*, \mathcal{X})\log(nT) + B}{\log(n)} = cR_A(f^*, \mathcal{X})$$

where $C(f^*, \mathcal{X})$ is the optimal solution to

$$\min_{\alpha \in (0,\infty]^{\mathcal{X}}} \sum_{x \in \mathcal{X}^-} \alpha_i \Delta_i \text{ s.t } 1/\alpha_i \leq \Delta_i^2/2 \; \forall x_i \in \mathcal{X}^-$$

where $\Delta_i := f^*(x_i) - \min_{x\in\mathcal{X}} f^*(x)$ and $\mathcal{X}^- := \{x \in \mathcal{X} : \Delta_x > 0\}$. Choosing $\alpha_i = 2/\Delta_i^2$ to saturate the bound, we have that

$$C(f^*, \mathcal{X}) = \sum_{x \in \mathcal{X}^-} \frac{2}{\Delta_i}.$$

Step 3: Constructing an instance where $R_{\mathcal{A}}(f^*, \mathcal{X})$ is large.

Fix $n \in \mathbb{N}$ and consider a set of points $\mathcal{X} = \{x_1, \cdots, x_{2n}\}$. We consider a family of n functions $f_1, \cdots, f_n$ mapping from $\mathcal{X}$ to $\{-1, 0, 1\}$. By definition each f_i is 1-separated. Define

$$f_i(x_j) = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{if } j \leq n \text{ and } j \neq i \\ -1 & \text{if } j \in [2n - i, 2n] \end{cases}$$

For any f_i, the set of suboptimal arms is given by $\mathcal{X}^- = [2n]\backslash\{i\}$. Choose $f^* \in \{f_1, \cdots, f_n\}$ uniformly. By step 2, we have that $R_{\mathcal{A}}(f^*, \mathcal{X}) \gtrsim (n-1)$ since $\Delta_i = O(1)$ where we have used the fact that $\mathcal{A}$ was an agnostic algorithm with bounded regret.

Conversely, a realizability-based algorithm that knows the version space $\{f_1, \cdots, f_n\}$ could perform binary search over $x_{n+1}, \cdots, x_{2n}$ and identify the maximizer of f^* in $O(\log(n))$ pulls and achieving regret at most $O(\log(n))$. □

B Construction of Sampling Oracles for Efficient Active Learning

We now show that non-trivial sampling oracles are available for many useful function classes.

Linear Models: Define $\mathcal{R} = \{\langle a, \cdot\rangle : a \in \mathbb{R}^d, \|a\| \leq B\}$ where $\|\cdot\|$ is some norm and $B \in \mathbb{R}$. Let π denote the uniform distribution on $D := \{a \in \mathbb{R}^d : \|a\| \leq B\}$. Since D a compact convex set, there exists a polynomial time algorithm to sample from D [14]. Now, define $\bar{D} = D \cap \cap_{j\in[m]}\{a : \langle a, w_j\rangle \leq b_j\}$ where $w_j n \mathbb{R}^d$ and $b_j \in \mathbb{R}$. Let $\bar{\pi}$ be the uniform distribution on $\bar{D}$. Note that D has a membership oracle: for a given $x \in \mathbb{R}^d$, one can check whether $x \in D$ by computing $\|x\|$. Since $\bar{D}$ is a convex body and has a membership oracle, there exists a polynomial time algorithm to sample from $\bar{D}$. Finally, note that $\mathrm{uniform}(\bar{D})$ is the desired condition distribution:

$$\mathbb{P}_{z\sim\mathrm{uniform}(\bar{D})}(z \in A) = \frac{\mathbb{P}_{z\sim\mathrm{uniform}(D)}(z \in A)}{\mathbb{P}_{z\sim\mathrm{uniform}(D)}(z \in \bar{D})}.$$

Thus, the sample oracle exists for $\mathcal{R}$.

Kernel Method: Let $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ be a kernel and let $\mathcal{R}$ be the associated RKHS. Let $\mathcal{R}_B = \{f \in \mathcal{R} : \|f\|_{\mathcal{R}} \leq B\}$. Fix $x_1, \ldots, x_n \in \mathcal{X}$. By standard arguments in kernel methods (e.g., Representer Theorem [32]), for every $f \in \mathcal{R}_B$, there exists $f' \in \mathcal{R}_B$ such that $f'(\cdot) = \sum_{i=1}^n \alpha_i k(\cdot, x_i)$ and $f'(x_i) = f(x_i)$ for all $i \in [n]$. Thus, it suffices to consider the function class

$$\tilde{\mathcal{R}} = \{\sum_{i=1}^n \alpha_i k(\cdot, x_i) : \|\alpha\|_{K^{1/2}} = \left\|\sum_{i=1}^n \alpha_i k(\cdot, x_i)\right\|_{\mathcal{R}} \leq B\}.$$

Thus, it is enough to sample from the uniform distribution on

$$D = \{\alpha \in \mathbb{R}^n : \|\alpha\|_{K^{1/2}} \leq B\}$$

which reduces to the linear case. Intersecting D with halfspaces is also similar to the linear case.

Convex Functions: Let $\mathcal{R} = \{f : \mathcal{X} \mapsto [-B_1, B_1], f \text{ convex}, \max_{x\in\mathcal{X}, g\in\partial f(x)} \|g\|_2 \leq B_2\}$ where $\partial f(x)$ denotes the subdifferential of f at x. Let $f \in \mathcal{R}$. Then, $f(x_i) \geq f(x_j) + g_j^\top(x_i - x_j)$ for all $i, j \in [n]$ where $g_j \in \partial f(x_j)$ with $\|g_j\|_2 \leq B_2$. Thus, there exists $\widehat{y}_i = f(x_i)$ and $g_i \in \mathbb{R}^d$ with $\|g_i\|_2 \leq R_2$ for $i = 1, \ldots, n$ such that

$$\widehat{y}_i \geq \widehat{y}_j + g_j^\top(x_i - x_j).$$

Note that $\bar{f}(x_i) = \max_{i=1,\ldots,n} \widehat{y}_i + g_i^\top(x - x_i)$ is convex and $\bar{f}(x_i) = f(x_i)$ for all i [33]. This shows that it suffices to sample from the set $D :=$

$$\{(\widehat{y}_i)_{i=1}^n (g_i)_{i=1}^n) \in \mathbb{R}^{n(d+1)} : \widehat{y}_i \geq \widehat{y}_j + g_j^\top(x_i - x_j) \wedge \widehat{y}_i \in [-B_1, B_1], \|g_i\|_2 \leq B_2 \forall i \in [n]\}.$$

Note that D has a membership oracle: one can efficiently check whether $x \in D$ by checking whether it satisfies the inequalities defining D. Since D is convex and compact, and has a membership oracle, one can construct a polynomial-time algorithm for sampling from it. Intersecting D with halfspaces is similar to the linear case. Note that one can construct a sampling oracle in a similar way for α**-strongly convex functions**: one need only replace the constraint $\widehat{y}_i \geq \widehat{y}_j + g_j^\top(x_i - x_j)$ with $\widehat{y}_i \geq \widehat{y}_j + g_j^\top(x_i - x_j) + \frac{\alpha}{2}\|x_i - x_j\|_2^2$.

C Best Arm Identification Proofs

Here we state and prove a strictly more general result than Theorem 1 for ϵ-good arm identification. Theorem 1 follows as a special case. First, we define:

Definition 1. *$\Lambda_{best,\epsilon}(\mathcal{F})$ is the smallest integer q such that there exists some algorithm $\mathcal{A}$ such that for every $f^* \in \mathcal{F}$, $\mathcal{A}$ outputs an element of $\{j \in [n] : f^*(x_j) \leq \min_{i\in[n]} f^*(x_i) + \epsilon\}$ after at most q queries.*

Theorem 6. *For any $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$, $\upsilon^*_{best,\epsilon}(\mathcal{F}) \leq \Lambda_{best,\epsilon}(\mathcal{F})$.*

Proof of Theorem 6. Consider an algorithm $\mathcal{A}$ that for all $f^* \in \mathcal{F}$ identifies $i_0 \in [n]$ such that $f^*(x_{i_0}) \leq \min_{i\in[n]} f^*(x_i) + \epsilon$ using at most $C(\mathcal{F})$ queries. Towards a contradiction, suppose

$C(\mathcal{F}) < \upsilon^*_{best,\epsilon}$. There exists some $g : \mathcal{X} \mapsto \mathcal{Y}$ achieving the maximum in υ^*_{best}. Let $I \subset [n]$ be the subset of arms queried by $\mathcal{A}$. Then, by definition of υ^*_{best}, $\forall j \in [n]$ there exist $f, f' \in \mathcal{F}$ such that $f(x_i) = f'(x_i) = g(x_i)$ for all $i \in I$ and either $f(x_j) > \min_{i\in[n]} f(x_i) + \epsilon$ or $f(x_j) > \min_{i\in[n]} f'(x_i) + \epsilon$. Thus, $\mathcal{A}$ cannot distinguish between these two functions and determine the ϵ-good arm. Therefore, if $\mathcal{A}$ outputs $j \in [n]$ after $C(\mathcal{F}) < \upsilon^*_{best}$ queries, there exists some $f \in \mathcal{F}$ on which $\mathcal{A}$ would make a mistake, leading to a contradiction. □

Proof of Proposition 4. **The Lower Bound.** Let $F \subset \mathcal{F}$. Define $g : \mathcal{X} \mapsto \mathcal{Y}$ by

$$g(x_i) = \text{argmin}_{y\in\mathcal{Y}} |F(x_i) \cup F((x_i, y))|$$

By definition of υ^*_{best}, there exists $\tilde{I} \subset [n]$ such that $|\tilde{I}| \leq \upsilon^*_{best}$ and

$$\{f \in F : f(x_i) = g(x_i)\, \forall i \in \tilde{I}\} \subset \{f \in F : j_0 \in \text{argmin}_{k\in[n]} f(x_k)\}$$

Then, defining $I = \tilde{I} \cup \{j_0\}$, we have that

$$F = \{f \in F : \exists i \in I \text{ s.t. } f(x_i) \neq g(x_i) \text{ or } i \in \text{argmin}_{k\in[n]} f(x_k)\}.$$

Therefore,

$$\begin{aligned}
|F| &= |\{f \in F : \exists i \in I \text{ s.t. } f(x_i) \neq g(x_i) \text{ or } i \in \text{argmin}_{k\in[n]} f(x_k)\}| \\
&\leq \sum_{i\in I} |\{f \in F : f(x_i) \neq g(x_i) \text{ or } i \in \text{argmin}_{k\in[n]} f(x_k)\}| \\
&\leq (\upsilon^*_{best} + 1)\text{max}_{i\in I} |\{f \in F : f(x_i) \neq g(x_i) \text{ or } i \in \text{argmin}_{k\in[n]} f(x_k)\}| \\
&\leq (\upsilon^*_{best} + 1)\text{max}_{i\in [n]} |\{f \in F : f(x_i) \neq g(x_i) \text{ or } i \in \text{argmin}_{k\in[n]} f(x_k)\}| \\
&= (\upsilon^*_{best} + 1)\text{max}_{i\in [n]}\text{argmin}_{y\in\mathcal{Y}} |\{f \in F : f(x_i) \neq y \text{ or } i \in \text{argmin}_{k\in[n]} f(x_k)\}| \\
&= (\upsilon^*_{best} + 1)\text{max}_{i\in [n]}\text{argmin}_{y\in\mathcal{Y}} |F((x_i, y)) \cup F(x_i)|
\end{aligned}$$

Rearranging, we have that

$$\frac{|F|}{\text{max}_{i\in[n]}\text{argmin}_{y\in\mathcal{Y}} |F((x_i, y)) \cup F(x_i)|} \leq \upsilon^*_{best} + 1.$$

Since the above inequality holds for any $F \subset \mathcal{F}$, we have that

$$\text{HD}(F) = \sup_{F\subset\mathcal{F}} \frac{|F|}{\text{max}_{i\in[n]}\text{argmin}_{y\in\mathcal{Y}} |F((x_i, y)) \cup F(x_i)|} \leq \upsilon^*_{best} + 1$$

This completes the lower bound.

The Upper Bound. The proof is essentially identical to the proof from [11]. Let $g : \mathcal{X} \mapsto \mathcal{Y}$ be arbitrary. Consider the algorithm from [11] that sets $\mathcal{F}_1 = \mathcal{F}$, $\mathcal{F}_{t+1} = \mathcal{F}_t \backslash \mathcal{F}_t(x_{I_t}) \cup \mathcal{F}_t((x_{I_t}, g(x_{I_t})))$, and at each round t chooses $I_t \in \arg\max_{i\in[n]} \inf_{y\in\mathcal{Y}} \mathcal{F}_t(x_i) \cup \mathcal{F}_t((x_{I_t}, y))$. Consider round $t \leq c\text{HD}(\mathcal{F})\ln(|\mathcal{F}|)$ for some positive constant to be determine later. If $|\mathcal{F}_t| = 0$, then we are done. Thus, suppose that $|\mathcal{F}_t| > 1$. By definition, we have that

$$\text{HD}(\mathcal{F}) \leq \gamma(\mathcal{F}_t) \leq \frac{|\mathcal{F}_t(x_{I_t}) \cup \mathcal{F}_t((x_{I_t}, f^*(x_{I_t})))|}{|\mathcal{F}_t|}.$$

Therefore, using $|\mathcal{F}_t| = |\mathcal{F}_{t+1}| + |\mathcal{F}_t(x_{I_t}) \cup \mathcal{F}_t((x_{I_t}, f^*(x_{I_t})))|$, we have that

$$|\mathcal{F}_{t+1}| \leq (1 - \text{HD}(\mathcal{F}))|\mathcal{F}_t|.$$

Unrolling the recurrence, we have that $|\mathcal{F}_{t+1}| \leq (1 - \text{HD}(\mathcal{F}))^t|\mathcal{F}|$. Plugging in $t = c\text{HD}(\mathcal{F})\ln(|\mathcal{F}|)$ for a large enough positive constant c shows that $\mathcal{F}_t = \emptyset$. Therefore there trivially exists $j \in [n]$ such that for any $f \in \mathcal{F}$ such that $f(x_i) = g(x_i)$ for all $i \in \{I_1, \ldots, I_t\}$, $j \in \text{argmin}_{k\in[n]} f(x_k)$. Thus, by definition of υ^*_{best}, we have that

$$\upsilon^*_{best} \leq c\text{HD}(\mathcal{F})\ln(|\mathcal{F}|).$$

□

Proof of Theorem 2. For the sake of abbreviation, let $\mathbb{P}(\cdot) := \mathbb{P}_{r\sim\pi}(\cdot)$ and $\mathbb{P}_k(\cdot) := \mathbb{P}_{r\sim P_k}(\cdot)$.

