Journal of Machine Learning Research 25 (2024) 1-42 Submitted 12/22; Revised 3/24; Published 9/24

Fast Rates in Pool-Based Batch Active Learning

Claudio Gentile CGENTILE@GOOGLE.COM
Google Research
New York City, NY, USA

Zhilei Wang ZHILEIWANG92@GMAIL.COM
WorldQuant LLC
New York City, NY, USA

Tong Zhang TOZHANG@ILLINOIS.EDU
University of Illinois Urbana-Champaign, IL

Editor: Sivan Sabato

Abstract

We consider a batch active learning scenario where the learner adaptively issues batches of points to a labeling oracle. Sampling labels in batches is highly desirable in practice due to the smaller number of interactive rounds with the labeling oracle (often human beings). However, batch active learning typically pays the price of a reduced adaptivity, leading to suboptimal results. In this paper we propose a solution which requires a careful trade off between the informativeness of the queried points and their diversity. We theoretically investigate batch active learning in the practically relevant scenario where the unlabeled pool of data is available beforehand (*pool-based* active learning). We analyze a novel stage-wise greedy algorithm and show that, as a function of the label complexity, the excess risk of this algorithm matches the known minimax rates in a standard statistical learning setting with linear function spaces. Our results also exhibit a mild dependence on the batch size. These initial results are then extended to hold for general function spaces with similar algorithmics. These are the first theoretical results that employ careful trade offs between informativeness and diversity to rigorously quantify the statistical performance of batch active learning in the pool-based scenario.

Keywords: Informativeness vs. diversity, Statistical learning, G-optimal design

1. Introduction

The aim of active learning is to reduce the data requirement of training processes through the careful selection of informative subsets of the data across several interactive rounds. This increased interactive power enables the adaptation of the sampling process to the actual state of the learning algorithm at hand, yet this benefit comes at the price of frequent re-training of the model and increased interactions with the labeling oracle (which is often just a pool of human labelers).

The *batch* mode of active learning is one where labels are queried in batches of suitable size, and the models are re-trained/updated either after each batch or even less frequently. This sampling mode often corresponds to the way labels are gathered in practical large-scale processing pipelines.

Batch active learning tries to strike a reasonable balance between the benefits of adaptivity and the costs associated with interaction and re-training. Yet, since the sampling is split into batches, and model updates can only be performed at the end of each batch, a batch active learning algorithm has to prevent to the extent possible the sampling of redundant points. The standard trade-off that arises is then to ensure that the sampled points are *informative* enough for the model, if taken in isolation, while at the same time being *diverse* enough so as to avoid sampling redundant labels.

We study batch active learning in the *pool-based* model, where an unlabeled pool of data is made available to the algorithm beforehand, and the goal is to single out a subset of the data so as to achieve the same statistical performance as if training were carried out on the entire pool. In this setting, we describe and analyze novel algorithms that obtain fast rates of convergence of their excess risk as a function of the number of requested labels. Interestingly enough, in the linear case, optimal rates are obtained even if we allow the batch size to grow with the pool size, the actual trade-off being ruled by the amount of noise in the data. Another appealing aspect is that our algorithms guarantee a number of re-training rounds which is at worst logarithmic, while being able to automatically adapt to the level of noise.

We operate in specific realizable settings, starting with linear models as a warmup, and then extending our results to the more general non-linear setting. Unlike what is traditionally done by many algorithmic solutions to active learning available in the literature (e.g., Balcan et al. (2007); Balcan and Long (2013); Zhang and Li (2021)), we do not formulate strong assumptions on the input distribution. We establish careful trade-offs between the informativeness and the diversity of the queried labels, and rigorously quantify the statistical performance on batch active learning in a noisy pool-based setting. To our knowledge, these are the first guarantees of this kind that apply to a noisy (hence realistic) batch pool-based active learning scenario. See also the related work contained in Section 3.

A short version of the current paper appeared in ICML 2022 (see Gentile et al., 2022b), and a longer version in Arxiv (Gentile et al., 2022a). The ICML version only contains the linear model analysis. This extended version greatly generalizes the linear analysis of Gentile et al. (2022b) to a broader class of nonlinear functions. This nonlinear analysis introduces a novel concept of eluder-like dimension of a function space, and an associated eluder-like confidence interval, which can have other applications beyond active learning—see, e.g., the recent paper by Ye et al. (2023), which built upon the Arxiv version of this paper (Gentile et al., 2022a).

1.1 Content and contributions

Our contributions can be described as follows.

1. We first present an efficient algorithm for pool-based batch active learning for noisy linear models (Algorithm 1). This algorithm generates pseudo-labels by computing sequences of linear classifiers that restrict their attention to exponentially small regions of the margin space, and then trains a single model based on the pseudo-labels only. The design inspiring the sampling within each stage is a G-optimal design, computed through a greedy strategy. We show (Theorem 1) that under the standard i.i.d. assumption of the (input, label) pairs, the model so trained enjoys an excess risk bound

with respect to the Bayes optimal predictor which is best possible, when expressed in terms of the total number of requested labels. The number of re-training stages (that is, the number of linear classifiers computed to generate pseudo-labels) is at most logarithmic in the pool size and, importantly, it automatically adapts to the noise level without knowing it in advance.

2. Since the above algorithm does not operate on a constant batch size B, we show in Section 4.2 an easy adaptation to the constant batch size, and make the observation that B therein may also scale as T^{β}, for some exponent $\beta < 1$ that depends on the amount of noise (see comments surrounding Corollary 2), still retaining the above-mentioned optimal rates.