Step 1: The Algorithm 1 breaks time into phases $[t_1, t_2]$, $[t_2, t_3]$, $\cdots$. In each phase k, the algorithm samples functions from a distinct distribution P_k. Consider phase k, let $t \in [t_k, t_{k+1}]$. Define $\theta \in \mathcal{Y}^n$ by

$$\theta_i = \begin{cases} y_i & i \in O_t \\ \text{argmin}_{y\in\mathcal{Y}}\mathbb{P}_k(r \in \mathcal{R}_t(x_i) \cup \mathcal{R}_{t,\epsilon}((x_i, y))) & \text{o/w} \end{cases}.$$

By definition of $\upsilon^*_{best,\epsilon}$, there exists $\tilde{I} = \{i_1, \ldots, i_{\upsilon^*_{best,\epsilon}}\}$ such that there exists $j_0 \in [n]$ satisfying

$$\{r \in \mathcal{R}_t : \mathcal{D}[r(x_i)] = \theta_i \ \forall i \in \tilde{I}\} \subset \{r \in \mathcal{R}_t : \mathcal{D}[r(x_{j_0})] \leq \min_{l\in[n]} \mathcal{D}[r(x_l)] + \epsilon\}$$

Define $I = \{i_1, \ldots, i_{\upsilon^*_{best}}, j_0\}$. Then, we have that

$$\mathcal{R}_t = \{r \in \mathcal{R}_t : \exists i \in I \text{ s.t. } \mathcal{D}[r(x_i)] \leq \min_{l\in[n]} \mathcal{D}[r(x_l)] + \epsilon \text{ or } \mathcal{D}[r(x_i)] \neq \theta_i\}.$$

From this, it follows that

$$\begin{aligned}
\mathbb{P}_k(\mathcal{R}_t) &= \mathbb{P}_k(\{r \in \mathcal{R}_t : \exists i \in I \text{ s.t. } \mathcal{D}[r(x_i)] \leq \min_{l\in[n]} \mathcal{D}[r(x_l)] + \epsilon \text{ or } \mathcal{D}[r(x_i)] \neq \theta_i\}) \\
&\leq \sum_{i\in I} \mathbb{P}_k(\{r \in \mathcal{R}_t : \mathcal{D}[r(x_i)] \leq \min_{l\in[n]} \mathcal{D}[r(x_l)] + \epsilon \text{ or } \mathcal{D}[r(x_i)] \neq \theta_i\}) \\
&\leq (\upsilon^*_{best,\epsilon} + 1)\text{max}_{i\in I}\mathbb{P}_k(\{r \in \mathcal{R}_t : \mathcal{D}[r(x_i)] \leq \min_{l\in[n]} \mathcal{D}[r(x_l)] + \epsilon \text{ or } \mathcal{D}[r(x_i)] \neq \theta_i\}) \\
&= (\upsilon^*_{best,\epsilon} + 1)\mathbb{P}_k(\{r \in \mathcal{R}_t : \mathcal{D}[r(x_{I_t})] \leq \min_{l\in[n]} \mathcal{D}[r(x_l)] + \epsilon \text{ or } \mathcal{D}[r(x_i)] \neq \theta_{I_t}\}) \qquad (1) \\
&\leq (\upsilon^*_{best,\epsilon} + 1)\mathbb{P}_k(\{r \in \mathcal{R}_t : \mathcal{D}[r(x_{I_t})] \leq \min_{l\in[n]} \mathcal{D}[r(x_l)] + \epsilon \text{ or } \mathcal{D}[r(x_i)] \neq y_{I_t}\}) \qquad (2) \\
&= (\upsilon^*_{best,\epsilon} + 1)\mathbb{P}_k(\mathcal{R}_{t,\epsilon}(x_{I_t}) \cup \mathcal{R}_t((x_{I_t}, y_{I_t})))
\end{aligned}$$

where line (1) comes from the definition of I_t and the definition of θ, and the line (2) comes from the definition of θ. Noticing that

$$\begin{aligned}
\mathcal{R}_t &= (\mathcal{R}_{t,\epsilon}(x_{I_t}) \cup \mathcal{R}_t((x_{I_t}, y_{I_t})) \cup (\mathcal{R}_t \setminus (\mathcal{R}_{t,\epsilon}(x_{I_t}) \cup \mathcal{R}_t((x_{I_t}, y_{I_t}))) \\
&= (\mathcal{R}_{t,\epsilon}(x_{I_t}) \cup \mathcal{R}_t((x_{I_t}, y_{I_t})) \cup \mathcal{R}_{t+1},
\end{aligned}$$

we have that

$$\mathbb{P}_k(\mathcal{R}_{t+1}) = \mathbb{P}_k(\mathcal{R}_t) - \mathbb{P}_k(\mathcal{R}_{t,\epsilon}(x_{I_t}) \cup \mathcal{R}_t((x_{I_t}, y_{I_t})) \leq (1 - \frac{1}{\upsilon^*_{best} + 1})\mathbb{P}_k(\mathcal{R}_t).$$

Unraveling this recursive statement, we have that

$$\mathbb{P}_k(\mathcal{R}_{t+1}) \leq (1 - \frac{1}{\upsilon^*_{best,\epsilon} + 1})^{t-t_k}.$$

Thus, we see that in phase k at $t = t_k + s_k$ with $s_k := O(\ln(n)\upsilon^*_{best})$, we have that

$$\mathbb{P}_k(\mathcal{R}_{t_k+s_k}) \leq \frac{1}{2n}. \qquad (3)$$

Step 2: Reduction in each version space. Fix the first round t such that $\mathbb{P}_k(r \in \mathcal{R}_t) \leq \frac{1}{2n}$. Note that $t_{k+1} = t$. Fix $l_0 \in [n] \setminus O_{t_k}$. Then, we have that

$$\begin{aligned}
\frac{1}{2n} &\geq \frac{1}{n - |O_{t_k}|} \sum_{l \in [n] \setminus O_{t_k}} \mathbb{P}_{f \sim \pi_{C_l(O_{t_k})}}(\mathcal{R}_t) \\
&\geq \frac{1}{n} \sum_{l \in [n] \setminus O_{t_k}} \mathbb{P}_{f \sim \pi_{C_l(O_{t_k})}}(\mathcal{R}_t) \\
&\geq \frac{1}{n} \mathbb{P}_{f \sim \pi_{C_{l_0}(O_{t_k})}}(\mathcal{R}_t) \\
&= \frac{1}{n} \frac{\mathbb{P}(\mathcal{R}_t \cap C_{l_0}(O_{t_k}))}{\mathbb{P}(C_{l_0}(O_{t_k}))} && (4) \\
&= \frac{1}{n} \frac{\mathbb{P}(\mathcal{R}_{t_{k+1}} \cap C_{l_0}(O_{t_k}))}{\mathbb{P}(C_{l_0}(O_{t_k}))} \\
&\geq \frac{1}{n} \frac{\mathbb{P}(C_{l_0}(O_{t_{k+1}}))}{\mathbb{P}(C_{l_0}(O_{t_k}))} && (5)
\end{aligned}$$

where the equality (4) follows by the definition of conditional probability and in the inequality (5) we used that

$$C_{l_0}(O_{t_{k+1}}) \subset \mathcal{R}_t \cap C_{l_0}(O_{t_k}).$$

The above display can be seen since by inspection of the definition for any $t_k \leq t_{k+1}$ and $l \in [n]$,

$$\begin{aligned}
C_l(O_{t_{k+1}}) &= \{r \in \mathcal{R} : \mathcal{D}[r(x_i)] = y_i \forall i \in O_{t_{k+1}}\} \cap \{\mathcal{D}[r(x_l)] + \epsilon < \mathcal{D}[r(x_i)] \forall i \in O_{t_{k+1}}\} \\
&\subset \{r \in \mathcal{R} : \mathcal{D}[r(x_i)] = y_i \forall i \in O_k\} \cap \{\mathcal{D}[r(x_l)] + \epsilon < \mathcal{D}[r(x_i)] \forall i \in O_{t_k}\} \\
&\subset C_l(O_{t_k})
\end{aligned}$$

and

$$C_l(O_{t_{k+1}}) \subset \cup_{j \in [n] \setminus O_{t_{k+1}}} C_j(O_{t_{k+1}}) = \mathcal{R}_{t_{k+1}}.$$

Thus, we have that

$$\frac{\mathbb{P}(C_{l_0}(O_{t_k}))}{2} \geq \mathbb{P}(C_{l_0}(O_{t_{k+1}})). \qquad (6)$$

Step 3: Eliminating functions. Let $f^* \in \mathcal{F}_{\mathcal{R}}$ such that there exists $r^* \in \mathbb{R}$ such that $f^*(x_i) = \mathcal{D}[r^*(x_i)] = y_i$ for all $i \in [n]$. Thus, using (3) and (6) in Steps 1 and 2 respectively, we see that after $\bar{t} := cv^*_{best} \log(\frac{1}{\mathbb{P}(S_{f^*})}) \log(n)$ queries we have that for all $l \in [n]$,

$$\mathbb{P}(C_l(O_{\bar{t}}))) < \mathbb{P}(S_{f^*}).$$

This implies that $S_{f^*} \not\subset C_l(O_{t_{\log(\frac{1}{\mathbb{P}(S_{f^*})})}}))$ for all l. Therefore,

$$S_{f^*} \not\subset \cup_{l=1}^{n} C_l(O_{\bar{t}})) = \mathcal{R}_{\bar{t}}.$$

Therefore, f^* was kicked out some round prior to $\bar{t}$. Since $\mathcal{D}[r^*(x_i)] = y_i$ for all $i \in [n]$, inspection of Algorithm 1 reveals that the only way that f^* could be removed from the version space is there exists some $s \leq \bar{t}$ I_s is queried where $f^*(x_{I_s}) \leq \min_{j \in [n]} f^*(x_j) + \epsilon$. This completes the proof.

□

D Cumulative Loss Minimization Proofs

Proof of Theorem 3. Consider an algorithm $\mathcal{A}$ that after incurring cost at most $C(\mathcal{F})$ declares that it has identified the minimizer. There exists some $\bar{g} : \mathcal{X} \mapsto \{w_1, \ldots, w_L\}$ achieving the maximum in v^*_{loss}. Suppose the algorithm queries $i_1, \ldots, i_{t_0}$ and receives feedback $g(x_{i_j})$ and time t_0 is the earliest point at which the algorithm can declare that it can identify $j_0 \in [n]$ such that

$$\{f \in \mathcal{F} \text{ s.t. } f(x_i) = \bar{g}(x_i) \forall i \in \{i_1, \ldots, i_{t_0}\}\} \subset \{f \in \mathcal{F} : j_0 \in \text{argmin}_{i \in [n]} f(x_i)\}.$$

```
$\mathcal{F}_1 \longleftarrow \mathcal{F}$ ;
for $t = 1, 2, \dots, T$ do
    Let $I_t \in \text{argmin}_{i \in [n] \setminus \{I_1, \dots, I_{t-1}\}} \text{max}_{\bar{f} \in \mathcal{F}_t} \frac{\bar{f}(x_i)}{|\mathcal{F}_t(x_i) \cup \mathcal{F}((x_i, \bar{f}(x_i)))|}$;
    Query $x_{I_t}$ and observe $f^*(x_{I_t})$;
    Let $\mathcal{F}_{t+1} \longleftarrow \mathcal{F}_t \setminus \mathcal{F}_t(x_{I_t}) \cup \mathcal{F}((x_{I_t}, f^*(x_{I_t}))$
```

Algorithm 3: `GRAILS` for Loss Minimization via Enumeration

We claim that there exists $\bar{f}$ such that

$$\{f \in \mathcal{F} \text{ s.t. } f(x_i) = \bar{f}(x_i) \forall i \in \{i_1, \dots, i_{t_0 - 1}\}\} \neq \emptyset.$$

If not, then the algorithm could terminate at time $t_0 - 1$, contradicting the definition of t_0. Thus, on the instance given by $\bar{f}$, the algorithm suffers

$$\begin{aligned}
&\sum_{s=1}^{t_0-1} \bar{f}(x_{i_s}) \\
&= \sum_{s=1}^{t_0-1} \bar{g}(x_{i_s}) \\
&\geq \min_{I \subset [n]} \min_{\tilde{I} \subset I : |\tilde{I}| = |I| - 1} \sum_{i \in \tilde{I}} \bar{g}(x_i) \\
&\qquad \text{s.t. } \exists j \in [n] \text{ s.t. } \{f \in \mathcal{F} \text{ s.t. } f(x_i) = \bar{g}(x_i) \forall i \in [I]\} \subset \{f \in \mathcal{F} : j \in \text{argmin}_{i \in [n]} f(x_i)\} \\
&= \text{max}_{g : \mathcal{X} \mapsto \mathcal{Y}} \min_{I \subset [n]} \min_{\tilde{I} \subset I : |\tilde{I}| = |I| - 1} \sum_{i \in \tilde{I}} g(x_i) \\
&\qquad \text{s.t. } \exists j \in [n] \text{ s.t. } \{f \in \mathcal{F} \text{ s.t. } f(x_i) = g(x_i) \forall i \in [I]\} \subset \{f \in \mathcal{F} : j \in \text{argmin}_{i \in [n]} f(x_i)\} \\
&= \upsilon^*_{loss}
\end{aligned}$$

where we used the definition of $\bar{g}$ as attaining the maximum in the definition of υ^*_{loss}. This completes the proof.

□

Here, we give an inefficient algorithm for cumulative loss minimization, Algorithm 3.

Proof of Theorem 5. Consider round t. Define the function $g : \{x_1, \dots, x_n\} \mapsto \mathcal{Y}$ in the following way:

$$g(x_i) = \begin{cases} \tilde{f}(x_i) \text{ s.t. } \tilde{f} = \arg\max_{\bar{f} \in \mathcal{F}_t} \frac{\bar{f}(x_i)}{|\mathcal{F}_t(x_i) \cup \mathcal{F}((x_i, \bar{f}(x_i)))|} & i \notin \{I_1, \dots, I_{t-1}\} \\ f^*(x_i) & \text{o/w} \end{cases}$$

Let $\tilde{I} = \{i_1, \dots, i_l\} \subset [n]$ attain the optimum in the following optimization problem:

$$\begin{aligned}
&\min_{l \in [n], i_1, \dots, i_l \in [n]} \sum_{j=1}^{l-1} g(x_{i_j}) && (7) \\
&\text{s.t. } \exists j \in [n] \text{ s.t. } \{f \in \mathcal{F}_t : f(x_i) = g(x_i) \forall i \in \{i_1, \dots, i_l\}\} && (8) \\
&\qquad \subset \{f \in \mathcal{F}_t : j \in \text{argmin}_{i \in [n]} f(x_i)\} && (9)
\end{aligned}$$

Define $\tilde{\mathcal{F}} = \{f \in \mathcal{F}_t : f(x_i) = g(x_i) \forall i \in \tilde{I}\}$. If $\tilde{F}$ is empty, set $I = \tilde{I}$. If $\tilde{F}$ is nonempty, then by definition of the above optimization problem, there exists $j_0 \in [n]$ such that for all $f \in \tilde{F}$, $j_0 \in \text{argmin}_{i \in [n]} f(x_i)$. Define $I = \tilde{I} \cup \{j_0\}$. **Step 1: Case 1.** Suppose that $|\mathcal{F}_t(x_{i_l}) \cup \mathcal{F}((x_i, g(x_{i_l}))| \leq \frac{|\mathcal{F}_t|}{2}$. Note that

$$\mathcal{F}_t \subset \{f \in \mathcal{F}_t : \exists i \in I \text{ s.t. } f(x_i) \neq g(x_i) \text{ or } i \in \text{argmin}_{j \in [n]} f(x_j)\}.$$

Then,

$$\begin{aligned}
|\mathcal{F}_t| &\leq |\{f \in \mathcal{F}_t : \exists i \in I \text{ s.t. } f(x_i) \neq g(x_i) \text{ or } i \in \text{argmin}_{j\in[n]} f(x_j)\}| \\
&\leq \sum_{i\in I} |\{f \in \mathcal{F}_t : f(x_i) \neq g(x_i) \text{ or } i \in \text{argmin}_{j\in[n]} f(x_j)\}| \\
&\leq \sum_{i\in I\backslash\{i_l\}} |\{f \in \mathcal{F}_t : f(x_i) \neq g(x_i) \text{ or } i \in \text{argmin}_{j\in[n]} f(x_j)\}| + \frac{|\mathcal{F}_t|}{2}
\end{aligned}$$

which implies that

$$\frac{|\mathcal{F}_t|}{2} \leq \sum_{i\in I\backslash\{i_l\}} |\{f \in \mathcal{F}_t : f(x_i) \neq g(x_i) \text{ or } i \in \text{argmin}_{j\in[n]} f(x_j)\}|. \tag{10}$$

Then, we have that

$$\begin{aligned}
&\frac{f^*(x_{I_t})}{|\mathcal{F}_t(x_{I_t}) \cup \mathcal{F}((x_i, f^*(x_{I_t}))|} \\
&\quad\leq \frac{g(x_{I_t})}{|\mathcal{F}_t(x_{I_t}) \cup \mathcal{F}((x_i, g(x_{I_t}))|} \qquad (11)\\
&\quad\leq \frac{\sum_{i\in I\backslash\{i_l\}} g(x_i)}{\sum_{i\in I\backslash\{i_l\}} |\{f \in \mathcal{F}_t : f(x_i) \neq g(x_i) \text{ or } i \in \text{argmin}_{j\in[n]} f(x_j)\}|} \qquad (12)\\
&\quad\leq 2\frac{\sum_{i\in I\backslash\{i_l\}} g(x_i)}{|\mathcal{F}_t|} \qquad (13)\\
&\quad\leq \frac{2(\upsilon^*_{loss} + \max_{i\in[n], f\in\mathcal{F}_t} f(x))}{|\mathcal{F}_t|} \qquad (14)
\end{aligned}$$

where (11) follows by the definition of g, (12) follows by definition of I_t and the mathematical fact that given positive numbers $a_1, \ldots, a_k$ and $b_1, \ldots, b_k$

$$\min_{i=1,\ldots,k} \frac{a_i}{b_i} \leq \frac{\sum_{i=1}^k a_i}{\sum_{i=1}^k b_i},$$

(13) follows by (10), and (14) follows by

$$\sum_{i\in\tilde{I}\backslash\{l_l\}} g(x_i) \leq \upsilon^*_{loss}$$

by the definition of the optimization problem in (7) and we used $g(x_{j_0}) \leq \max_{i\in[n], f\in\mathcal{F}_t} f(x_i))$ by definition of g. Rearranging, we have that

$$\begin{aligned}
f^*(x_{I_t}) &\leq \frac{2|\mathcal{F}_t(x_{I_t}) \cup \mathcal{F}((x_i, f^*(x_{I_t}))(\upsilon^*_{loss} + \max_{i\in[n], f\in\mathcal{F}_t} f(x_i))}{|\mathcal{F}_t|} \\
&= 2\frac{(|\mathcal{F}_t| - |\mathcal{F}_{t+1}|)(\upsilon^*_{loss} + \max_{i\in[n], f\in\mathcal{F}_t} f(x_i))}{|\mathcal{F}_t|}.
\end{aligned}$$

Step 2: Case 2. On the other hand, suppose that $|\mathcal{F}_t(x_{i_l}) \cup \mathcal{F}((x_i, g(x_{i_l}))| \geq \frac{|\mathcal{F}_t|}{2}$. Then,

$$\begin{aligned}
\frac{f^*(x_{I_t})}{|\mathcal{F}_t(x_{I_t}) \cup \mathcal{F}((x_i, f^*(x_{I_t}))|} &\leq \frac{g(x_{I_t})}{|\mathcal{F}_t(x_{I_t}) \cup \mathcal{F}((x_i, g(x_{I_t}))|} \\
&\leq \frac{g(x_{i_l})}{|\mathcal{F}_t(x_{i_l}) \cup \mathcal{F}((x_i, g(x_{i_l}))|} \qquad (15)\\
&\leq \frac{2\max_{i\in[n], f\in\mathcal{F}_t} f(x_i)}{|\mathcal{F}_t|} \qquad (16)
\end{aligned}$$

where (15) follows by the definition of I_t and (16) follows by the definition of g. Then, we have rearranging once again

$$f^*(x_{I_t}) \leq 2\frac{2|\mathcal{F}_t(x_{I_t}) \cup \mathcal{F}((x_i, f^*(x_{I_t}))|\text{max}_{i\in[n],f\in\mathcal{F}_t} f(x)}{|\mathcal{F}_t|}$$

Step 3: Putting it together. Thus, we have that

$$\begin{aligned}\sum_{t\geq 1} f^*(x_{I_t}) &\leq 2[v^*_{\mathit{loss}} + \text{max}_{i\in[n],f\in\mathcal{F}_t} f(x)] \sum_{t\geq 1} \frac{(|\mathcal{F}_t| - |\mathcal{F}_{t+1}|)}{|\mathcal{F}_t|} \\ &\leq 2[v^*_{\mathit{loss}} + \text{max}_{i\in[n],f\in\mathcal{F}_t} f(x)] \sum_{t\geq 1} \frac{1}{|\mathcal{F}_t|} + \frac{1}{|\mathcal{F}_t| - 1} + \ldots + + \frac{1}{|\mathcal{F}_{t+1}| + 1} \\ &\leq c2[v^*_{\mathit{loss}} + \text{max}_{i\in[n],f\in\mathcal{F}_t} f(x)] \ln(|\mathcal{F}|)\end{aligned}$$

where we used that $\frac{1}{|\mathcal{F}_t|} \leq \frac{1}{i}$ for $i \leq |\mathcal{F}_t|$ and that $\sum_{l=1}^{k} \frac{1}{l} = \Theta(\ln(k))$.