3. Our algorithmic technique can be seen as a skeleton technique that can be applied to more general situations, provided the estimators employed at each stage and the diversity measure guiding the design have matching properties. We provide extensions to the general nonlinear case in Section 5, and give an analysis that covers general non-linear function spaces (Theorem 3). The non-linear extension relies on a novel notion of eluder-like dimension of a function space, which allows us to show learnability whenever the function space has eluder dimension at level ϵ of the form $\texttt{poly}(1/\epsilon)$. Also this algorithm can be made to work in the constant batch size case, and the same considerations as in the linear case apply to the general nonlinear case.

2. Preliminaries and Notation

We denote by $\mathcal{X}$ the input space (e.g., $\mathcal{X} = \mathbb{R}^d$), by $\mathcal{Y}$ the output space, and by $\mathcal{D}$ an unknown distribution over $\mathcal{X} \times \mathcal{Y}$. The corresponding random variables will be denoted by $\mathbf{x}$ and y. We also denote by $\mathcal{D}_{\mathcal{X}}$ the marginal distribution of $\mathcal{D}$ over $\mathcal{X}$. Given a function h (also called a *hypothesis* or a *model*) mapping $\mathcal{X}$ to $\mathcal{Y}$, the *population loss* (often referred to as *risk*) of h is denoted by $\mathcal{L}(h)$, and defined as $\mathcal{L}(h) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[loss(h(x), y)]$, where $loss(\cdot,\cdot) : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ is a given *loss* function. For simplicity of presentation, we restrict ourselves to a binary classification setting with 0-1 loss, so that $\mathcal{Y} = \{-1, +1\}$, and $loss(\hat{y}, y) = \mathbb{1}\{\hat{y} \neq y\} \in \{0, 1\}$, being $\mathbb{1}\{\cdot\}$ the indicator function of the predicate at argument. When clear from the surrounding context, we will omit subscripts like "$(\mathbf{x}, y) \sim \mathcal{D}$" from probabilities and expectations.

We are given a class of models $\mathcal{F} = \{f : \mathcal{X} \to [0, 1]\}$ and the Bayes optimal predictor $h^*(x) = \mathrm{sgn}\,(f^*(x) - 1/2)$, where

$$f^*(\mathbf{x}) = \mathbb{P}(y = 1|\mathbf{x})$$

is assumed to belong to class $\mathcal{F}$ (the so-called *realizability* assumption). This assumption is reasonable whenever the model class $\mathcal{F}$ we operate on is wide enough. For instance, a realizability (or quasi-realizability) assumption seems natural in overparameterized settings implemented by nowadays' Deep Neural Networks.

A simple example is a generalized linear model

$$f^*(\mathbf{x}) = \sigma(\langle \mathbf{w}^*, \mathbf{x}\rangle)\ , \tag{1}$$

where $\sigma : \mathbb{R} \to [0,1]$ is a suitable sigmoidal function, e.g., $\sigma(z) = \frac{e^z}{1+e^z}$, $\mathbf{w}^*$ is an unknown vector in $\mathbb{R}^d$, with bounded (Euclidean) norm $||\mathbf{w}|| \leq R$ for some $R \geq 1$, and $\langle \cdot, \cdot \rangle$ denotes the usual inner product in $\mathbb{R}^d$.

Throughout this paper, we adopt the commonly used low-noise condition on the marginal distribution $\mathcal{D}_{\mathcal{X}}$ of Mammen and Tsybakov (1999): there are constant $c > 0$, $\epsilon_0 \in (0,1]$ and exponent $\alpha \geq 0$ such that for all $\epsilon \in (0, \epsilon_0]$ we have

$$\mathbb{P}\big(|f^*(\mathbf{x}) - 1/2| < \epsilon/2\big) \leq c\,\epsilon^{\alpha} \ . \tag{2}$$

Note, in particular, that $\alpha \to \infty$ gives the so-called *hard margin* condition $\mathbb{P}\big(|f^*(\mathbf{x}) - 1/2| < \epsilon\big) = 0$ for all $\epsilon \leq \epsilon_0$ while, at the opposite end of the spectrum, exponent $\alpha = 0$ (and $c = 1$) corresponds to making *no assumptions whatsoever* on $\mathcal{D}_{\mathcal{X}}$. For simplicity, we shall assume throughout that the above low-noise condition holds for $c = 1$. The noise exponent α and range constant ϵ_0 are typically unknown, and our algorithms will not rely on the prior knowledge of them.

We are given a class of models $\mathcal{F}$, and a pool $\mathcal{P}$ of T unlabeled instances $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \mathcal{X}$, drawn i.i.d. according to a marginal distribution $\mathcal{D}_{\mathcal{X}}$ obeying condition (2) (with $c = 1$). The associated labels $y_1, \ldots, y_T \in \mathcal{Y}$ are such that the pairs $(\mathbf{x}_t, y_t)$, $t = 1, \ldots, T$, are drawn i.i.d. according to $\mathcal{D}$, the labels being generated according to the conditional distribution determined by some $f^* \in \mathcal{F}$. The labels are not initially revealed to us, and the goal of the active learning algorithm is to come up at the end of training with a model $\widehat{h} : \mathcal{X} \to \mathcal{Y}$ whose *excess risk* $\mathcal{L}(\widehat{h}) - \mathcal{L}(h^*)$ is as small as possible, while querying as few labels as possible for points in $\mathcal{P}$.

The way labels are queried follows the standard batch active learning protocol. We are given a *batch size* $B \geq 1$. Label acquisition and learning proceeds in a sequence of *stages*, $\ell = 1, 2, \ldots$. At each stage ℓ, the algorithm is allowed to query B-many labels by only relying on labels acquired in the past $\ell - 1$ stages. Note that each point $\mathbf{x}_t$ in pool $\mathcal{P}$ can only be queried once, which is somehow equivalent to assuming that the noise in the corresponding label y_t is *persistent*. We shall henceforth denote by $N_T(\mathcal{P})$ the total number of labels (sometimes referred to as *label complexity*) queried by the algorithm at hand on pool $\mathcal{P}$, and by $N_{T,B}(\mathcal{P})$ the same quantity if we want to emphasize the dependence on B.

The analysis of our algorithms hinges upon a suitable measure of *diversity*, $D(\mathbf{x}, S)$, that quantifies how far off a data point $\mathbf{x} \in \mathcal{X}$ is from a finite set of points $S \subseteq \mathcal{X}$. Though many diversity measures may be adopted for practical purposes (e.g., Wei et al., 2015; Sener and Savarese, 2018; Kirsch et al., 2019; Ash et al., 2020; Killamsetty et al., 2021; Kirsch et al., 2021; Citovsky et al., 2021), the one enabling tight theoretical analyses for our algorithms is one that is somehow coupled with the estimators our active learning algorithms rely upon. Specifically, given diversity measure $D(\mathbf{x}, S)$, an estimator $\widehat{f} = \widehat{f}(S)$ in a fixed design scenarios is coupled with $D(\mathbf{x}, S)$ if we can guarantee L_∞ approximation bounds of the form

$$|\widehat{f}_S(\mathbf{x}) - f^*(\mathbf{x})| \leq D(\mathbf{x}, S) \qquad \forall \mathbf{x} \ . \tag{3}$$

In the case of linear function spaces over $\mathcal{X} = \mathbb{R}^d$, the estimators $\widehat{f} = \widehat{f}(S)$ will essentially be least-squares predictors, and a coupled diversity measure will be spectral-like: $D(\mathbf{x}, S) = \langle \mathbf{x}, \mathbf{x} \rangle^{\frac{1}{2}}_{A_S^{-1}} = ||\mathbf{x}||_{A_S^{-1}} = \sqrt{\mathbf{x}^\top A_S^{-1} \mathbf{x}}$, that is, the Mahalanobis norm of $\mathbf{x}$ w.r.t. the positive semi-definite matrix A_S^{-1}, where $A_S = I + \sum_{\mathbf{z} \in S} \mathbf{z}\mathbf{z}^\top$, being I the $d \times d$ identity matrix.

Note that $D(\mathbf{x}, S)$ is large when $\mathbf{x}$ is aligned with small eigenvectors of A_S, while it is small if $\mathbf{x}$ is aligned with large eigenvectors of that matrix. In particular, $D(\mathbf{x}, S)$ achieves its maximal value $||\mathbf{x}||^2$ when $\mathbf{x}$ is *orthogonal* to the space spanned by S. Hence, $\mathbf{x}$ is "very different" from S as measured by $D(\mathbf{x}, S)$ if $\mathbf{x}$ contributes a direction of the input space which is not already spanned by S. We denote by $|A_S|$ the determinant of matrix A_S.

In the more general nonlinear case, our diversity measure is similar in spirit to the *eluder dimension* of Russo and Van Roy (2013), as well as to the more recent *online decoupling coefficient*, as defined by Dann et al. (2021)—see Section 5 for details.

At an intuitive level, since the label requests are batched, and model updates are typically performed only at the end of each stage, a batch active learning algorithm is compelled to operate within each stage by trading off the (predicted) informativeness of the selected labels against the diversity of the data points whose labels are requested. Moreover, the larger the batch size B the less adaptive the algorithm is forced to be, hence we expect B to somehow play a role in the performance of the algorithm.

From a practical standpoint, there are indeed two separate notions of adaptivity to consider. One is the number of interactive rounds with the labeling oracle, the other is the number of times we *re-train* (or update) a model based on the labels gathered during the interactive rounds. The two notions *need not* coincide. While the former essentially accounts for the cost of interacting with human labelers, the latter is more related to the cost of re-training/updating a (potentially very complex) learning system.

3. Related Work

While experimental studies on batch active learning are reported since the early 2000s (see, e.g., Hoi et al., 2006), it is only with the deployment at scale of deep neural networks that we have seen a general resurgence of interest in active learning, and batch active learning in particular. The batch pool-based model studied here is the one that has spurred the widest attention, as it corresponds to the way in practice labels are gathered in large-scale processing pipelines. This interest has generated a flurry of recent investigations, mainly of experimental nature, yet containing a lot of interesting and diverse approaches to batch active learning. Among these are the papers by Gu et al. (2012, 2014); Sener and Savarese (2018); Kirsch et al. (2019); Zhdanov (2019); Shui et al. (2020); Ash et al. (2020); Kim et al. (2020); Killamsetty et al. (2021); Kirsch et al. (2021); Ghorbani et al. (2021); Citovsky et al. (2021); Kothawade et al. (2021).