□

Proof of Theorem 4. For the sake of abbreviation, let $\mathbb{P}(\cdot) := \mathbb{P}_{r\sim\pi}(\cdot)$ and $\mathbb{P}_k(\cdot) := \mathbb{P}_{r\sim P_k}(\cdot)$.

Step 1: Bounding the loss in each phase

Consider round t in phase k. Define the function $g : \{x_1, \ldots, x_n\} \mapsto \mathcal{Y}$ in the following way:

$$g(x_i) = \begin{cases} \mathcal{D}[\tilde{r}(x_i)] \text{ s.t. } \tilde{r} = \arg\max_{\bar{r}\in\mathcal{R}_t} \frac{\bar{r}(x_i)}{\mathbb{P}_k(\mathcal{R}_t(x_i)\cup\mathcal{R}((x_i,\bar{f}(x_i)))} & i \notin \{I_1, \ldots, I_{t-1}\} \\ f^*(x_i) & \text{o/w} \end{cases}$$

Let $\tilde{I} = \{i_1, \ldots, i_l\} \subset [n]$ attain the optimum in the following optimization problem:

$$\begin{aligned} \min_{l\in[n],i_1,\ldots,i_l\in[n]} & \sum_{j=1}^{l-1} g(x_{i_j}) \\ & \text{s.t. } \exists j \in [n] : \{r \in \mathcal{R}_t : \mathcal{D}[r](x_i) = g(x_i)\, \forall i \in \{i_1, \ldots, i_l\}\} \\ & \qquad\qquad \subset \{r \in \mathbb{R}_t : j \in \text{argmin}_{i\in[n]} \mathcal{D}[r(x_i)]\} \end{aligned}$$

Define $\tilde{\mathcal{R}} = \{r \in \mathcal{R}_t : \mathcal{D}[r(x_i)] = g(x_i) \forall i \in \tilde{I}\}$. If $\tilde{\mathcal{R}}$ is empty, set $I = \tilde{I}$. If $\tilde{\mathcal{R}}$ is nonempty, then by definition of the above optimization problem, there exists $j_0 \in [n]$ such that for all $r \in \tilde{\mathcal{R}}$, $j_0 \in \text{argmin}_{i\in[n]} \mathcal{D}[r(x_i)]$. Define $I = \tilde{I} \cup \{j_0\}$.

Step 2.1: Case 1. Suppose that $\mathbb{P}_k(\mathcal{R}_t(x_{i_l}) \cup \mathcal{R}((x_i, g(x_{i_l}))) \leq \frac{\mathbb{P}_k(\mathcal{R}_t)}{2}$. Note that

$$\mathcal{R}_t \subset \{r \in \mathcal{R}_t : \exists i \in I \text{ s.t. } \mathcal{D}[r(x_i)] \neq g(x_i) \text{ or } i \in \text{argmin}_{j\in[n]} \mathcal{D}[r(x_j)]\}.$$

Then,

$$\begin{aligned}\mathbb{P}_k(\mathcal{R}_t) &\leq \mathbb{P}_k(\{r \in \mathcal{R}_t : \exists i \in I \text{ s.t. } \mathcal{D}[r(x_i)] \neq g(x_i) \text{ or } i \in \text{argmin}_{j\in[n]} \mathcal{D}[r(x_j)]\}) \\ &\leq \sum_{i\in I} \mathbb{P}_k(\{r \in \mathcal{R}_t : \exists i \in I \text{ s.t. } \mathcal{D}[r(x_i)] \neq g(x_i) \text{ or } i \in \text{argmin}_{j\in[n]} \mathcal{D}[r(x_j)]\}) \\ &\leq \sum_{i\in I\setminus\{i_l\}} \mathbb{P}_k(\{r \in \mathcal{R}_t : \exists i \in I \text{ s.t. } \mathcal{D}[r(x_i)] \neq g(x_i) \text{ or } i \in \text{argmin}_{j\in[n]} \mathcal{D}[r(x_j)]\}) \\ &+ \frac{\mathbb{P}_k(\mathcal{R}_t)}{2}\end{aligned}$$

which implies that

$$\frac{\mathbb{P}_k(\mathcal{R}_t)}{2} \leq \sum_{i\in I\setminus\{i_l\}} \mathbb{P}_k(\{r \in \mathcal{R}_t : \exists i \in I \text{ s.t. } \mathcal{D}[r(x_i)] \neq g(x_i) \text{ or } i \in \text{argmin}_{j\in[n]} \mathcal{D}[r(x_j)]\}).$$

Then, we have that

$$\begin{align}
&\frac{f^*(x_{I_t})}{\mathbb{P}_k(\mathcal{R}_t(x_{I_t}) \cup \mathcal{R}((x_i, f^*(x_{I_t})))} \tag{17}\\
&\leq \frac{g(x_{I_t})}{\mathbb{P}_k(\mathcal{R}_t(x_{I_t}) \cup \mathcal{R}((x_i, g(x_{I_t})))} \tag{18}\\
&\leq \frac{\sum_{i \in I \setminus \{i_l\}} g(x_i)}{\sum_{i \in I \setminus \{i_l\}} \mathbb{P}_k(\{r \in \mathcal{R}_t : \exists i \in l \text{ s.t. } \mathcal{D}[r(x_i)] \neq g(x_i) \text{ or } i \in \text{argmin}_{j \in [n]} \mathcal{D}[r(x_j)]\})} \notag\\
&\leq 2 \frac{\sum_{i \in I \setminus \{i_l\}} g(x_i)}{\mathbb{P}_k(\mathcal{R}_t)} \notag\\
&= 2 \frac{\sum_{i \in \tilde{I} \setminus \{i_l\} \cup \{j_0\}} g(x_i)}{\mathbb{P}_k(\mathcal{R}_t)} \notag\\
&\leq \frac{2(\upsilon^*_{loss} + \text{max}_{i \in [n], r \in \mathcal{R}_t} \mathcal{D}[r(x_i)])}{\mathbb{P}_k(\mathcal{R}_t)} \notag
\end{align}$$

where the above series of inequalities follows by the same arguments used to show the inequalities (11) to (14) Rearranging, we have that

$$\begin{aligned}
f^*(x_{I_t}) &\leq \frac{2\mathbb{P}_k(\mathcal{R}_t(x_{I_t}) \cup \mathcal{R}((x_i, f^*(x_{I_t})))[\upsilon^*_{loss} + \text{max}_{i \in [n], r \in \mathcal{R}_t} \mathcal{D}[r(x_i)]]}{\mathbb{P}_k(\mathcal{R}_t)} \\
&= 2 \frac{(\mathbb{P}_k(\mathcal{R}_t) - \mathbb{P}_k(\mathcal{R}_{t+1}))[\upsilon^*_{loss} + \text{max}_{i \in [n], r \in \mathcal{R}_t} \mathcal{D}[r(x_i)]]}{\mathbb{P}_k(\mathcal{R}_t)}.
\end{aligned}$$

Step 2.2: Case 2. On the other hand, suppose that $\mathbb{P}_k(\mathcal{R}_t(x_{i_l}) \cup \mathcal{R}((x_i, g(x_{i_l}))) \geq \frac{\mathbb{P}_k(\mathcal{R}_t)}{2}$. Then,

$$\begin{align}
\frac{f^*(x_{I_t})}{\mathbb{P}_k(\mathcal{R}_t(x_{I_t}) \cup \mathcal{R}((x_i, f^*(x_{I_t})))} &\leq \frac{g(x_{I_t})}{\mathbb{P}_k(\mathcal{R}_t(x_{I_t}) \cup \mathcal{R}((x_i, g(x_{I_t})))} \tag{19}\\
&\leq \frac{g(x_{i_l})}{\mathbb{P}_k(\mathcal{R}_t(x_{i_l}) \cup \mathcal{R}((x_i, g(x_{i_l})))} \notag\\
&\leq \frac{2\text{max}_{i \in [n], r \in \mathcal{R}_t} \mathcal{D}[r(x_i)]}{\mathbb{P}_k(\mathcal{R}_t)} \notag
\end{align}$$

where the above series of inequalities follows by the same arguments used to show (15) to (11). These imply

$$f^*(x_{I_t}) \leq 2 \frac{\mathbb{P}_k(\mathcal{R}_t(x_{I_t}) \cup \mathcal{R}((x_i, f^*(x_{I_t})))\text{max}_{i \in [n], r \in \mathcal{R}_t} \mathcal{D}[r(x_i)]}{\mathbb{P}_k(\mathcal{R}_t)}$$

Thus, we have that

$$\begin{aligned}
&\sum_{t \geq t_k : \mathbb{P}_k(\mathcal{R}_t) \geq 1/2n} f^*(x_{I_t}) \\
&\leq 2[\upsilon^*_{loss} + \text{max}_{i \in [n], r \in \mathcal{R}_t} \mathcal{D}[r(x_i)]] \sum_{t \geq t_k : \mathbb{P}_k(\mathcal{R}_t) \geq 1/2n} \frac{\mathbb{P}_k(\mathcal{R}_t) - \mathbb{P}_k(\mathcal{R}_{t+1})}{\mathbb{P}_k(\mathcal{R}_t)} \\
&= 2[\upsilon^*_{loss} + \text{max}_{i \in [n], r \in \mathcal{R}_t} \mathcal{D}[r(x_i)]] \sum_{t \geq t_k : \mathbb{P}_k(\mathcal{R}_t) \geq 1/2n} \frac{2n(\mathbb{P}_k(\mathcal{R}_t) - \mathbb{P}_k(\mathcal{R}_{t+1}))}{2n\mathbb{P}_k(\mathcal{R}_t)} \\
&\leq 2[\upsilon^*_{loss} + \text{max}_{i \in [n], r \in \mathcal{R}_t} \mathcal{D}[r(x_i)]] \\
&\cdot [1 + \sum_{t \geq t_k : \mathbb{P}_k(\mathcal{R}_{t+1}) \geq 1/2n} \frac{1}{2n\mathbb{P}_k(\mathcal{R}_t)} + \frac{1}{2n\mathbb{P}_k(\mathcal{R}_t) - 1} + \ldots + \frac{1}{2n\mathbb{P}_k(\mathcal{R}_{t+1}) + 1}] \\
&\leq c \ln(n)[\upsilon^*_{loss} + \text{max}_{i \in [n], r \in \mathcal{R}_t} \mathcal{D}[r(x_i)]]
\end{aligned}$$

where we used that $\frac{1}{2n\mathbb{P}_k(\mathcal{R}_t)} \leq \frac{1}{i}$ for $i \leq 2n\mathbb{P}_k(\mathcal{R}_t)$ and that $\sum_{l=1}^{k} \frac{1}{l} = \Theta(\ln(k))$. This bounds the cumulative loss incurred in round k.

Step 2: Finish the proof. We have that

$$\frac{\mathbb{P}(C_{l_0}(O_{t_k}))}{2} \geq \mathbb{P}(C_{l_0}(O_{t_{k+1}})) \tag{20}$$

by the same argument used in the proof of Theorem 2. Thus, the loss incurred by the algorithm is at most

$$\ln(n)\ln(\frac{1}{\mathbb{P}(S_{\text{min}})})[\upsilon^*_{\textit{loss}} + \text{max}_{i\in[n],r\in\mathcal{R}_t}\mathcal{D}[r(x_i)]].$$

□

D.1 Other Cumulative Loss Minimization Results

Proof of Proposition 7. Fix $f^* \in \mathcal{F}$. Let $T_0 = n$. There exists a round $\bar{T} \leq T_0$ at which Algorithm 3 identifies $i_* \in \min_{j\in[n]} f^*(x_j)$, at which point it only plays i_*. Note that since the algorithm is deterministic and there is no noise $\bar{T}$ is deterministic. We have that

$$\sum_{t=1}^{T_0}(f^*(x_{I_t}) - \min_{j\in[n]} f^*(x_j)) = \sum_{t=1}^{\bar{T}} f^*(x_{I_t}) - \bar{T}\min_{j\in[n]} f^*(x_j)$$

since no regret is incurred for $t > \bar{T}$.

Case 1: Suppose that $\bar{T} \geq 4\upsilon^*_{\textit{best}}\ln(|\mathcal{F}|)$. First, we note that by Theorem 5, we have that

$$\begin{aligned}
\sum_{t=1}^{\bar{T}} f^*(x_{I_t}) &\leq 2(\upsilon^*_{\textit{loss}} + \text{max}_{i\in[n],f\in\mathcal{F}}f^*(x_i))\ln(|\mathcal{F}|) &(21)\\
&\leq 2(\upsilon^*_{\textit{best}}\text{max}_{y\in\mathcal{Y}}y + \text{max}_{i\in[n],f\in\mathcal{F}}f^*(x_i))\ln(|\mathcal{F}|) &(22)\\
&\leq 4\upsilon^*_{\textit{best}}\ln(|\mathcal{F}|)\text{max}_{y\in\mathcal{Y}}y &(23)
\end{aligned}$$

where (22) follows since $\upsilon^*_{\textit{loss}} \leq \upsilon^*_{\textit{best}}\text{max}_{y\in\mathcal{Y}}y$ by Proposition 6, and (23) follows since $\upsilon^*_{\textit{best}} \geq 1$ by assumption on $\mathcal{F}_{\mathcal{R}}$. Since $\bar{T} \geq 4\upsilon^*_{\textit{best}}\ln(|\mathcal{F}_{\mathcal{R}}|)$ and $\min_{y\in\mathcal{Y}} y \geq 0$, we have that

$$\begin{aligned}
\sum_{t=1}^{\bar{T}} f^*(x_{I_t}) - \bar{T}\min_{j\in[n]} f^*(x_j) &\leq 4\upsilon^*_{\textit{best}}\ln(|\mathcal{F}_{\mathcal{R}}|)\text{max}_{y\in\mathcal{Y}}y - \bar{T}\min_{j\in[n]} f^*(x_j)\\
&\leq 4\upsilon^*_{\textit{best}}\ln(|\mathcal{F}|)\text{max}_{y\in\mathcal{Y}}y - 4\upsilon^*_{\textit{best}}\ln(|\mathcal{F}|)\min_{y\in\mathcal{Y}} y\\
&= 4\upsilon^*_{\textit{best}}\ln(|\mathcal{F}_{\mathcal{R}}|)\Delta_{\text{max}}.
\end{aligned}$$

Case 2: Now, suppose that $\bar{T} \leq 4\upsilon^*_{\textit{best}}\ln(|\mathcal{F}_{\mathcal{R}}|)$. Then,

$$\begin{aligned}
\sum_{t=1}^{\bar{T}} f^*(x_{I_t}) - \bar{T}\min_{j\in[n]} f^*(x_j) &\leq \bar{T}(\text{max}_{y\in\mathcal{Y}}y - \min_{j\in[n]} f^*(x_j))\\
&\leq \bar{T}(\text{max}_{y\in\mathcal{Y}}y - \min_{y\in\mathcal{Y}} y)\\
&\leq 4\upsilon^*_{\textit{best}}\ln(|\mathcal{F}|)\Delta_{\text{max}}
\end{aligned}$$

□

Proof of Proposition 8. Let $\bar{y} > \tilde{y} > y_*$. We have $\Delta_{\text{min}} = \tilde{y} - y_*$ and $\Delta_{\text{max}} = \bar{y} - y_*$. Let m be even and $n = m + m/2$. Let $\mathcal{F} = \{f_1, \ldots, f_m\}$ where

$$f_j(x_i) = \begin{cases} \bar{y} & i \in [m/2] \setminus \{2j-1, 2j\}\\ y_* & i \in \{2j-1, 2j\}\\ \tilde{y} & i \in ([m+m/2] \setminus [m/2]) \setminus \{m/2 + j\}\\ y_* & i = m/2 + j \end{cases}$$

The greedy algorithm in the regret minimization algorithm from [11] finds the best arm using a greedy best arm identification algorithm that greedily removes functions from the version space. Once if has found the best arm, it pulls the best arm for the remainder of the game. Each query in $[m/2]$ removes 2 functions while each query in $[m + m/2] \setminus [m/2]$ removes 1 function. Thus, the regret minimization algorithm from [11] would use the queries in $[m/2]$ to remove functions, incurring a regret of Δ_{max} for each query. In the worst case, it would incur a regret of $\Omega(\Delta_{\text{max}} m) = \Omega(\Delta_{\text{max}} n)$. On the other hand, Algorithm 3 would query only in $[m+m/2] \setminus [m/2]$ if $\Delta_{\text{max}} >> \Delta_{\min}$, incurring a regret of $O(m\Delta_{\min}) = O(n\Delta_{\min})$. □

Proof of Proposition 9.

$$\begin{aligned}
\text{max}_{f\in\mathcal{F}}\text{loss}(\bar{\mathcal{A}}; f; T_0) &= \text{max}_{f\in\mathcal{F}} \sum_{t=1}^{T_0} f(x_{I_t}) \\
&= \text{max}_{f\in\mathcal{F}} \sum_{t=1}^{T_0} (f(x_{I_t}) - \min_{j\in[n]} f(x_j)) + T_0 \min_{j\in[n]} f(x_j) \\
&= \text{max}_{f\in\mathcal{F}}\text{regret}(\mathcal{A}; f; T_0) + T_0 \min_{j\in[n]} f(x_j) \\
&\leq \text{max}_{f\in\mathcal{F}}\text{regret}(\mathcal{A}; f; T_0) + \text{max}_{f\in\mathcal{F}} T_0 \min_{j\in[n]} f(x_j) \\
&\leq c \min_{\mathcal{A}} \text{max}_{f\in\mathcal{F}}\text{regret}(\mathcal{A}; f; T_0) + \text{max}_{f\in\mathcal{F}} T_0 \min_{j\in[n]} f(x_j) \\
&\leq (c+1) \min_{\mathcal{A}} \text{max}_{f\in\mathcal{F}}\text{loss}(\mathcal{A}; f; T_0)
\end{aligned}$$

where in the last line we used that $\text{regret}(\mathcal{A}; f; T_0) \leq \text{loss}(\mathcal{A}; f; T_0)$ since $\min_{y\in\mathcal{Y}} y \geq 0$ by assumption, which implies that $\min_{\mathcal{A}} \text{max}_{f\in\mathcal{F}}\text{regret}(\mathcal{A}; f; T_0) \leq \min_{\mathcal{A}} \text{max}_{f\in\mathcal{F}}\text{loss}(\mathcal{A}; f; T_0)$, and $\text{max}_{f\in\mathcal{F}} T_0 \min_{j\in[n]} f(x_j) \leq \min_{\mathcal{A}} \text{max}_{f\in\mathcal{F}}\text{loss}(\mathcal{A}; f; T_0)$. □

Proof of Proposition 6. Fix $g : \mathcal{X} \mapsto \mathcal{Y}$. By definition of v^*_{best}, there exists $\bar{I} \subset [n]$ such that $|\bar{I}| \leq v^*_{best}$ and there exists $j_0 \in [n]$ such that

$$\{f \in \mathcal{F} : f(x_i) = g(x_i) \, \forall i \in \bar{I}\} \subset \{f \in \mathcal{F} : j_0 \in \text{argmin}_{k\in[n]} f(x_k)\}$$

This immediately implies that

$$\begin{aligned}
&\min_{l\in[n], i_1, \ldots, i_l} \sum_{j=1}^{l-1} g(x_{i_j}) \\
&\qquad \text{s.t. } \exists j \in [n] \text{ s.t. } \{f \in \mathcal{F} : f(x_i) = g(x_i) \forall i \in \{i_1, \ldots, i_l\} \\
&\qquad\qquad\qquad \subset \{f \in \mathcal{F} : j \in \text{argmin}_{i\in[n]} f(x_i)\} \\
&\qquad \leq \sum_{i\in\bar{I}} g(x_i) \\
&\qquad \leq |\bar{I}| \text{max}_{y\in\mathcal{Y}} y \\
&\qquad \leq v^*_{best} \text{max}_{y\in\mathcal{Y}} y
\end{aligned}$$

□

Proof of Proposition 10. Suppose that $n = 2$ and $\mathcal{F} = \{f_1, f_2\}$ such that $f_1(x_1) = \Delta$, $f_1(x_2) = \Delta - 1$, $f_2(x_1) = 1$, and $f_2(x_2) = \Delta/2 + 1$. Suppose f_1 is true. If one queries x_1, then one obtains a regret of 1 and a loss of Δ. If one queries x_2, one obtains a regret of 0 and a loss of $\Delta - 1$. Now, suppose f_2 is true. If one queries x_1, then one obtains a regret of 0 and a loss of 1. If one queries x_2, one obtains a regret of $\Delta/2$ and a loss of $\Delta/2 + 1$. Thus, to minimize worst-case regret, one must query x_1, obtaining a worst case regret of 1. To minimize worst-case loss, one must query x_2, obtaining a worst-case loss of $\Delta - 1$. But, querying x_2 has a regret of $\Delta/2$, completing the proof. □

```
$k \longleftarrow 1, t_1 = 1, O_1 \longleftarrow \emptyset$;
for $t = 1, 2, \ldots$ do
    Let $I_t \in \arg\max_{i \in [n] \setminus O_t} \min_{j \in [k]} \mathbb{P}_{f \sim P_k}(f \in \mathcal{F}_t(x_i) \cup \mathcal{F}_t((x_i, L_j)))$;
    Query $x_{I_t}$ and observe $y_{I_t}$ and set $O_{t+1} \longleftarrow O_t \cup \{I_t\}$ ;
    Let $\mathcal{F}_{t+1} \longleftarrow \mathcal{F}_t \setminus (\mathcal{F}_t(x_{I_t}) \cup \mathcal{F}_t((x_{I_t}, L(y_{I_t}))))$;
    if $\mathbb{P}_{f \sim P_k}(f \in \mathcal{F}_{t+1}) \leq \frac{1}{2n}$ then
        $P_{k+1} \longleftarrow \frac{1}{n - |O_{t_k}|} \sum_{l \in [n] \setminus O_{t_k}} \pi_{C_l(O_{t_k})}$;
        $k \longleftarrow k + 1$;
        $t_k \longleftarrow t + 1$;
```

Algorithm 4: `GRAILS` for Continuous Output Space

E Extension to Continuous Output Space

Now, we turn to the continuous output setting where we assume that the scores are continuous, that is, $y_1, \ldots, y_n \in [-1, 1]$. Let $\mathcal{F} \subset [-1, 1]^{\mathcal{X}}$ and suppose that there exists $f^* \in \mathcal{F}$ such that $f^*(x_i) = y_i$ for all $i = 1, \ldots, n$. Let $\epsilon > 0$. The goal is to identify $j \in [n]$ such that $f^*(x_j) \leq \min_{i \in [n]} f^*(x_i) + \epsilon$.

Let $\delta > 0$ and $L_i = [-1 + (i-1)\delta, -1 + i\delta)$ for $i = 1, 2, \ldots, \frac{2}{\delta} =: p$. $L_1, \ldots, L_p$ is a grid on the output space $[-1, 1]$. For $\tilde{y} \in [-1, 1]$, let $L(\tilde{y})$ denote the interval L_j such that $\tilde{y} \in L_j$. Furthermore, define $\bar{L}_j := \frac{1}{2}(-1 + (j-1)\delta + (-1 + j\delta))$, the midpoint of the L_j interval. Define

$$\begin{aligned} \mathcal{F}((x_i, L_j)) &= \{f \in \mathcal{F} : f(x_i) \notin L_j\} \\ \mathcal{F}(x_i) &= \{f \in \mathcal{F} : f(x_i) \leq \min_{j \in [n]} f(x_j) + \epsilon\}. \end{aligned}$$

We overload notation, defining

$$C_l(O_t) := \{f \in \mathcal{F} : f(x_i) \in L(y_i) \forall i \in O_t\} \cap \{f(x_i) \leq f(x_l) + \epsilon \forall i \in O_t\}.$$

Define the following extended teaching dimension notion for the continuous output setting:

$$\begin{aligned} &\upsilon_{\epsilon,\delta} := \max\nolimits_{g : \mathcal{X} \mapsto \mathbb{R}} \min_{I \subset [n]} |I| \\ &\quad \text{s.t. } \exists j \in [n] : \{f \in \mathcal{F} : |f(x_i) - g(x_i)| \leq \delta \forall i \in I\} \subset \{f(x_j) \leq \min_{l \in [n]} f(x_l) + \epsilon\}. \end{aligned}$$

Define for every $f \in \mathcal{F}$ the set

$$S_f = \{f' \in \mathcal{F} : L(f'(x_i)) = L(f(x_i)) \forall i \in [n]\}.$$

The sets S_f induce a partition of $\mathcal{F}$.

Theorem 7. *Let $f^* \in \mathcal{F}$ such that $f^*(x_i) = y_i$ for all $i \in [n]$. After $c \ln(\frac{1}{\mathbb{P}_\pi(S_{f^*})}) \upsilon^*_{\epsilon,\delta}$ queries, Algorithm 4 has queried some $i \in [n]$ satisfying $f^*(x_i) \leq \min_{j \in [n]} f^*(x_j) + \epsilon$*

We note that our algorithm for loss minimization can be extended to the continuous output space with a similar guarantee.

Proof of Theorem 7. For the sake of abbreviation, let $\mathbb{P}(\cdot) := \mathbb{P}_{f \sim \pi}(\cdot)$ and $\mathbb{P}_k(\cdot) := \mathbb{P}_{f \sim P_k}(\cdot)$.

Step 1: The Algorithm 1 breaks time into phases $[t_1, t_2]$, $[t_2, t_3]$, $\cdots$. In each phase k, the algorithm samples functions from a distinct distribution P_k. Consider round k, let $t \geq t_k$. Define $\theta \in \{w_1, \ldots, w_L\}^n$ by

$$\theta_i = \begin{cases} y_i & i \in O_t \\ \text{argmin}_{\tilde{y} \in \{\bar{L}_1, \ldots, \bar{L}_p\}} \mathbb{P}_k(f \in \mathcal{F}_t(x_i) \cup \mathcal{F}_t((x_i, L(\tilde{y})))) & \text{o/w} \end{cases}.$$

By definition of $\upsilon^*_{\epsilon,\delta}$, there exists $\tilde{I} = \{i_1, \ldots, i_{\upsilon^*}\}$ such that there exists $j_0 \in [n]$ such that $\forall f \in \mathcal{F}_t$ that satisfy $|f(x_i) - \theta_i| \leq \delta$ for all $i \in \tilde{I}$, it holds that $f(x_{j_0}) - \epsilon \leq \min_{l \in [n]} f(x_l)$. Define

$I = \{i_1, \ldots, i_{\upsilon^*}, j_0\}$. Then, we have that

$$\begin{aligned}\mathcal{F}_t &= \{f \in \mathcal{F}_t : \exists i \in I \text{ s.t. } f(x_i) \leq \min_{l\in[n]} f(x_l) + \epsilon \text{ or } |f(x_i) - \theta_i| > \delta\} \\ &= \{f \in \mathcal{F}_t : \exists i \in I \text{ s.t. } f(x_i) \leq \min_{l\in[n]} f(x_l) + \epsilon \text{ or } f(x_i) \notin L(\theta_i)\}\end{aligned}$$

where the last equality follows since by definition of $L(\cdot)$, $|f(x_i)-\theta_i| > \delta$ implies that $f(x_i) \notin L(\theta_i)$. From this, it follows that

$$\begin{aligned}\mathbb{P}_k(\mathcal{F}_t) &= \mathbb{P}_k(\{f \in \mathcal{F}_t : \exists i \in I \text{ s.t. } f(x_i) \leq \min_{l\in[n]} f(x_l) + \epsilon \text{ or } f(x_i) \notin L(\theta_i)\}) \\ &\leq \sum_{i\in I} \mathbb{P}_k(\{f \in \mathcal{F}_t : f(x_i) \leq \min_{l\in[n]} f(x_l) + \epsilon \text{ or } f(x_i) \notin L(\theta_i)\}) \\ &\leq (\upsilon^*_{\epsilon,\delta} + 1)\text{max}_{i\in I}\mathbb{P}_k(\{f \in \mathcal{F}_t : f(x_i) \leq \min_{l\in[n]} f(x_l) + \epsilon \text{ or } f(x_i) \notin L(\theta_i)\}) \\ &= (\upsilon^*_{\epsilon,\delta} + 1)\mathbb{P}_k(\{f \in \mathcal{F}_t : f(x_{I_t}) \leq \min_{l\in[n]} f(x_l) + \epsilon \text{ or } f(x_{I_t}) \notin L(\theta_{I_t})\}) \quad (24) \\ &\leq (\upsilon^*_{\epsilon,\delta} + 1)\mathbb{P}_k(\{f \in \mathcal{F}_t : f(x_{I_t}) \leq \min_{l\in[n]} f(x_l) + \epsilon \text{ or } f(x_{I_t}) \notin L(y_{I_t})\}) \quad (25) \\ &= (\upsilon^*_{\epsilon,\delta} + 1)\mathbb{P}_k(\mathcal{F}_t(x_{I_t}) \cup \mathcal{F}_t((x_{I_t}, L(y_{I_t})))\end{aligned}$$

where line (24) comes from the definition of I_t and the line (25) comes from the definition of θ. Noticing that

$$\begin{aligned}\mathcal{F}_t &= (\mathcal{F}_t(x_{I_t}) \cup \mathcal{F}_t((x_{I_t}, L(y_{I_t}))) \cup \mathcal{F}_t \setminus (\mathcal{F}_t(x_{I_t}) \cup \mathcal{F}_t((x_{I_t}, L(y_{I_t}))) \\ &= (\mathcal{F}_t(x_{I_t}) \cup \mathcal{F}_t((x_{I_t}, L(y_{I_t}))) \cup \mathcal{F}_{t+1},\end{aligned}$$

we have that

$$\mathbb{P}_k(\mathcal{F}_{t+1}) = \mathbb{P}_k(\mathcal{F}_t) - \mathbb{P}_k(\mathcal{F}_t(x_{I_t}) \cup \mathcal{F}_t((x_{I_t}, L(y_{I_t}))) \leq (1 - \frac{1}{\upsilon^*_{\epsilon,\delta} + 1})\mathbb{P}_k(\mathcal{F}_t).$$

Unraveling this recursive statement, we have that

$$\mathbb{P}_k(\mathcal{F}_{t+1}) \leq (1 - \frac{1}{\upsilon^* + 1})^{t-t_k}.$$

Thus, we see that in phase k after $s_k := O(\ln(n)\upsilon^*_{\epsilon,\delta})$ rounds , we have that

$$\mathbb{P}_k(\mathcal{F}_{t_k+s_k}) \leq \frac{1}{2n}. \quad (26)$$

The rest of the proof is similar to the proof of Theorem 2.

□

F Active Classification and Extensions to other Objectives

In this Section, we apply our framework to other problems such as active classification and a problem for identifying a sufficiently good arm, defined shortly. We give a result for active classification and give brief sketches for the other objectives since their treatment is similar to the problems already studied in this paper.

Active Classification. We recall the active classification setting. Let $\mathcal{X}$ denote the input space and $\mathcal{Y} = \{-1, 1\}$ the output space. Let $x_1, \ldots, x_n \in \mathcal{X}$ be a fixed pool with associated scores $y_1, \ldots, y_n \in \mathcal{Y}$. At each round t, the learner selects (or queries) $I_t \in [n]$ and observes y_{I_t}. The goal is to identify $\text{argmin}_{f\in\mathcal{R}_\mathcal{R}} \sum_{i=1}^n \mathbb{1}\{y_i \neq f(x_i)\}$ using as few queries as possible. Recall that under realizability there exists $f^* \in \mathcal{F}_\mathcal{R}$ such that $f^*(x_i) = y_i$ for all $i \in [n]$.