On the theoretical side, active learning is a well-studied sub-field of statistical learning. General references in pool-based active learning include the papers by Dasgupta (2004, 2005); Hanneke (2014); Nowak (2011); Tosh and Dasgupta (2017), and specific algorithms for half-spaces under classes of input distributions are contained, e.g., in Balcan et al. (2007); Balcan and Long (2013); Zhang and Li (2021). However, none of these papers tackle the practically relevant scenario of *batch* active learning. In fact, restricting to theoretical aspects of batch active learning makes the research landscape far less populated. Below we briefly summarize what we think are among the most relevant papers to our work, as directly related to batch active learning, and then mention recent efforts in contiguous fields, like adaptive sampling and subset selection, which may serve as a general reference and inspiration.

Batch active learning in the pool-based scenario is one of the motivating applications in Chen and Krause (2013), where the main concern is to investigate general conditions under which a batch greedy policy achieves similar performance as the optimal policy that operates with the same batch size. Yet, the authors consider simple noise free scenarios, while the important observation (Theorem 2 therein) that a batch greedy algorithm is also competitive with respect to an optimal fully sequential policy (batch size one) does not apply to active learning. Chen et al. (2015, 2017) are along similar lines, with the addition of persistent noise, but do not tackle batch active learning problems.

A paper with a similar aim as ours, yet operating in the streaming setting of active learning, is Amin et al. (2020). The authors show that some classes of fully sequential active algorithms can be turned into sequential algorithms that query labels in batches and suffer only an additive (times log factors) overhead in the label complexity. This transformation is essentially obtained by freezing the state of the fully sequential algorithm, but it is unclear whether any notion of diversity over the batch is enforced by the resulting batch algorithms.

A recent stream-based active learning paper that is worth mentioning is Camilleri et al. (2021b), which shares a similar method and modeling assumptions as ours in leveraging optimal design, but it does not deal with batch active learning. The main concern there is essentially to improve the performance of adaptive sampling by reducing the variance of the estimators. Moreover, a related work of Katz-Samuels et al. (2021) considered the batch active learning setting for binary classification, with 0-1 valued hypothesis class trained via binary classification loss instead of regression loss of this paper. Confidence interval resulting from binary classification loss minimization requires a concept similar to disagreement coefficient (Hanneke, 2014), which is quite different from regression confidence interval in statistics which we employ here. Therefore their problem setting and results are not comparable to ours.

Another couple of very recent works in the streaming setting of active learning which are relevant to this paper (specifically, for the general function space setting) are Zhu and Nowak (2022) and Sekhari et al. (2023). Both papers assume a sequential i.i.d. setting, hence they are not batch pool-based. Being i.i.d. sequential (and not pool-based), their algorithms are able to sidestep the dependence on our eluder dimension-like notion of complexity, and instead rely on the so-called value-function disagreement coefficient (Foster et al., 2020), which is likely to be smaller than eluder-like dimension complexity measures. Again, those results are incomparable to ours.

Our algorithms borrow ideas from the classical experimental design literature (Pronzato and Pázman, 2013), and is also motivated by works in active learning, such as Chaudhuri et al. (2015); Mukherjee et al. (2022). However, these works consider different objective functions than us. Specifically, they try to find designs with finite sample guarantees that can minimize the log-loss or regression loss. In such settings, as long as there is noise, they cannot achieve faster convergence rates than $O(1/N_T(\mathcal{P}))$. In contrast, our batch active learning setting directly considers the classification objective, which can potentially achieve faster rates than the corresponding regression problem when low noise conditions are satisfied. Our algorithms employ stage-wise batch sample selection and elimination to achieve such faster rates.

A learning problem somewhat similar to pool-based batch active learning is *training subset selection* (sometimes called dataset summarization), whose goal is to come up with a compressed version of a (big) dataset that offers to a given learning algorithm the same inference capabilities as if applied to the original dataset. The problem can be organized in rounds (as in batch active learning) and bridging one to the other can in practice be done by label hallucination/pseudo-labeling. Representative works include Wei et al. (2015); Killamsetty et al. (2021); Borsos et al. (2021).

A final word before delving into the technical sections: Our work is theoretical in nature. We try to understand the statistical sample efficiency of pool based batch active learning, although some of the proposed methods in the paper may not pay specific attention to the computational aspects of the involved procedures. Specifically, whereas our methods are computationally efficient in the linear function case, they need not be in the general nonlinear case.

4. Warmup: The Linear Case

As a warm up, we start by considering a simple linear model of the form $f^*(\mathbf{x}) = \frac{1+\langle \mathbf{w}^*, \mathbf{x} \rangle}{2}$, where both $\mathbf{w}^*$ and $\mathbf{x}$ lie in the d-dimensional Euclidean unit ball (so that $\langle \mathbf{w}^*, \mathbf{x} \rangle \in [-1, 1]$ and $f^*(\mathbf{x}) \in [0, 1]$). Algorithm 1 contains in a nutshell the main ideas behind our algorithmic solutions, which is to greedily approximate a G-optimal design in the selection of points at each stage. The way it is formulated, Algorithm 1 does not operate with a constant batch size B per stage. We will reduce to the constant batch size case in Section 4.2. The same algorithmic skeleton will be applied to the general nonlinear case in Section 5.