We note that for active classification, the version space is convex, and so we may use a simpler version space. We also note that STOP($\mathcal{F}_\mathcal{R}$,O_t) can check whether there are $f, f' \in \mathcal{F}_t$ such that $f \neq f'$ by solving the following optimization problem for $j \in [n] \setminus O_t$ and $y \in \mathcal{Y}$

$$\exists f \text{ s.t. } f(x_i) = y_i \, \forall i \in O_t \wedge f(x_j) = y. \quad (27)$$

```
$k \longleftarrow 1, t_1 = 1, O_1 \longleftarrow \emptyset$;
for $t = 1, 2, \ldots$ do
    Let $I_t \in \arg\max_{i \in [n] \setminus O_t} \min_{y \in \mathcal{Y}} \mathbb{P}_{r \sim \pi_{\mathcal{R}_t}}(\mathcal{R}_t((x_i, y)))$;
    Query $x_{I_t}$ and observe $y_{I_t}$ and set $O_{t+1} \longleftarrow O_t \cup \{I_t\}$ ;
    Let $\mathcal{R}_{t+1} \longleftarrow \mathcal{R}_t \setminus \mathcal{R}_t((x_{I_t}, y_{I_t}))$;
    if STOP($\mathcal{F}_{\mathcal{R}}$,$O_t$) then
        return $\text{argmin}_{i \in O_t} y_i$
```

Algorithm 5: `GRAILS` for Active Classification

If there exists j such that there exists $f, f' \in \mathcal{F}_t$ and $f(x_j) \neq f(x_j)$, then the Algorithm does not terminate. If not, then the algorithm terminates. We note that (27) is a convex optimization problem for function classes like linear functions, kernel methods, and convex functions.

We introduce the extended teaching dimension notion [21], which has previously been shown to be a lower bound for active classification:

$$\upsilon^*_{\textit{class}}(\mathcal{F}) = \max_{g : \mathcal{X} \mapsto \mathcal{Y}} \min_{I \subset [n]} |I|$$

$$\text{s.t. } |\{f \in \mathcal{R} : \mathcal{D}[f(x_i)] = g(x_i) \forall i \in I\}| \leq 1.$$

Recall the definition $S_{\min} = \text{argmin}_{S_f : f \in \mathcal{F}_{\mathcal{R}}, \mathbb{P}_\pi(S_f) > 0} \mathbb{P}_\pi(S_f)$.

Theorem 8. *After $c \log(\frac{1}{\mathbb{P}(S_{\min})}) \upsilon^*_{\textit{class}}(\mathcal{F}_{\mathcal{R}})$ queries, the version space only contains S_{f^*}.*

Proof of Theorem 8. For the sake of abbreviation, let $\mathbb{P}(\cdot) := \mathbb{P}_{f \sim \pi}(\cdot)$ and $\mathbb{P}_t(\cdot) := \mathbb{P}_{f \sim \pi_t}(\cdot)$ where $\pi_t := \pi_{\mathcal{R}_t}$.

Step 1: Consider round t. Define

$$\theta_i = \begin{cases} y_i & i \in O_t \\ \text{argmin}_{y \in \mathcal{Y}} \mathbb{P}(r \in \mathcal{R}_t((x_i, y))) & \text{o/w} \end{cases}.$$

By definition of the extended teaching dimension $\upsilon^*_{\textit{class}}(\mathcal{F}_{\mathcal{R}})$, there exists $I = \{i_1, \ldots, i_{\upsilon^*_{\textit{class}}}\}$ such that

$$|\mathcal{G}_t| \leq 1$$

where

$$\mathcal{G}_t := \{\mathcal{D}[r] \in \mathcal{R}_t : \mathcal{D}[r(x_i)] = \theta_i \; \forall i \in I\}. \tag{28}$$

From this, it follows that

$$\begin{aligned}
\mathbb{P}(\mathcal{R}_t \setminus \mathcal{G}_t) &= \mathbb{P}(\{r \in \mathcal{R}_t : \exists i \in I \text{ s.t. } \mathcal{D}[r(x_i)] \neq \theta_i\}) \\
&\leq \sum_{i \in I} \mathbb{P}(\{r \in \mathcal{R}_t : \mathcal{D}[r(x_i)] \neq \theta_i\}) \\
&\leq \upsilon^*_{\textit{class}} \max_{i \in I} \mathbb{P}(\{r \in \mathcal{R}_t : \mathcal{D}[r(x_i)] \neq \theta_i\}) \\
&= \upsilon^*_{\textit{class}} \mathbb{P}(\{r \in \mathcal{R}_t : I_t \in \text{argmin}_{l \in [n]} \mathcal{D}[r(x_l)] \text{ or } \mathcal{D}[r(x_i)] \neq \theta_{I_t}\}) \qquad (29) \\
&\leq \upsilon^*_{\textit{class}} \mathbb{P}(\{r \in \mathcal{R}_t : \mathcal{D}[r(x_i)] \neq y_{I_t}\}) \qquad (30) \\
&= \upsilon^*_{\textit{class}} \mathbb{P}(\mathcal{R}_t((x_{I_t}, y_{I_t})))
\end{aligned}$$

where line (29) comes from the definition of I_t, the definition of θ, and the fact that

$$\text{argmin}_i \mathbb{P}(r \in \mathcal{R}_t((x_i, \theta_i))) = \text{argmin}_i \frac{\mathbb{P}(r \in \mathcal{R}_t((x_i, \theta_i)))}{\mathbb{P}(\mathcal{R}_t)} = \text{argmin}_i \mathbb{P}_t(r \in \mathcal{R}_t((x_i, \theta_i)))$$

and the line (30) comes from the definition of θ. Noticing that

$$\begin{aligned}
\mathcal{R}_t &= \mathcal{R}_t((x_{I_t}, y_{I_t}) \cup \mathcal{R}_t \setminus \mathcal{R}_t((x_{I_t}, y_{I_t})) \\
&= \mathcal{R}_t((x_{I_t}, y_{I_t}) \cup \mathcal{R}_{t+1},
\end{aligned}$$

we have that

$$\mathbb{P}(\mathcal{R}_{t+1}) = \mathbb{P}(\mathcal{R}_t) - \mathbb{P}(\mathcal{R}_t((x_{I_t}, y_{I_t}))) \leq \mathbb{P}(\mathcal{R}_t) - \frac{1}{\upsilon^*_{class}} \mathbb{P}(\mathcal{R}_t \setminus \mathcal{G}_t). \tag{31}$$

We have that

$$\begin{aligned}\sum_{t=1}^{\infty} \mathbb{1}\{\mathcal{R}_t \neq S_{f^*}\} = \sum_{t=1}^{\infty} & \mathbb{1}\{\mathcal{R}_t \neq S_{f^*} \wedge \mathbb{P}(\mathcal{G}_t) \leq \frac{1}{2}\mathbb{P}(\mathcal{R}_t)\} \\ & + \mathbb{1}\{\mathcal{R}_t \neq S_{f^*} \wedge \mathbb{P}(\mathcal{G}_t) > \frac{1}{2}\mathbb{P}(\mathcal{R}_t) \wedge \mathcal{G}_t \neq S_{f^*}\} \\ & + \mathbb{1}\{\mathcal{R}_t \neq S_{f^*} \wedge \mathbb{P}(\mathcal{G}_t) > \frac{1}{2}\mathbb{P}(\mathcal{R}_t) \wedge \mathcal{G}_t = S_{f^*}\}.\end{aligned}$$

Step 2: Bounding the first term. We begin by bounding the first sum. Suppose $\mathbb{P}(\mathcal{G}_t) \leq \frac{1}{2}\mathbb{P}(\mathcal{R}_t)$. Then, (31) implies that we have that

$$\mathbb{P}(\mathcal{R}_{t+1}) \leq \mathbb{P}(\mathcal{R}_t) - \frac{1}{\upsilon^*_{class}} \mathbb{P}(\mathcal{R}_t \setminus \mathcal{G}_t) = (1 - \frac{1}{\upsilon^*_{class}})\mathbb{P}(\mathcal{R}_t) + \frac{1}{\upsilon^*_{class}}\mathbb{P}(\mathcal{G}_t) \leq (1 - \frac{1}{2\upsilon^*_{class}})\mathbb{P}(\mathcal{R}_t).$$

Thus, if for some $t' \geq t$ $\mathbb{1}\{\mathcal{R}_t \neq S_{f^*} \wedge \mathbb{P}(\mathcal{G}_t) \leq \frac{1}{2}\mathbb{P}(\mathcal{R}_t)\}$ occurs k times in the rounds between t and t', we have that

$$\mathbb{P}(\mathcal{R}_{t'}) \leq (1 - \frac{1}{2\upsilon^*_{class}})^k \mathbb{P}(\mathcal{R}_t).$$

This implies that if this event occurs $c \log(\frac{1}{\mathbb{P}(S_{f^*})})\upsilon^*_{class}$ times, we have that $\mathcal{R}_{t'} = S_{f^*}$. This completes bounding the second term.

Step 3: Bounding the second term. Fix round t and suppose that event $\mathbb{1}\{\mathcal{R}_t \neq S_{f^*} \wedge \mathbb{P}(\mathcal{G}_t) > \frac{1}{2}\mathbb{P}(\mathcal{R}_t)\}$ occurs. Since $\mathbb{P}(\mathcal{G}_t) > \frac{1}{2}\mathbb{P}(\mathcal{R}_t)$, there exists $\bar{f} \in \mathcal{F}_{\mathcal{R}}$ such that $\bar{f} \neq f^*$, $\mathcal{G}_t = S_{\bar{f}}$, and $\bar{r} \in \mathcal{R}$ such that $\mathcal{D}[\bar{r}] = \bar{f}$. Let $t' \geq t$ be the round at which $\bar{f}$ is kicked out or is the last remaining function. We claim that for all $s \in [t, t']$, $\mathcal{G}_s = S_{\bar{f}}$. Fix $i \in [n]$ and note that

$$\begin{aligned}\mathbb{P}_{r \sim \pi_s}(\mathcal{D}[r(x_i)] = \mathcal{D}[\bar{r}(x_i)]) &\geq \mathbb{P}_{r \sim \pi_s}(\mathcal{D}[r] = \mathcal{D}[\bar{r}]) && (32) \\ &= \frac{\mathbb{P}(\mathcal{D}[r] = \mathcal{D}[\bar{r}])}{\mathbb{P}(\mathcal{R}_s)} && (33) \\ &> \frac{1}{2}\frac{\mathbb{P}(\mathcal{R}_t)}{\mathbb{P}(\mathcal{R}_s)} && (34) \\ &\geq \frac{1}{2} && (35)\end{aligned}$$

where (32) follows since if $\mathcal{D}[r] = \mathcal{D}[\bar{r}]$, then $\mathcal{D}[r(x_i)] = \mathcal{D}[\bar{r}(x_i)]$, (33) follows by the definition of conditional probability, (34) follows since $\mathbb{P}(S_{\bar{f}}) = \mathbb{P}(\mathcal{G}_t) > \frac{1}{2}\mathbb{P}(\mathcal{R}_t)$, and (35) follows since $\mathcal{R}_t \supset \mathcal{R}_s$ because $s \geq t$ and the version spaces are monotonically decreasing. Thus, inspection of the definition of $\mathcal{G}_s$ from (28) and the definition of θ as

$$\theta_i = \text{argmin}_{y \in \mathcal{Y}} \mathbb{P}(r \in \mathcal{R}_s : \mathcal{D}[r(x_i)] \neq y) = \arg\max_{y \in \mathcal{Y}} \mathbb{P}(r \in \mathcal{R}_s : \mathcal{D}[r(x_i)] = y)$$

with $|\mathcal{Y}| = 2$ shows the claim that for all $s \in [t, t']$, $\mathcal{G}_s = S_{\bar{f}}$.

Now, consider round $t + 1$. Then, by (31) and $\mathcal{G}_{t+1} = S_{\bar{f}}$, we have that

$$\mathbb{P}_t(\mathcal{R}_{t+1} \setminus S_{\bar{f}}) + \Pr(S_{\bar{f}}) \leq (1 - \frac{1}{\upsilon^*_{class}})\mathbb{P}_t(\mathcal{R}_t \setminus S_{\bar{f}}) + \Pr(S_{\bar{f}}),$$

which implies that

$$\mathbb{P}(\mathcal{R}_{t+1} \setminus S_{\bar{f}}) \leq (1 - \frac{1}{\upsilon^*_{class}})\mathbb{P}(\mathcal{R}_t \setminus S_{\bar{f}}).$$

Unraveling this recurrence up to t', we have that

$$\mathbb{P}_t(\mathcal{R}_{t'}) = \mathbb{P}_t(\mathcal{R}_{t'} \setminus S_{\bar{f}}) \leq (1 - \frac{1}{\upsilon^*_{class}})^{t'-t}\mathbb{P}_t(\mathcal{R}_t \setminus S_{\bar{f}}) \leq (1 - \frac{1}{\upsilon^*_{class}})^{t'-t}\mathbb{P}(\mathcal{R}_t).$$

where in the first equality we used the fact that by definition of t', $\mathcal{R}_{t'} \cap S_{\bar{f}} = \emptyset$. Thus, we see that the second term is bounded above by at most $c\log(\frac{1}{\mathbb{P}(\mathcal{S}_{f^*})})\upsilon^*_{class}$.

Step 4. Bounding the third term. By the argument in the previous step, we have that $\mathcal{G}_t = \mathcal{S}_{f^*}$ for the rest of the game. Thus, we have that from (31)

$$\begin{aligned}\mathbb{P}(\mathcal{R}_{t+1}) &\leq \mathbb{P}(\mathcal{R}_t) - \frac{1}{\upsilon^*_{class}}\mathbb{P}(\mathcal{R}_t \setminus S_{f^*}) \\ &= (1 - \frac{1}{\upsilon^*_{class}})\mathbb{P}(\mathcal{R}_t) + \frac{1}{\upsilon^*_{class}}\mathbb{P}(S_{f^*}).\end{aligned}$$

Thus, unravelling the recurrence, after k rounds, we have that

$$\begin{aligned}\mathbb{P}(\mathcal{R}_{t+k}) &\leq (1 - \frac{1}{\upsilon^*_{class}})^k\mathbb{P}(\mathcal{R}_t) + \frac{1}{\upsilon^*_{class}}\sum_{l=1}^{k-1}(1 - \frac{1}{\upsilon^*_{class}})^l\mathbb{P}(S_{f^*}) \\ &\leq (1 - \frac{1}{\upsilon^*_{class}})^k\mathbb{P}(\mathcal{R}_t) + \mathbb{P}(S_{f^*})\end{aligned}$$

where the last inequality followed by the geometric series. This implies that if this event occurs $c\log(\frac{1}{\mathbb{P}(S_{\min}})\upsilon^*_{class}$ times, we have that $\mathcal{R}_{t'} = S_{f^*}$. This completes bounding the third term.

□

Corollary 2. *If the most probable classifier is output at each round, then after* $c\log(\frac{1}{\mathbb{P}(S_{f^*}})\upsilon^*_{class}(\mathcal{F}_{\mathcal{R}})$ *queries, only* f^* *will be output for the rest of the game.*

Proof. The only step that is slightly different is step 4. Here, it can be seen that after $c\log(\frac{1}{\mathbb{P}(S_{f^*}})\upsilon^*_{class}(\mathcal{F}_{\mathcal{R}})$ queries, f^* is the most probable classifier and it will remain the most probable classifier for the remainder of the game. □

Next, we consider some other settings briefly.

Threshold/Satisficing Bandits. Here we are given a threshold $\gamma \in \mathbb{R}$, and the goal is to identify $i \in [n]$ such that $y_i \leq \gamma$ (this problem was introduced and studied in the standard multi-armed bandit setting in [34]). Then, the extended-teaching dimension notion is

$$\begin{aligned}&\upsilon^*_\gamma = \max\nolimits_{g:\mathcal{X}\mapsto\mathcal{Y}} \min_{I\subset[n]} |I| \\ &\quad \text{s.t. } \exists j \in [n] : \{f \in \mathcal{F} : \mathcal{D}[f(x_i)] = g(x_i)\forall i \in I\} \subset \{f \in \mathcal{F} : \mathcal{D}[f(x_j)] \leq \gamma\}.\end{aligned}$$

The version space is given by

$$\mathcal{F}_t = \{f \in \mathcal{F} : \mathcal{D}[f(x_i)] = y_i \forall i \in O_t\}.$$

The algorithm would query by the following rule

$$I_t \in \underset{i\in[n]\setminus O_t}{\arg\max} \min_{y\in\mathcal{Y}} \mathbb{P}_{f\sim P_k}(f \in \mathcal{F}_{t,\gamma}(x_i) \cup \mathcal{F}_t((x_i, y)))$$

where $\mathcal{F}_{t,\gamma}(x_i) = \{f \in \mathcal{F}_t : \mathcal{D}[f(x_i)] \leq \gamma\}$. We note that for this problem it is not necessary to sample from the mixture P_k because the version space is convex.

We conjecture that there are extensions to settings where the feedback is multi-dimensional (e.g., [35]) or the goal is constrained best arm identification (e.g., [36]).

```
**Input:** distribution $P$ ; event $\Sigma$, accuracy $\epsilon$, failure probability $\delta$, ;
$N \longleftarrow 2\epsilon^{-2}\log(1/\delta)$;
Sample $z_1, \dots, z_N \sim P$ ;
**return** $\frac{\sum_{i=1}^{N} \mathbb{1}\{z_i \in \Sigma\}}{N}$
```

Algorithm 6: EstimateEvent

```
$P_1 \longleftarrow \pi, \mathcal{R}_1 \longleftarrow \mathcal{R}, O_1 \longleftarrow \emptyset$ ;
$k \longleftarrow 1, t_1 = 1, \delta_t = \frac{\delta}{2t^2|\mathcal{Y}|n}$ ;
**for** $t = 1, 2, \dots$ **do**
    Let $I_t \in \arg\max_{i\in[n]\setminus O_t} \min_{y\in\mathcal{Y}}$ EstimateEvent$(P_k, \mathcal{R}_t(x_i) \cup \mathcal{R}_t((x_i, y)), \frac{1}{32n^2}, \delta_t)$;
    Query $x_{I_t}$ and observe $y_{I_t}$ and set $O_{t+1} \longleftarrow O_t \cup \{I_t\}$ ;
    Let $\mathcal{R}_{t+1} \longleftarrow \mathcal{R}_t \setminus (\mathcal{R}_t(x_{I_t}) \cup \mathcal{R}_t((x_{I_t}, y_{I_t})))$;
    **if** *EstimateEvent*$(P_k, \mathcal{R}_{t+1}, \frac{1}{8n}, \delta_t) \leq \frac{1}{4n}$ **then**
        $P_{k+1} \longleftarrow \frac{1}{n-|O_{t_k}|}\sum_{l\in[n]\setminus O_{t_k}} \pi_{C_l(O_{t_k})}$;
        $k \longleftarrow k+1$;
        $t_k \longleftarrow t+1$;
    **if** *STOP(*$\mathcal{F}_{\mathcal{R}}$*,*$O_t$*)* **then**
        **return** argmin$_{i\in O_t} y_i$
```

Algorithm 7: `GRAILS` with Estimation

G Stopping Condition

Here we briefly give a concrete case for the stopping condition for linear models. Let

$$\mathcal{R} = \{\langle a, \cdot\rangle : a \in \mathbb{R}^d, \|a\| \leq 1\}$$

and $\mathcal{Y} = [K]$ for some $K \in \mathbb{N}$. Suppose that at round t $O_t = \{I_1, \dots, I_t\}$ and $y_{I_1}, \dots, y_{I_t}$ have been observed. Suppose $y_{I_1}, \dots, y_{I_t} \neq 1$ since we are otherwise done and let $\bar{y} = \max_{y < \min_{s:\, y_{I_s}}} y$. Fix $l \in [n] \setminus \{I_1, \dots, I_t\}$. Then, we have that

$$C_l(O_t) = \{a \in \mathbb{R}^d : \|a\| \leq 1, |a^\top x_i - y_i| \leq 1/2 \forall i \in O_t, a^\top x_l \leq \bar{y} + 1/2\}.$$

Determining whether $C_l(O_t)$ is nonempty is a simple convex feasibility problem.

H The Algorithms with Approximation

H.1 Best Arm Identification

Algorithm 7 is the version of Algorithm 1 for best arm identification that estimates the events. It uses the subroutine EstimateEvent, Algorithm 6, to estimate the events. Note that the sampling oracle is called a a number of times that is polynomial in n and polynomial in $|\mathcal{Y}|$.

Theorem 9. *With probability at least $1 - \delta$, if $t \geq c v^*_{best} \log(\frac{1}{\mathbb{P}_\pi(S_{f^*})}) \ln(n)$ where $c > 0$ is a universal constant, Algorithm 7 has pulled an arm I_s for $s \leq t$ such that $I_s \in$ argmin$_{i\in[n]} y_i$.*

Proof of Theorem 9. The proof is quite similar to the proof of Theorem 2.

Step 0. Defining the good event. Define

$$\begin{aligned}
\Sigma_{1,t} &= \{\forall y \in \mathcal{Y}, \forall i \in [n] : |\text{EstimateEvent}(P_k, \mathcal{R}_t(x_i) \cup \mathcal{R}_t((x_i, y)), \frac{1}{32n^2}, \delta_t) \\
&\qquad - \mathbb{P}_{f\sim P_k}(r \in \mathcal{R}_t(x_i) \cup \mathcal{R}_t((x_i, y)))| \leq \frac{1}{32n^2}\} \\
\Sigma_{2,t} &= \{|\text{EstimateEvent}(P_k, \mathcal{R}_{t+1}, \frac{1}{8n}, \delta_t) - \mathbb{P}_{f\sim P_k}(r \in \mathcal{R}_{t+1})| \leq \frac{1}{8n}\} \\
\Sigma_t &= \Sigma_{1,t} \cap \Sigma_{2,t} \\
\Sigma &= \cap_{t=1}^{\infty} \Sigma_t.
\end{aligned}$$

By the total law of probability, we have that

$$\begin{aligned}
\mathbb{P}(\Sigma^c) &\leq \sum_{t=1}^{\infty} \mathbb{P}(\Sigma_t^c | \cap_{s=1}^{t-1} \Sigma_s) \\
&\leq \sum_{t=1}^{\infty} \mathbb{P}(\Sigma_{t,1}^c | \cap_{s=1}^{t-1} \Sigma_s) + \mathbb{P}(\Sigma_{t,2}^c | \cap_{s=1}^{t-1} \Sigma_s) \\
&\leq \frac{6}{\pi^2} \sum_{t=1}^{\infty} \frac{\delta}{t^2} \\
&\leq \delta
\end{aligned}$$

where the last line follows by the definition of the Algorithm EstimateEvent and standard Hoeffding bounds. For the rest of the proof, we will condition on the event Σ.

Step 1. This step is very similar to step 1 in the proof of Theorem 2. Consider phase k, let $t \geq t_k$. Define $\theta \in \mathcal{Y}^n$ by

$$\theta_i = \begin{cases} y_i & i \in O_t \\ \text{argmin}_{y \in \mathcal{Y}} \mathbb{P}_k(r \in \mathcal{R}_t(x_i) \cup \mathcal{R}_t((x_i, y))) & \text{o/w} \end{cases}.$$

By definition of υ^*_{best}, there exists $\tilde{I} = \{i_1, \ldots, i_{\upsilon^*_{best}}\}$ such that there exists $j_0 \in [n] \setminus O_t$ such that

$$\{r \in \mathcal{R}_t : \mathcal{D}[r(x_i)] = \theta_i \; \forall i \in \tilde{I}\} \subset \{r \in \mathcal{R}_t : j_0 \in \text{argmin}_{l \in [n]} \mathcal{D}[r(x_l)]\}$$

Define $I = \{i_1, \ldots, i_{\upsilon^*_{best}}, j_0\}$. Then, we have that

$$\mathcal{R}_t = \{r \in \mathcal{R}_t : \exists i \in I \text{ s.t. } i \in \text{argmin}_{l \in [n]} \mathcal{D}[r(x_l)] \text{ or } \mathcal{D}[r(x_i)] \neq \theta_i\}.$$

Define

$$\begin{aligned}
J_t &:= \arg\max_{i \in I} \mathbb{P}_k(\{r \in \mathcal{R}_t : i \in \text{argmin}_{l \in [n]} \mathcal{D}[r(x_l)] \text{ or } \mathcal{D}[r(x_i)] \neq \theta_i\}) \\
\bar{y} &:= \text{argmin}_{y \in \mathcal{Y}} \text{EstimateEvent}(P_k, \mathcal{R}_t(x_{I_t}) \cup \mathcal{R}_t((x_{I_t}, \bar{y}), \frac{1}{32n^2}, \delta_t)
\end{aligned}$$

Note J_t would be chosen if there were no noise from estimation.

By the event $\Sigma_{1,t}$,

$$\begin{aligned}
&\mathbb{P}_k(\{r \in \mathcal{R}_t : J_t \in \text{argmin}_{l \in [n]} \mathcal{D}[r(x_l)] \text{ or } \mathcal{D}[r(x_{J_t})] \neq \theta_{J_t}\}) \\
&\qquad \leq \text{EstimateEvent}(P_k, \mathcal{R}_t(x_{J_t}) \cup \mathcal{R}_t((x_{J_t}, \theta_{J_t})), \frac{1}{32n^2}, \delta_t) + \frac{1}{32n^2} && (36) \\
&\qquad \leq \text{EstimateEvent}(P_k, \mathcal{R}_t(x_{I_t}) \cup \mathcal{R}_t((x_{I_t}, \bar{y})), \frac{1}{32n^2}, \delta_t) + \frac{1}{32n^2} && (37) \\
&\qquad \leq \mathbb{P}_k(\{r \in \mathcal{R}_t : I_t \in \text{argmin}_{l \in [n]} \mathcal{D}[r(x_l)] \text{ or } \mathcal{D}[r(x_{I_t})] \neq \bar{y}\}) + \frac{1}{16n^2} && (38) \\
&\qquad \leq \mathbb{P}_k(\{r \in \mathcal{R}_t : I_t \in \text{argmin}_{l \in [n]} \mathcal{D}[r(x_l)] \text{ or } \mathcal{D}[r(x_{I_t})] \neq \theta_{I_t}\}) + \frac{1}{16n^2} && (39)
\end{aligned}$$

where lines (36) and (38) follow from event $\Sigma_{1,t}$, line (37) follows from the definition of I_t and $\bar{y}$, and line (39) used the definition of θ_{I_t}.

From this, it follows that

$$\begin{aligned}\mathbb{P}_k(\mathcal{R}_t) &= \mathbb{P}_k(\{r \in \mathcal{R}_t : \exists i \in I \text{ s.t. } i \in \text{argmin}_{l\in[n]}\mathcal{D}[r(x_l)] \text{ or } \mathcal{D}[r(x_i)] \neq \theta_i\}) \\ &\leq \sum_{i\in I} \mathbb{P}_k(\{r \in \mathcal{R}_t : i \in \text{argmin}_{l\in[n]}\mathcal{D}[r(x_l)] \text{ or } \mathcal{D}[r(x_i)] \neq \theta_i\}) \\ &\leq (\upsilon^*_{best}+1)\text{max}_{i\in I}\mathbb{P}_k(\{r \in \mathcal{R}_t : i \in \text{argmin}_{l\in[n]}\mathcal{D}[r(x_l)] \text{ or } \mathcal{D}[r(x_i)] \neq \theta_i\}) \\ &= (\upsilon^*_{best}+1)\mathbb{P}_k(\{r \in \mathcal{R}_t : J_t \in \text{argmin}_{l\in[n]}\mathcal{D}[r(x_l)] \text{ or } \mathcal{D}[r(x_i)] \neq \theta_{I_t}\}) \\ &= (\upsilon^*_{best}+1)[\mathbb{P}_k(\{r \in \mathcal{R}_t : I_t \in \text{argmin}_{l\in[n]}\mathcal{D}[r(x_l)] \text{ or } \mathcal{D}[r(x_i)] \neq \theta_{I_t}\}) + \frac{1}{16n^2}] \quad (40) \\ &\leq (\upsilon^*_{best}+1)[\mathbb{P}_k(\{r \in \mathcal{R}_t : I_t \in \text{argmin}_{l\in[n]}\mathcal{D}[r(x_l)] \text{ or } \mathcal{D}[r(x_i)] \neq y_{I_t}\}) + \frac{1}{16n^2}] \quad (41) \\ &= (\upsilon^*_{best}+1)[\mathbb{P}_k(\mathcal{R}_t(x_{I_t}) \cup \mathcal{R}_t((x_{I_t}, y_{I_t})) + \frac{1}{16n^2}]\end{aligned}$$

where line (40) uses (39), and the line (41) comes from the definition of θ. Noticing that

$$\begin{aligned}\mathcal{R}_t &= (\mathcal{R}_t(x_{I_t}) \cup \mathcal{R}_t((x_{I_t}, y_{I_t})) \cup \mathcal{R}_t \setminus (\mathcal{R}_t(x_{I_t}) \cup \mathcal{R}_t((x_{I_t}, y_{I_t})) \\ &= (\mathcal{R}_t(x_{I_t}) \cup \mathcal{R}_t((x_{I_t}, y_{I_t})) \cup \mathcal{R}_{t+1},\end{aligned}$$

we have that

$$\mathbb{P}_k(\mathcal{R}_{t+1}) = \mathbb{P}_k(\mathcal{R}_t) - \mathbb{P}_k(\mathcal{R}_t(x_{I_t}) \cup \mathcal{R}_t((x_{I_t}, y_{I_t})) \leq (1 - \frac{1}{\upsilon^*_{best}+1})\mathbb{P}_k(\mathcal{R}_t) + \frac{1}{16n^2}.$$

Unraveling this recursive statement, we have that

$$\begin{aligned}\mathbb{P}_k(\mathcal{R}_{t+1}) &\leq (1 - \frac{1}{\upsilon^*_{best}+1})^{t-t_k} + (t-t_k)\frac{1}{16n^2} \\ &\leq (1 - \frac{1}{\upsilon^*_{best}+1})^{t-t_k} + n\frac{1}{16n^2} \\ &\leq (1 - \frac{1}{\upsilon^*_{best}+1})^{t-t_k} + \frac{1}{16n}.\end{aligned}$$

where we used the fact that the game lasts at most n rounds because it is the noiseless setting. Thus, we see that in phase k at $t = t_k + s_k$ with $s_k := O(\ln(n)\upsilon^*_{best})$, we have that

$$\mathbb{P}_k(\mathcal{R}_{t_k+s_k}) \leq \frac{1}{8n}. \quad (42)$$

Step 2.1: Necessary condition for entering the next phase. Now, we show that if Algorithm 7 enters the phase $k+1$, then $\mathbb{P}_{f\sim P_k}(r \in \mathcal{R}_{t+1}) < \frac{1}{2n}$. note that if $\frac{1}{4n} \geq$ EstimateEvent$(P_k, \mathcal{R}_{t+1}, \frac{1}{8n}, \delta_t)$, then

$$\frac{1}{4n} \geq \text{EstimateEvent}(P_k, \mathcal{R}_{t+1}, \frac{1}{8n}, \delta_t) \geq \mathbb{P}_{f\sim P_k}(r \in \mathcal{R}_{t+1}) - \frac{1}{8n}$$

implying that

$$\mathbb{P}_{f\sim P_k}(r \in \mathcal{R}_{t+1}) < \frac{1}{2n}.$$

Step 2.2: Necessary condition for not entering the next phase. Now, we show that if $\mathbb{P}_{f\sim P_k}(r \in \mathcal{R}_{t+1}) \geq \frac{1}{2n}$, then Algorithm 7 does not the phase $k+1$. Suppose $\mathbb{P}_{f\sim P_k}(r \in \mathcal{R}_{t+1}) \geq \frac{1}{2n}$. Then,

$$\frac{1}{2n} \leq \mathbb{P}_{f\sim P_k}(r \in \mathcal{R}_{t+1}) \leq \text{EstimateEvent}(P_k, \mathcal{R}_{t+1}, \frac{1}{8n}, \delta_t) + \frac{1}{8n},$$

implying that EstimateEvent$(P_k, \mathcal{R}_{t+1}, \frac{1}{8n}, \delta_t) > \frac{1}{4n}$ and the algorithm does not update the phase.

Given these conditions, the rest of the proof is identical to the proof of Theorem 2.

□

```
**Input:** distribution $P$ ; event $\Sigma$, multiplicative accuracy $\epsilon \in (0,1)$, failure probability $\delta \in (0,1)$;
$s \longleftarrow 0$;
while $\sqrt{\frac{2\log(\frac{s^2\pi^2}{\delta 6})}{s}} \geq \frac{1}{2n^2 max_{y\in\mathcal{Y}} y}$ do
    Sample $z_s \sim P$;
    Form $\widehat{\mu}_s = \frac{\sum_{i=1}^{s} \mathbb{1}\{z_i \in \Sigma\}}{s}$;
    if $\epsilon \cdot \widehat{\mu}_s \geq \sqrt{2\log(\frac{s^2\pi^2}{\delta 6})/s}$ then
        return $\widehat{\mu}_s$
    $s \longleftarrow s+1$;
return 0
```

Algorithm 8: EstimateEventMult

```
$P_1 \longleftarrow \pi, \mathcal{R}_1 \longleftarrow \mathcal{R}, O_1 \longleftarrow \emptyset$ ;
$k \longleftarrow 1, t_1 = 1, \delta_t = \frac{\delta}{2t^2|\mathcal{Y}|n}$;
for $t = 1, 2, \ldots$ do
    Let $I_t \in \text{argmin}_{i\in[n]\setminus O_t} \text{max}_{y\in\mathcal{Y}} \frac{y}{\text{EstimateEventMult}(P_k, \mathcal{R}_t(x_i) \cup \mathcal{R}_t((x_i, y)), \frac{1}{2}, \delta_t)}$;
    Query $x_{I_t}$ and observe $y_{I_t}$ and set $O_{t+1} \longleftarrow O_t \cup \{I_t\}$ ;
    Let $\mathcal{R}_{t+1} \longleftarrow \mathcal{R}_t \setminus (\mathcal{R}_t(x_{I_t}) \cup \mathcal{R}_t((x_{I_t}, y_{I_t})))$;
    if *EstimateEvent*$(P_k, \mathcal{R}_{t+1}, \frac{1}{8n}, \delta_t) \leq \frac{1}{4n}$ then
        $P_{k+1} \longleftarrow \frac{1}{n-|O_{t_k}|} \sum_{l\in[n]\setminus O_{t_k}} \pi_{C_l(O_{t_k})})$;
        $k \longleftarrow k+1$;
        $t_k \longleftarrow t+1$;
    if *STOP*$(\mathcal{F}_{\mathcal{R}}, O_t)$ then
        return $\text{argmin}_{i\in O_t} y_i$
```

Algorithm 9: `GRAILS` for Loss Minimization with Estimation

H.2 Cumulative Loss Minimization

For simplicity, in this section, we assume that $\min_{y\in\mathcal{Y}} y \geq 1$. One can simply add a constant to $\mathcal{R}$ to arrive at this setting. We use the convention that $\frac{1}{0} = \infty$.