The algorithm for the linear case takes as input a finite pool of points $\mathcal{P}$ of size T and proceeds across stages $\ell = 1, 2, \ldots$ by generating at each stage ℓ a (linear-threshold) predictor $\mathrm{sgn}(\langle \mathbf{w}_\ell, \mathbf{x} \rangle)$, where $\mathbf{w}_\ell$ is a ridge regression estimator computed only on the labeled pairs $(\mathbf{x}_{\ell,1}, y_{\ell,1}), \ldots, (\mathbf{x}_{\ell,T_\ell}, y_{\ell,T_\ell})$ collected during that stage. These predictors are used to trim the current pool $\mathcal{P}_{\ell-1}$ by eliminating both the points on which $\mathbf{w}_\ell$ is itself confident (set $\mathcal{C}_\ell$) and those whose labels have just been queried (set $\mathcal{Q}_\ell$). At each stage ℓ, the points $\mathbf{x}_{\ell,t}$ to query are selected in a greedy fashion by maximizing[1] $D(\mathbf{x}, \mathcal{Q}_\ell) = ||\mathbf{x}||_{A^{-1}_{\ell,t-1}}$ over the current pool $\mathcal{P}_{\ell-1}$ (excluding the already selected points $\mathcal{Q}_\ell$, which are contained in $A_{\ell,t-1}$), so as to make $\mathbf{x}_{\ell,t}$ maximally different from $\mathcal{Q}_\ell$.

When stage ℓ terminates, we are guaranteed that we have collected a set of points $\mathcal{Q}_\ell$ such that all remaining points $\mathbf{x}$ in the pool satisfy $D(\mathbf{x}, \mathcal{Q}_\ell) \le \epsilon_\ell$. Threshold ϵ_ℓ, defined at the beginning of the stage, is exponentially decaying with ℓ. It is this threshold that determines the actual length of the stage, and rules the elimination of unqueried points from the pool, along with the corresponding generation of pseudo-labels during the stage.

Algorithm 1 stops generating new stages when the size $|\mathcal{P}_\ell|$ of pool $\mathcal{P}_\ell$ triggers the condition $d/2^{-\ell+1} > 2^{-\ell+1}|\mathcal{P}_\ell|$ (which is satisfied, in particular, when $\mathcal{P}_\ell$ becomes empty). In that case, the current stage ℓ becomes the final stage L.

Finally, the algorithm uses the subset of points $\cup_{\ell=1}^{L} \mathcal{C}_\ell$ and the associated pseudo-labels $\hat{y}$ generated during the L stages to train a linear classifier $\widehat{\mathbf{w}}$ (e.g., a support vector machine) to zero empirical error on that subset. Our analysis (see Appendix A) shows that with high probability such a consistent linear classifier exists. Each point $\mathbf{x}$ that remains in the pool,

[1] As a matter of fact, the chosen $\mathbf{x}_{\ell,t}$ need not be the maximizer of $D(\mathbf{x}, \mathcal{Q}_\ell)$, the analysis only requires $D(\mathbf{x}_{\ell,t}, \mathcal{Q}_\ell) > \epsilon_\ell$.

Algorithm 1: Pool-based batch active learning algorithm for linear models.

```
**Input:** Confidence level $\delta \in (0,1]$, pool of instances $\mathcal{P} \subseteq \mathbb{R}^d$ of size $|\mathcal{P}| = T$
**Initialize:** $\mathcal{P}_0 = \mathcal{P}$
**for** $\ell = 1, 2, \ldots,$
    Initialize within stage $\ell$:
        • $\epsilon_\ell = 2^{-\ell} / (\sqrt{2 \log \frac{2\ell(\ell+1)T}{\delta}} + 1)$
        • $A_{\ell,0} = I, \;\; t = 0, \;\; \mathcal{Q}_\ell = \emptyset$
    **while** $\mathcal{P}_{\ell-1} \backslash \mathcal{Q}_\ell \neq \emptyset$ *and* $\max_{\mathbf{x} \in \mathcal{P}_{\ell-1} \backslash \mathcal{Q}_\ell} \|\mathbf{x}\|_{A^{-1}_{\ell,t}} > \epsilon_\ell$
        • $t = t + 1$
        • Pick $\mathbf{x}_{\ell,t} \in \operatorname{argmax}_{\mathbf{x} \in \mathcal{P}_{\ell-1} \backslash \mathcal{Q}_\ell} \|\mathbf{x}\|_{A^{-1}_{\ell,t-1}}$
        • Update $\;\; A_{\ell,t} = A_{\ell,t-1} + \mathbf{x}_{\ell,t} \mathbf{x}_{\ell,t}^\top$
        • $\mathcal{Q}_\ell = \mathcal{Q}_\ell \cup \{\mathbf{x}_{\ell,t}\}$
    Set $T_\ell = t$, the number of queries made in stage $\ell$
    **if** $\mathcal{Q}_\ell \neq \emptyset$
        • Query the labels $y_{\ell,1}, \ldots, y_{\ell,T_\ell}$ associated with the unlabeled data in $\mathcal{Q}_\ell$,
          and compute
              $\mathbf{w}_\ell = A^{-1}_{\ell,T_\ell} \sum_{t=1}^{T_\ell} y_{\ell,t} \mathbf{x}_{\ell,t}$
        • Set $\;\; \mathcal{C}_\ell = \{\mathbf{x} \in \mathcal{P}_{\ell-1} \backslash \mathcal{Q}_\ell : |\langle \mathbf{w}_\ell, \mathbf{x} \rangle| > 2^{-\ell}\}$
        • Compute pseudo-labels on each $\mathbf{x} \in \mathcal{C}_\ell$ as $\hat{y} = \operatorname{sgn}\langle \mathbf{w}_\ell, \mathbf{x} \rangle$
    **else**
        $\mathbf{w}_\ell = \mathbf{0}$, $\mathcal{C}_\ell = \emptyset$
    Set $\mathcal{P}_\ell = \mathcal{P}_{\ell-1} \backslash (\mathcal{C}_\ell \cup \mathcal{Q}_\ell)$
    **if** $d/2^{-\ell+1} > 2^{-\ell+1} |\mathcal{P}_\ell|$
        • $L = \ell$
        • Exit the for-loop ($L$ is the total number of stages)
Predict labels in pool $\mathcal{P}$:
    • Train an SVM classifier $\widehat{\mathbf{w}}$ on $\cup_{\ell=1}^{L} \mathcal{C}_\ell$ via the generated pseudo-labels $\hat{y}$
    • Predict on each $\mathbf{x} \in (\cup_{\ell=1}^{L} \mathcal{Q}_\ell) \cup \mathcal{P}_L$ through $\operatorname{sgn}(\langle \widehat{\mathbf{w}}, \mathbf{x} \rangle)$
```