Lemma 1. *Let Σ denote some event that has positive probability under P such that $\epsilon(1-\epsilon)\mathbb{P}(\Sigma) \geq \frac{1}{2n^2 max_{y\in\mathcal{Y}} y}$. With probability at least $1-\delta$, Algorithm 8 satisfies*

$$(1-\epsilon)\textit{EstimateEventMult}(P, \Sigma, \epsilon, \delta) \leq P(\Sigma) \leq (1+\epsilon)\textit{EstimateEventMult}(P, \Sigma, \epsilon, \delta)$$

Proof of Lemma 1. Define the event

$$E = \{|P(\Sigma) - \widehat{\mu}_s| \leq \sqrt{\frac{2\log(\frac{s^2\pi^2}{\delta 6})}{s}}\ \forall s \in \mathbb{N}\}$$

By a union bound and a standard Hoeffding bound, $\mathbb{P}(E) \geq 1-\delta$. Suppose E holds for the remainder of the proof.

Step 1. First, we show that if $\epsilon(1-\epsilon)\mathbb{P}(\Sigma) \geq \frac{1}{n^3 \text{max}_{y\in\mathcal{Y}}}$, then Algorithm 8 does not return 0. Towards a contradiction, suppose the algorithm returns 0 at round s_0. Then, we have on the event E that

$$|P(\Sigma) - \widehat{\mu}_{s_0}| \leq \sqrt{\frac{2\log(\frac{s_0^2\pi^2}{\delta 6})}{s_0}} \leq \frac{1}{2n^2 \text{max}_{y\in\mathcal{Y}} y} \leq \epsilon(1-\epsilon)\mathbb{P}(\Sigma).$$

Rearranging we have that

$$P(\Sigma) \leq \frac{1}{1-\epsilon}\widehat{\mu}_{s_0},$$

which implies

$$\sqrt{\frac{2\log(\frac{s^2\pi^2}{\delta 6})}{s}} \leq \frac{1}{2n^2 \text{max}_{y\in\mathcal{Y}}} \leq \epsilon(1-\epsilon)\mathbb{P}(\Sigma) \leq \epsilon\widehat{\mu}_{s_0}.$$

But this implies the algorithm would not have returned 0, yielding a contradiction.

Step 2. Suppose that Algorithm 8 does not return 0. Then, there exists a round $s_0 \in \mathbb{N}$ at which

$$\sqrt{\log(\frac{s_0^2\pi^2}{\delta 6})/s_0} \leq \epsilon\widehat{\mu}_{s_0}.$$

Then, by the event E, we have that

$$P(\Sigma) \leq (1+\epsilon)\widehat{\mu}_{s_0}.$$

The other inequality follows as well, completing the proof. □

Theorem 10. *With probability at least $1-\delta$, Algorithm 9 incurs a loss of at most $2(v^*_{\mathit{loss}} + \mathit{max}_{i\in[n],f\in\mathcal{F}_\mathcal{R}} f(x_i))\log(n)\log(\frac{1}{\mathbb{P}(S_{\mathit{min}})})$ to identify $\mathit{argmin}_{i\in[n]} f^*(x_i)$.*

Proof. The proof is very similar to the proof of Theorem 4.

Step 0. Defining the good event. Define

$$\begin{aligned}
\Sigma_{1,t} &= \{\forall y \in \mathcal{Y}, \forall i \in [n] : \mathbb{P}_{f\sim P_k}(f \in \mathcal{R}_t(x_i) \cup \mathcal{R}_t((x_i,y)) \geq \frac{2}{n^2 \text{max}_{y\in\mathcal{Y}} y} \implies \\
&\qquad \frac{1}{2}\text{EstimateEventMult}(P_k, \mathcal{R}_t(x_i) \cup \mathcal{R}_t((x_i,y), \frac{1}{2}, \delta_t) \\
&\qquad \leq \mathbb{P}_{f\sim P_k}(f \in \mathcal{R}_t(x_i) \cup \mathcal{R}_t((x_i,y))) \\
&\qquad \leq \frac{3}{2}\text{EstimateEventMult}(P_k, \mathcal{R}_t(x_i) \cup \mathcal{R}_t((x_i,y)), \frac{1}{2}, \delta_t)\} \\
\Sigma_{2,t} &= \{|\text{EstimateEvent}(P_k, \mathcal{R}_{t+1}, \frac{1}{8n}, \delta_t) - \mathbb{P}_{f\sim P_k}(f \in \mathcal{R}_{t+1})| \leq \frac{1}{8n}\} \\
\Sigma_t &= \Sigma_{1,t} \cap \Sigma_{2,t} \\
\Sigma &= \cap_{t=1}^{\infty} \Sigma_t.
\end{aligned}$$

By the total law of probability, a Standard Hoeffding bound, and Lemma 1, we have that

$$\begin{aligned}
\mathbb{P}(\Sigma^c) &\leq \sum_{t=1}^{\infty} \mathbb{P}(\Sigma_t^c | \cap_{s=1}^{t-1} \Sigma_s) \\
&\leq \sum_{t=1}^{\infty} \mathbb{P}(\Sigma_{t,1}^c | \cap_{s=1}^{t-1} \Sigma_s) + \mathbb{P}(\Sigma_{t,2}^c | \cap_{s=1}^{t-1} \Sigma_s) \\
&\leq \frac{6}{\pi^2}\sum_{t=1}^{\infty}\frac{\delta}{t^2} \\
&\leq \delta.
\end{aligned}$$

For the rest of the proof, we will condition on the event Σ.

Step 1: Define the function $g : \{x_1, \ldots, x_n\} \mapsto \mathcal{Y}$ in the following way:

$$g(x_i) = \begin{cases} \mathcal{D}[\tilde{r}(x_i)] \text{ s.t. } \tilde{r} = \arg\max_{\bar{r}\in\mathcal{R}_t} \frac{\mathcal{D}[\bar{r}(x_i)]}{\mathbb{P}_k(\mathcal{R}_t(x_i)\cup\mathcal{R}((x_i,\bar{f}(x_i)))} & i \notin \{I_1, \ldots, I_{t-1}\} \\ f^*(x_i) & \text{o/w} \end{cases}$$

Define

$$J_t := \text{argmin}_{j\in[n]\setminus O_t} \frac{g(x_j)}{\mathbb{P}_k(\mathcal{R}_t(x_j) \cup \mathcal{R}_t((x_i, g(x_j)))},$$

the arm that would be pulled at round t if the probabilities were known exactly. Note that

We have that

$$
\begin{aligned}
&\frac{g(x_{J_t})}{\mathbb{P}_k(\mathcal{R}_t(x_{J_t}) \cup \mathcal{R}_t((x_i, g(x_{J_t})))} \\
&\leq \frac{\sum_{i \in [n] \setminus O_t} g(x_i)}{\sum_{i \in [n] \setminus O_t} \mathbb{P}_k(\{r \in \mathcal{R}_t : \mathcal{D}[r(x_i)] \neq g(x_i) \text{ or } i \in \text{argmin}_{j \in [n]} \mathcal{D}[r(x_j)]\})} \\
&\leq \frac{\sum_{i \in [n] \setminus O_t} g(x_i)}{\mathbb{P}_k(\mathcal{R}_t)} \\
&\leq \frac{n \text{max}_{y \in \mathcal{Y}} y}{\mathbb{P}_k(\mathcal{R}_t)} \\
&\leq n^2 \text{max}_{y \in \mathcal{Y}} y,
\end{aligned}
$$

where the last line uses Step 2.2 in the proof of Theorem 9, which shows that $\mathbb{P}_k(\mathcal{R}_t) \geq \frac{1}{2n}$. This implies that

$$
\mathbb{P}_k(\mathcal{R}_t(x_{J_t}) \cup \mathcal{R}((x_i, g(x_{J_t}))) \geq \frac{g(x_{J_t})}{cn^3 \text{max}_{y \in \mathcal{Y}} y} \geq \frac{1}{cn^3 \text{max}_{y \in \mathcal{Y}} y}
$$

where we used the fact that $\min_{y \in \mathcal{Y}} y \geq 1$ by assumption.

Thus, by event $\Sigma_{1,t}$, we have that

$$
\begin{aligned}
&\frac{1}{2} \text{EstimateEventMult}(P_k, \mathcal{R}_t(x_{J_t}) \cup \mathcal{R}_t((x_{J_t}, g(x_{J_t})), \frac{1}{2}, \delta_t) \\
\leq &\mathbb{P}_{f \sim P_k}(f \in \mathcal{R}_t(x_{J_t}) \cup \mathcal{R}_t((x_{J_t}, g(x_{J_t})))) \\
\leq &\frac{3}{2} \text{EstimateEventMult}(P_k, \mathcal{R}_t(x_{J_t}) \cup \mathcal{R}_t((x_{J_t}, g(x_{J_t})), \frac{1}{2}, \delta_t)
\end{aligned}
$$

Then, by standard arguments, it can be shown that Algorithm 9 chooses an arm I_t that satisfies

$$
\frac{g(x_{I_t})}{\mathbb{P}_{f \sim P_k}(f \in \mathcal{R}_t(x_{I_t}) \cup \mathcal{R}_t((x_{I_t}, g(x_{I_t})))} \leq c \frac{g(x_{J_t})}{\mathbb{P}_{f \sim P_k}(f \in \mathcal{R}_t(x_{J_t}) \cup \mathcal{R}_t((x_{J_t}, g(x_{J_t})))}
$$

where c is a universal constant.

Step 2: The only difference in the proof with Theorem 4 comes in lines (18) and (19). In these lines, we lose a mulitiplicative constant factor from using EstimateEventMult$(P_k, \mathcal{R}_t(x_i) \cup \mathcal{R}_t((x_i, y), \frac{1}{2}, \delta_t)$ instead of the exact probability, as shown in the previous step. The argument for changing phases is identical to the argument in the proof of Theorem 9. The rest of the proof for bounding the cumulative loss incurred is identical.

Step 3: The total number of samples taken is a polynomial function of n, max$_{y \in \mathcal{Y}} y$, and $\log(1/\delta)$ by definition of Algorithm 8.

□

I Discussion of Function Classes

Proof of Proposition 5. $\mathcal{F}_{\mathcal{R}}$ consists of functions $f_1, \ldots, f_n$ of the following form

$$
f_j(x_i) \begin{cases} 10 & i = j+1 \\ 1 & i \geq j \\ 0 & i < j \end{cases}.
$$

Suppose that $g : \mathcal{X} \mapsto \mathcal{Y}$ is such that $g \notin \mathcal{F}_{\mathcal{R}}$. Then, a simple computation shows that at most 3 measurements are required to eliminate all functions from the version space. On the other hand, if $g = f_j$ for some $f_j \in\in \mathcal{F}_{\mathcal{R}}$, then it suffices to query x_j and x_{j-1}. Therefore, a simple computation shows that $v^*_{best}(\mathcal{F}_{\mathcal{R}}) \leq 3$.

Since π is uniform over $[0, 1]$, we have that $\mathbb{P}_\pi(S_{f^*}) \geq \frac{c}{n}$ for some positive universal constant. Thus, the result follows.

□

Linear Separators ϵ-good arm identification: We give an intuitive sufficient condition for lower bounding $\mathbb{P}_\pi(S_f)$.

Assumption 1. *Fix $y_1, \ldots, y_n \in \mathcal{Y}$. Define the* minimum margin*:*

$$\gamma^* := max_{r^* \in \mathcal{R}: \mathcal{D}[r^*(x_i)] = y_i \forall i \in [n]} \min_{i \in [n]} \min_{y \neq y_i} |r^*(x_i) - y| - |r^*(x_i) - y_i|.$$

The following Proposition lower bounds $\mathbb{P}_\pi(S_{f^*})$ for linear functions.

Proposition 13. *Let $\mathcal{R} = \{\langle a, \cdot \rangle : a \in \mathbb{R}^d, \|a\| \leq R_1\}$. Let $x_1, \ldots, x_n \in \mathbb{R}^d$ such that $max_{i \in [n]} \|x_i\|_* \leq R_2$ where $\|\cdot\|_*$ denotes the dual norm of $\|\cdot\|$. Let $y_1, \ldots, y_n \in \mathcal{Y}$. Suppose that $\gamma^* > 0$. Let $f^* = \mathcal{D}[\langle a_*, \cdot \rangle]$ for some a_* with $\|a_*\| \leq R_1$. Let π be the uniform distribution over $\{a : \|a\| \leq R_1\}$. Then,*

$$\mathbb{P}_\pi(S_{f^*}) \geq \frac{volume(\{z \in \mathbb{R}^d : \|z\| \leq \frac{\gamma^*}{R_2}\}}{volume(\{z \in \mathbb{R}^d : \|z\| \leq R_1\})}.$$

In particular, if $\|\cdot\| = \|\cdot\|_2$, then $\mathbb{P}_\pi(S_{f^*}) \geq (\frac{\gamma^*}{R_1 R_2})^d$.

The above Proposition shows that for linear separators with a Euclidean constraint and a margin γ^*, $\log(\frac{1}{\mathbb{P}_\pi(S_{f^*})}) \lesssim d \log(\gamma^*)$.

Proof of Proposition 13. By assumption $\gamma^* > 0$. Define $\delta_0 = \frac{\gamma^*}{6R_2}$. Let $f^*(\cdot) = \langle a_*, \cdot \rangle$ where $a_* \in \mathbb{R}^d$ and $\|a_*\| \leq R_1$. Define

$$\bar{B}_\alpha = B_{\delta_0}(\alpha a_*)$$

where $B_\gamma(v) := \{v \in \mathbb{R}^d : \|v\| \leq \gamma\}$.

Step 1. $\bar{B}_\alpha$ is feasible. We claim that for all $\alpha \in [0, 1 - \delta_0/R_1]$, $\bar{B}_\alpha \subset \{a : \|a\| \leq R_1\}$. Let $\alpha \in [0, 1 - \delta_0/R_1]$. Let v such that $\|v\| \leq \delta_0$. Then,

$$\|\alpha a_* + v\| \leq \alpha \|a_*\| + \|v\| \leq (1 - \delta_0/R_1) \|a_*\| + \|v\| \leq R_1 - \delta_0 + \delta_0 = R_1,$$

showing the claim.

Step 2. $a_* \in \bar{B}_{\alpha_0}$ **for some** α_0 Let $\alpha_0 = \max(1 - \delta_0/\|a_*\|, 0) \leq 1 - \delta_0/R_1$. We claim that $a_* \in \bar{B}_{\alpha_0}$. If $1 - \delta_0/\|a_*\| \geq 0$, then

$$\|(\alpha_0 - 1)a_*\| = \frac{\delta_0}{\|a_*\|} \|a_*\| = \delta_0$$

so that $a_* \in \bar{B}_{\alpha_0}$. On the other hand, if $1 - \delta_0/\|a_*\| < 0$, then $\|a_*\| < \delta_0$, so that $a_0 \in \bar{B}_{\alpha_0}$ since $\alpha_0 = 0$.

Step 3. Putting it together. Now, let $a \in \bar{B}_{\alpha_0}$. Then,

$$\|a_* - a\| \leq 2\delta_0.$$

Then, using Holder's inequality,

$$|a_*^\top x_i - (a^\top x_i)| \leq \|a_* - a\| \max_i \|x_i\|_* \leq 2\delta_0 R_2 \leq \frac{\gamma^*}{3}$$

by definition of δ_0.

Let $y \neq y_i$. Then,

$$\begin{aligned} |\langle a, x_i \rangle - y| - |\langle a, x_i \rangle - y_i| &> |\langle a_*, x_i \rangle - y_i| - |\langle a_*, x_i \rangle - y_i| - \frac{2}{3}\gamma^* \\ &> 0 \end{aligned}$$

where the last line uses the definition of γ^*. Thus, we have that if $f = \langle a, \cdot, \rangle \in S_{f^*}$. Thus,

$$\mathbb{P}_\pi(S_{a_0}) \geq \frac{\text{volume}(\{z \in \mathbb{R}^d : \|z\| \leq \delta_0\}}{\text{volume}(\{z \in \mathbb{R}^d : \|z\| \leq R_1\})}.$$

The result for the case $\|\cdot\| = \|\cdot\|_2$ follows from standard formula for the volume of an n-dimensional ball.

□

Next, we give a simple bound on $\upsilon^*_{best,\epsilon}$ for linear functions. For simplicity, we let $|\mathcal{Y}|$ be countably infinite, but it could be extended to the $|\mathcal{Y}| < \infty$ case.

Proposition 14. *Suppose $\mathcal{Y} = \{\ldots, -2\Delta, -\Delta, 0, \Delta, 2\Delta, \ldots\}$ where $\Delta > 0$. Let $\mathcal{R} = \{r : r(\cdot) = \langle a, \cdot \rangle, a \in \mathbb{R}^d\}$. Let $x_1, \ldots, x_n \in \mathbb{R}^d$ such that $\max_{i \in [n]} \|x_i\|_i \leq 1$. Define*

$$\phi := \min_{i_1, \ldots, i_d \in [n]} \sigma_d \left(\begin{pmatrix} x_{i_1}^\top \\ \vdots \\ x_{i_d} \end{pmatrix} \right)$$

where $\sigma_d(A)$ denotes the dth singular value of a matrix A. Then if $\Delta \leq \min(\frac{\phi\epsilon}{4\sqrt{d}}, \frac{\epsilon}{2})$, $\upsilon^*_{best,\epsilon} \leq d$.

This result, combined with Proposition 13, suggests that `GRAILS` passes the sanity check of not sampling too many linearly dependent arms for the linear functions case.

Proof of Proposition 14. **Step 1:** Let $g : \mathbb{R}^d \mapsto \mathcal{Y}$ be an arbitrary mappying. By assumption, there exists $i_1, \ldots, i_d \in [n]$ such that defining $\bar{X} := \begin{pmatrix} x_{i_1}^\top \\ \vdots \\ x_{i_d}^\top \end{pmatrix}$, we have that $\sigma_d(\bar{X}) = \phi$. Suppose that the algorithm queries $i_1, \ldots, i_d$. If there is no $a_* \in \mathbb{R}^d$ such that $g(x_{i_j}) = \mathcal{D}[a_*^\top x_{i_j}]$ for all $j \in [d]$, then we are done. So, assume that there exists $a_* \in \mathbb{R}^d$ such that $g(x_{i_j}) = \mathcal{D}[a_*^\top x_{i_j}]$ for all $j \in [d]$. Suppose wlog that $x_1 \in \text{argmin}_{i \in [n]} a_*^\top x_i$. Now, we will show that all models consistent with $g(x_{i_j})$ for all $j \in [d]$ put x_1 as an ϵ-good arm. Let $a \in \mathbb{R}^d$ be another vector such that

$$\mathcal{D}[a^\top x_{i_j}] = g(x_{i_j})$$

for all $j \in [d]$. Then, we have that

$$\Delta\sqrt{d} \geq \left\|\bar{X}(a_* - a)\right\| \geq \sigma_d(\bar{X}) \left\|a - a_*\right\|.$$

Let $x_i \neq x_1$. Note that by definition of the rounding operator and $\mathcal{Y} = \{\ldots, -2\Delta, -\Delta, 0, \Delta, 2\Delta, \ldots\}$,

$$a_*^\top x - \frac{\Delta}{2} \leq \mathcal{D}[a_*^\top x] \leq a_*^\top x + \frac{\Delta}{2}$$

for all x.

We have that

$$\begin{aligned} \mathcal{D}[a^\top x_1] - \mathcal{D}[a^\top x_i] - \Delta \leq a^\top(x_i - x_1) &\leq (a - a_*)^\top(x_1 - x_i) + a_*^\top(x_1 - x_i) \\ &\leq \|a - a_*\| \|x_i - x_1\| + a_*^\top(x_1 - x_i) \\ &\leq \frac{\Delta\sqrt{d}}{\phi} 2 + a_*^\top(x_1 - x_i) \\ &\leq \frac{\epsilon}{2} \end{aligned}$$

where we used $a_*^\top x_1 - a_*^\top x_i \leq 0$ by definition of x_1 and the setting of Δ. Thus, rearranging, we have that

$$\mathcal{D}[a^\top x_1] - \mathcal{D}[a^\top x_i] \leq \Delta + \frac{\epsilon}{2} \leq \epsilon$$

implying that x_1 is an ϵ-good arm for any a such that

$$\mathcal{D}[a^\top x_{i_j}] = g(x_{i_j}).$$

This completes the proof.

□

Active Classification with Linear Separators: Next, we lower bound the minimum probability for linear separators under a certain geometric that can be interpreted as a quantitative version of the concept of general position. Define

$$\Gamma(\mathcal{X}) = \{(y_1, \ldots, y_n)^\top \in \{-1,1\}^n : \exists w \in \mathbb{R}^d,\, b \in \mathbb{R} \text{ s.t. } \operatorname{sign}(w^\top x_i + b) = y_i \,\forall i \in [n]\}.$$

Proposition 15. *Let $\gamma > 0$. Suppose $\mathcal{X} = \{x_1, \ldots, x_n\}$ satisfy $\forall S \subset \mathcal{X}$ such that $|S| \leq d-1$,*

$$\min_{x \in \mathcal{X} \setminus S} \operatorname{dist}(x, \textit{aff}(S)) \geq \gamma.$$

Suppose $\max_{i\in[n]} \|x_i\| \leq R$. Let $\mathcal{F}_R = \{\langle w, \cdot \rangle + b : (w^\top, b)^\top \in W\}$ and $W = \{(w^\top, b)^\top : \|w\|_2^2 + b^2 \leq R^2\}$. where R is large enough such that for every $(y_1, \ldots, y_n)^\top \in \Gamma(\mathcal{X})$, the max-margin separator achieving $(y_1, \ldots, y_n)^\top$ belongs to $\mathcal{F}_R$. Let π denote the uniform probability distribution over W. Then,

$$\min_{(y_1,\ldots,y_n)^\top \in \Gamma(\mathcal{X})} \mathbb{P}_{(w^\top,b)^\top \sim \pi}(\operatorname{sign}(w^\top x_i + b) = y_i \,\forall i \in [n]) \geq (\frac{\gamma}{R})^d$$

Thus, under the condition in the Proposition 15, the sample complexity in Theorem 8 scales as $cd\log(\gamma)v^*_{class}(\mathcal{F}_\mathcal{R})$. Often $\log(|\mathcal{F}_\mathcal{R}|) = \Omega(d)$, in which case Theorem 8 is larger than the active classification upper bound in [21] by a factor of $\log(\gamma)$. On the other hand, Algorithm 5 is computationally efficient.

Proof of Proposition 15. Fix $(\bar{y}_1, \ldots, \bar{y}_n)^\top \in \Gamma(\mathcal{X})$. Let $f = \langle \bar{w}, \cdot \rangle + \bar{b} \in \mathcal{F}_B$ attain the max-margin separator achieving $(\bar{y}_1, \ldots, \bar{y}_n)^\top$, which exists by assumption. Let $v_1, \ldots, v_l$ denote the support vectors with positive labels and $u_1, \ldots, u_k$ the support vectors with negative labels. By definition of support vectors, we have that

$$\operatorname{aff}(v_1, \ldots, v_l) = \operatorname{aff}(u_1, \ldots, u_k) + z$$

for some $z \in \mathbb{R}^d$. We claim that $\|z\|_2 \geq \gamma$. Towards a contradiction, suppose not. Then,

$$\operatorname{dist}(v_1, \operatorname{aff}(u_1, \ldots, u_k)) = \|z\|_2 < \gamma$$

which contradicts our assumption. This implies that the margin is at least $\gamma/2$ and that

$$|\bar{w}^\top x - \bar{b}| \geq \frac{\gamma}{2}$$

for all $x \in \mathcal{X}$. Then, for any w, b such that $\|w\|_2^2 + b^2 \leq [\gamma/4]^2$, we have that

$$|\bar{w}^\top x - \bar{b} - [(\bar{w} + w)^\top x - \bar{b} - b]| = |w^\top x - b| \leq 2\left\|(w^\top, b)^\top\right\|_2 \leq \frac{\gamma}{2}$$

The result follows by Proposition 13. □

Strongly Convex Functions: Finally, we consider strongly convex functions. The following Proposition considers a setting of strongly convex functions where $v^*_{best} \leq O(1)$.

Proposition 16. *Let $K \in \mathbb{N}$ and let $\mathcal{X} = [K]$. Let $\mathcal{R} = \{r : r \text{ is } \alpha\text{-strongly convex}\}$. Let $\Delta = \min_{y \neq y' \in \mathcal{Y}} |y - y'|$. If $\alpha \geq 2\Delta$ Then, $v^*_{best}(\mathcal{F}_\mathcal{R}) \leq 3$.*

Proof. **Step 1:** Suppose that $g : \mathcal{X} \mapsto \mathcal{Y}$ such that there exists $y \in \mathcal{Y}$ such that $g(x) = y$ for all $x \in \mathcal{X}$. Now, let $x_0 \in \{2, \ldots, K-1\}$. Let r be any α-strongly convex function defined over $[1, K]$. Then, $x_0 \in \operatorname{intdom}(r)$. Then, there exists a subgradient at x, g. Let $z_1 = x_0 - 1$ and $z_2 = x_0 + 1$. Then, either $g \cdot (z_1 - x) \geq 0$ or $g \cdot (z_2 - x_0) \geq 0$. Suppose wlog that $g \cdot (z_1 - x_0) \geq 0$. Then, using α-strong convexity we have that

$$r(z_1) \geq r(x_0) + g^{\cdot}(z_1 - x_0) + \frac{\alpha}{2}|x_0 - z_1| \geq r(x_0) + 2\Delta$$

where we used $\alpha \geq 2\Delta$. This shows that for any α-strongly convex function defined over $[1, K]^d$, either $\mathcal{D}[r(z_1)] \neq \mathcal{D}[r(x_0)]$ or $\mathcal{D}[r(z_1)] \neq \mathcal{D}[r(x_0)]$. Thus, it suffices to query x_0, z_1, z_2 to finish this case.

Step 2: Now, suppose that $g : \mathcal{X} \mapsto \mathcal{Y}$ is such that there exists $x, x' \in \mathcal{X}$ such that $g(x) \neq g(x')$. Then, let $I \in \operatorname{argmin}_{i\in[n]} g(x_i)$. It suffices to query x_{I-1}, x_I, and x_{I+1}. By the prior step, we have that $g(x_{I-1}) > g(x_I) < g(x_{I+1})$ for any α-strongly convex function such that $\mathcal{D}[r(x)] = g(x)$ for all $x \in \mathcal{X}$. Thus, this completes the case.

□

J `OFUL` is not minimax-optimal

Fix $x_1, \ldots, x_{2n}$ and let $\mathcal{F} = \{f_1, \ldots, f_n\}$ where

$$f_i(x_j) = \begin{cases} 1 & j = i \\ .5 & j \in \{n+1, \ldots, n+i\} \\ 0 & \text{o/w} \end{cases} .$$

Consider the instance given by $y_i = f_n(x_i)$ for all $i \in [2n]$. Then, inspection reveals that `OFUL` selects all of the arms in $\{1, \ldots, n\}$ before terminating, obtaining a sample complexity of n. On the other hand, it is possible to solve any instance $f \in \mathcal{F}$ in $O(\log(n))$ queries using a binary search procedure..

K Additional Experiment and Implementation Details

All experiments were computed on a 36 core cluster machine, and all algorithms were implemented in python. To estimate the probabilities in the algorithm for sample selection, we used 300 independently drawn functions from the version space in the RKHS and constraints experiments and 500 in the case of convex functions. We used 50 iterations of hit and run for the RKHS and constraints experiments and 100 for the convex function experiment. Note that `GRAILS` divides the output space into intervals. In the RKHS experiments, we used 100 evenly spaced intervals in the output space of f^* for computational efficiency. For the constraints experiment, we divided the output space of f^* into 300 spaced for better sample complexity at the cost of computational efficiency. Lastly in the convex experiment, we used 1500 evenly spaced intervals.

Furthermore, note that hit and run must compute a projection onto the boundary of the convex set in order to choose the random step. As the step is along a uniformly chosen direction, we can compute this projection by bisection search. This was performed with tolerance 1e-6 for suitable performance with minimal computational overhead.

Lastly, our general implementation of hit and run requires a membership oracle for each function class. We implemented oracles for RKHS functions and 1d convex functions. In the case of the RKHS, we linearize the problem using the kernel trick by first computing the Grammian matrix and then projecting all data onto the span of the eigenvalues. Furthermore, in order to use hit and run, the MCMC iteration must begin with a point in the convex set. As our set is cut out by hyperplane and elliptical constraints, we achieve this by choosing a random starting point in and using CVXPy with the ECOS solver to compute the projection of that point into the convex set to generate a random stating point for hit and run.

For our implementation, we followed the update of [23] directly in the setting where the noise variance is 0. For our implementation of OFUL, we track our constraint sets using equality constraints for the observed data and norm constraints for the norm of the unknown parameters. Then we use CVXPy to compute the upper and lower bound on the function value of any point x_i.

References

[1] Kevin Jamieson and Lalit Jain. A bandit approach to multiple testing with false discovery control. *arXiv preprint arXiv:1809.02235*, 2018.

[2] Burr Settles. Active learning literature survey. 2009.

[3] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[4] Sanjoy Dasgupta, Daniel J Hsu, and Claire Monteleoni. *A general agnostic active learning algorithm*. Citeseer, 2007.

[5] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89, 2009.

[6] Steve Hanneke et al. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309, 2014.

[7] Marta Soare, Alessandro Lazaric, and Rémi Munos. Best-arm identification in linear bandits. In *Advances in Neural Information Processing Systems*, pages 828–836, 2014.

[8] Tanner Fiez, Lalit Jain, Kevin G Jamieson, and Lillian Ratliff. Sequential experimental design for transductive linear bandits. In *Advances in Neural Information Processing Systems*, pages 10666–10676, 2019.

[9] Julian Katz-Samuels, Lalit Jain, Kevin G Jamieson, et al. An empirical process approach to the union bound: Practical algorithms for combinatorial and linear bandits. *Advances in Neural Information Processing Systems*, 33, 2020.

[10] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

[11] Kareem Amin, Michael Kearns, and Umar Syed. Bandits, query learning, and the haystack dimension. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 87–106. JMLR Workshop and Conference Proceedings, 2011.

[12] Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil'ucb: An optimal exploration algorithm for multi-armed bandits. In *Conference on Learning Theory*, pages 423–439, 2014.

[13] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1):1–42, 2016.

[14] László Lovász and Santosh Vempala. Fast algorithms for logconcave functions: Sampling, rounding, integration and optimization. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 57–68. IEEE, 2006.

[15] Christopher Tosh and Daniel Hsu. Diameter-based interactive structure discovery. In *International Conference on Artificial Intelligence and Statistics*, pages 580–590. PMLR, 2020.

[16] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine learning*, 15(2):201–221, 1994.

[17] Sanjoy Dasgupta. Coarse sample complexity bounds for active learning. In *NIPS*, volume 18, pages 235–242, 2005.

[18] Christopher Tosh and Sanjoy Dasgupta. Diameter-based active learning. In *International Conference on Machine Learning*, pages 3444–3452. PMLR, 2017.

[19] Alina Beygelzimer, Daniel Hsu, John Langford, and Tong Zhang. Agnostic active learning without constraints. *arXiv preprint arXiv:1006.2588*, 2010.

[20] Julian Katz-Samuels, Jifan Zhang, Lalit Jain, and Kevin Jamieson. Improved algorithms for agnostic pool-based active classification. *International Conference on Machine Learning*, 2021.

[21] Tibor Hegedus. Generalized teaching dimensions and the query complexity of learning. In *Proceedings of the eighth annual conference on Computational learning theory*, pages 108–117, 1995.

[22] José L Balcázar, Jorge Castro, and David Guijarro. A new abstract combinatorial dimension for exact learning via queries. *Journal of Computer and System Sciences*, 64(1):2–21, 2002.

[23] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[24] Adam D Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(10), 2011.

[25] Romain Camilleri, Julian Katz-Samuels, and Kevin Jamieson. High-dimensional experimental design and kernel bandits. *International Conference on Machine Learning*, 2021.

[26] Lalit Jain and Kevin G Jamieson. A new perspective on pool-based active classification and false-discovery control. In *Advances in Neural Information Processing Systems*, pages 13992–14003, 2019.

[27] Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory*, pages 998–1027, 2016.

[28] David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.

[29] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *NIPS*, volume 11, pages 2312–2320, 2011.

[30] Laura P Swiler, Mamikon Gulian, Ari L Frankel, Cosmin Safta, and John D Jakeman. A survey of constrained gaussian process regression: Approaches and implementation challenges. *Journal of Machine Learning for Modeling and Computing*, 1(2), 2020.

[31] Tor Lattimore and Csaba Szepesvari. The end of optimism? an asymptotic analysis of finite-armed linear bandits. In *Artificial Intelligence and Statistics*, pages 728–737. PMLR, 2017.

[32] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer, 2001.

[33] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[34] Julian Katz-Samuels and Kevin Jamieson. The true sample complexity of identifying good arms. In *International Conference on Artificial Intelligence and Statistics*, pages 1781–1791. PMLR, 2020.

[35] Julian Katz-Samuels and Clay Scott. Feasible arm identification. In *International Conference on Machine Learning*, pages 2535–2543. PMLR, 2018.

[36] Julian Katz-Samuels and Clayton Scott. Top feasible arm identification. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1593–1601. PMLR, 2019.