that is, each $\mathbf{x} \in (\cup_{\ell=1}^{L} \mathcal{Q}_\ell) \cup \mathcal{P}_L$, is assigned label $\mathrm{sgn}(\langle \widehat{\mathbf{w}}, \mathbf{x} \rangle)$. Notice, in particular, that $\widehat{\mathbf{w}}$ is not trying to fit the queried labels of $\cup_{\ell=1}^{L} \mathcal{Q}_\ell$, but only the pseudo-labels of $\cup_{\ell=1}^{L} \mathcal{C}_\ell$.

It is also worth observing how Algorithm 1 resolves the trade-off between informativeness and diversity we alluded to in previous sections. Once we reach stage ℓ, what remains in the pool are only the points $\mathbf{x}$ such that $|\langle \mathbf{w}_{\ell-1}, \mathbf{x} \rangle| \leq 2^{-\ell+1}$ (this is because we have eliminated in stage $\ell - 1$ all the points in $\mathcal{C}_{\ell-1}$). Hence, the remaining points which the approximate G-optimal design operates with in stage ℓ are those which the previous model $\mathbf{w}_{\ell-1}$ is not sufficiently confident on. The algorithm then puts all these low-confident points on the same footing (that is, they are considered equally informative if taken in isolation), and then relies on the approximate G-optimal design scheme to maximize diversity among them. The set-wise diversity measure we end up maximizing is indeed a determinant-like diversity measure. This is easily seen from the fact that $\sum_{t=1}^{T_\ell} ||\mathbf{x}_{\ell,t}||^2_{A^{-1}_{\ell,t-1}} \approx \log |A_{\ell,T_\ell}|$.

On one hand, this careful selection of points contributes to keeping the variance of estimator $\mathbf{w}_\ell$ under control. On the other hand, the fact that we stop accumulating labels when $\max_{\mathbf{x} \in \mathcal{P}_{\ell-1} \setminus \mathcal{Q}_\ell} \|\mathbf{x}\|_{A^{-1}_{\ell,T_\ell}} \leq \epsilon_\ell$ essentially implies that $\mathrm{sgn}(\langle \mathbf{w}_\ell, \mathbf{x} \rangle) = \mathrm{sgn}(\langle \mathbf{w}^*, \mathbf{x} \rangle)$ on all points $\mathbf{x}$ we generate pseudo-labels for, which in turn ensures that these pseudo-labels are consistent with $\mathbf{w}^*$.

Sequential experimental design has become popular, e.g., in the (contextual) bandits literature, see Ch. 22 in Lattimore and Szepesvari (2020), and is explicitly contained in recent works on best arm identification (e.g., Fiez et al., 2019; Camilleri et al., 2021a). Notice that in those works a design is a distribution over the set of actions (which would correspond to pool $\mathcal{P}$ in our case), and the algorithm is afforded to sample a given action $\mathbf{x}_t$ *multiple times*, obtaining each time a fresh reward value y_t such that $\mathbb{E}[y_t \,|\, \mathbf{x}_t] = \langle \mathbf{w}^*, \mathbf{x}_t \rangle$. This is not conceivable in a pool-based active learning scenario where label noise is persistent, and each "action" $\mathbf{x}_t$ can only be played once. This explains why the design we rely upon here is necessarily more restrained than in those papers.

4.1 Analysis

The following result applies to the linear function case.[2]

Theorem 1 *Let $T \geq d$ and assume that $\|\mathbf{x}\|_2 \leq 1$ for all $\mathbf{x} \in \mathcal{P}$. Then with probability at least $1 - \delta$ over the random draw of $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T) \sim \mathcal{D}$ the excess risk $\mathcal{L}(\widehat{\mathbf{w}}) - \mathcal{L}(\mathbf{w}^*)$, the label complexity $N_T(\mathcal{P})$, and the number of stages L generated by Algorithm 1 are simultaneously upper bounded as follows:*

$$\mathcal{L}(\widehat{\mathbf{w}}) - \mathcal{L}(\mathbf{w}^*) \leq \bar{C}\, C(\delta, T, \epsilon_0) \left(\max\left\{ \left(\frac{d}{T}\right)^{\frac{\alpha+1}{\alpha+2}},\ \frac{d}{T\epsilon_0} \right\} + \frac{\log\left(\frac{\log T}{\delta}\right)}{T} \right),$$

$$N_T(\mathcal{P}) \leq \bar{C}\, C(\delta, T, \epsilon_0) \left(\max\left\{ d^{\frac{\alpha}{\alpha+2}} T^{\frac{2}{\alpha+2}},\ \frac{d}{\epsilon_0^2} \right\} + \log^2\left(\frac{\log T}{\delta}\right) \right),$$

$$L \leq \bar{C} \left(\max\left\{ \frac{\log\left(\frac{T}{d}\right)}{\alpha+2},\ \log\left(\frac{4}{\epsilon_0}\right) \right\} + \log\left(\frac{\log T}{\delta}\right) \right),$$

[2] Detailed proofs are deferred to the appendices.

for an absolute constant $\bar{C}$ *and*

$$C(\delta, T, \epsilon_0) = \log^2\left(\frac{T}{\delta}\right)\left(1 + \log^2\left(\frac{1}{\epsilon_0}\right)\right) \ .$$

4.2 Constant batch size

We now describe a simple modification to Algorithm 1 that makes it work in the constant batch size case. Let us denote by T_ℓ the length of stage ℓ in Algorithm 1. The modified algorithm simply runs Algorithm 1: If $T_\ell < B$ the modified algorithm relies on model $\mathbf{w}_\ell$ generated by Algorithm 1 without saturating the budget of B labels at that stage. On the contrary, if $T_\ell \geq B$, the modified algorithm splits stage ℓ of Algorithm 1 into $\lceil T_\ell/B \rceil$ stages of size B (except, possibly, for the last one), and then uses the queried set $\mathcal{Q}_\ell$ generated by Algorithm 1 across all those stages. Hence, in this case, the modified algorithm is not exploiting the potential benefit of updating the model every B queried labels. For instance, if $B = 100$ and $T_\ell = |\mathcal{Q}_\ell| = 240$, the modified algorithm will split this stage into three successive stages of size 100, 100, and 40, respectively, and then rely on the 240 labels queried by Algorithm 1 across the three stages. In particular, the update of the model $\mathbf{w}_\ell$, and the associated pseudo-label computation on sets $\mathcal{C}_\ell$ is only performed at the *end* of the third stage.

Notice that the modified algorithm we just described is a legitimate pool-based batch active learning algorithm operating on a constant batch size B, and its analysis is a direct consequence of the one in Theorem 1, after we take care of the possible over-counting that may arise in the reduction. Specifically, observe that the final hypothesis $\widehat{\mathbf{w}}$ produced by the modified algorithm is the same as the one computed by Algorithm 1, hence the same bound on the excess risk applies. As for label complexity, if we stipulate that a batch algorithm operating on a constant batch size B will be billed B labels at each stage even if it ends up querying less, then the label complexity of the modified algorithm will over-count the number of labels simply due to the rounding off in $\lceil T_\ell/B \rceil$. However, at each of the L stages of Algorithm 1, the over-counting is bounded by B, so that, overall, the label complexity of the constant batch size variant exceeds the one of Algorithm 1 by at most an additive BL term which, due to the bound on L in Theorem 1, is of the form $\max\left\{\frac{B}{\alpha+2}\log\left(\frac{T}{d}\right), B\log\left(\frac{1}{\epsilon_0}\right)\right\}$. This is summarized in the following corollary.

Corollary 2 *With the same assumptions and notation as in Theorem 1, with probability at least* $1-\delta$ *over the random draw of* $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T) \sim \mathcal{D}$ *the label complexity* $N_{T,B}(\mathcal{P})$ *achieved by the modified algorithm operating on a batch of size B is bounded as follows:*

$$N_{T,B}(\mathcal{P}) \leq \bar{C}\, C(\delta, T, \epsilon_0)\left(\max\left\{d^{\frac{\alpha}{\alpha+2}} T^{\frac{2}{\alpha+2}},\ \frac{d}{\epsilon_0^2}\right\} + \log^2\left(\frac{\log T}{\delta}\right)\right)$$
$$+ B\,\bar{C}\left(\max\left\{\frac{\log\left(\frac{T}{d}\right)}{\alpha+2},\ \log\left(\frac{4}{\epsilon_0}\right)\right\} + \log\left(\frac{\log T}{\delta}\right)\right) \ ,$$

where $\bar{C}, C(\delta, T, \epsilon_0)$ *are the same as in Theorem 1.*

A few comments are in order.

1. An important practical aspect of this modified algorithm (inherited from Algorithm 1) is the very mild number of re-trainings required to achieve the claimed performance. Despite the total number of labels can be as large as $T^{\frac{2}{\alpha+2}}$, the number L of times the model is actually re-trained is not $T^{\frac{2}{\alpha+2}}/B$, but only *logarithmic* in T, irrespective of the noise level α (that is, even when the low-noise assumption (2) is vacuous). On the other hand, it is also important to observe that the bound on L shrinks as α increases, that is, when the problem becomes easier. Overall, these properties make the algorithm attractive in practical learning scenarios where the re-training time turns out to be the main bottleneck in the data acquisition process, and a learning procedure is needed that automatically adapts the re-training effort to the hardness of the problem.

2. Let us disregard lower order terms and only consider the asymptotic behavior as $T \to \infty$. Comparing the excess risk bound in Theorem 1 to the label complexity bound in Corollary 2, one can see that when $B = O(T^{\frac{2}{\alpha+2}})$ we have with high probability

$$\mathcal{L}(\widehat{\mathbf{w}}) - \mathcal{L}(\mathbf{w}^*) \approx \frac{1}{(N_{T,B}(\mathcal{P}))^{\frac{1+\alpha}{2}}} \ ,$$

   which is the minimax rate one can achieve for VC classes under the low-noise condition (2) with exponent α (e.g., Castro and Nowak, 2008; Hanneke, 2009; Koltchinskii, 2010; Dekel et al., 2012). Hence, in order to achieve high-probability minimax rates, one need not try to make the algorithm *more adaptive* by having it operate with an even smaller B: any B as small as $T^{\frac{2}{\alpha+2}}$ will indeed suffice in our learning scenario.

3. Similar minimax bounds on excess risk against label complexity for linear function spaces have been shown in the streaming setting in Dekel et al. (2012); Wang et al. (2021), though their results only hold in the fully sequential case (that is, $B = 1$) and only hold in expectation over the random draw of the data, not with high probability.

The fact that a batch greedy algorithm can be competitive with a fully sequential policy has also been observed in problems which are similar in spirit to active learning, like influence maximization (see, in particular, Chen and Krause, 2013). More recently, in the context of adaptive sequential decision making, Esfandiari et al. (2021) have proposed an efficient semi-adaptive policy that performs logarithmically-many rounds of interaction achieving similar performance as the fully sequential policy. This paper improves on the original ideas contained in Golovin and Krause (2017). Yet, when adapted to active learning, these results turn out to apply to very stylized scenarios that assume lack of noise in the labels, and/or disregard the computational aspects associated with maintaining a posterior distribution or a version space (which would be of size $O(T^d)$ in our case).

5. The Nonlinear Case

We are now ready to consider the more general non-linear scenario described in Section 2.

Given an arbitrary set of points $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ (not necessarily i.i.d. and possibly with repeated items), denote by $P(\cdot)$ a permutations over the indices $\{1, \ldots, T\}$, and

$\langle \mathbf{x}_{P(1)}, \ldots, \mathbf{x}_{P(T)} \rangle$ the permutation of S corresponding to P. We define the dimension $\mathcal{D}im(\mathcal{F}, S)$ of the class of functions $\mathcal{F}$ projected onto S as

$$\mathcal{D}im(\mathcal{F}, S) = \sup_P \sum_{t=1}^{T} D^2\Big(\mathbf{x}_{P(t)}; \langle \mathbf{x}_{P(1)}, \ldots, \mathbf{x}_{P(t-1)} \rangle\Big),$$

where

$$D^2(\mathbf{x}, \langle \mathbf{x}_1, \ldots, \mathbf{x}_{t-1} \rangle) = \sup_{f,g \in \mathcal{F}} \frac{(f(\mathbf{x}) - g(\mathbf{x}))^2}{\sum_{i=1}^{t-1}(f(\mathbf{x}_i) - g(\mathbf{x}_i))^2 + 1} \ .$$

The function $D(\mathbf{x}, \langle \mathbf{x}_1, \ldots, \mathbf{x}_{t-1} \rangle)$ defined above is our ($\mathcal{F}$-dependent) measure of diversity of $\mathbf{x}$ from $\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}$, and will be used to select diverse points $\mathbf{x}$ to query in each batch.

We will give below notable examples where $\mathcal{D}im(\mathcal{F}, S)$ is suitably bounded. For now, what is important to keep in mind is that our active learning analyses will be non-vacuous whenever

$$\mathcal{D}im_T(\mathcal{F}) = \sup_{S \subseteq \mathcal{X} \,:\, |S|=T} \mathcal{D}im(\mathcal{F}, S)$$

is *sub-linear* in T, e.g., $\mathcal{D}im_T(\mathcal{F}) = O(poly(\log T))$ or $\mathcal{D}im_T(\mathcal{F}) = O(T^\zeta)$, for $\zeta \in [0, 1)$. A sufficient condition for this to happen is discussed next.

The reader familiar with the literature in non-linear contextual bandits will recognize $\mathcal{D}im(\mathcal{F}, S)$ as a close relative to the so-called *eluder dimension* (Russo and Van Roy, 2013), as well as to the more recent *online decoupling coefficient*, as defined in Dann et al. (2021).

We recall that for a function space $\mathcal{F} = \{f : \mathcal{X} \to [0, 1]\}$, and scale $\epsilon > 0$, the eluder dimension $dim_E(\mathcal{F}, \epsilon)$ can be defined as follows. A point $\mathbf{x} \in \mathcal{X}$ is ϵ-dependent on points $\{\mathbf{x}_1, \ldots, \mathbf{x}_t\} \subseteq \mathcal{X}$ with respect to $\mathcal{F}$ if any pair of functions $f, g \in \mathcal{F}$ satisfying $\sum_{i=1}^{t}(f(\mathbf{x}_i) - g(\mathbf{x}_i))^2 \le \epsilon^2$ also satisfies $(f(\mathbf{x}) - g(\mathbf{x}))^2 \le \epsilon^2$. Point $\mathbf{x}$ is ϵ-independent of $\{\mathbf{x}_1, \ldots, \mathbf{x}_t\}$ with respect to $\mathcal{F}$ if $\mathbf{x}$ is not ϵ-dependent on $\{\mathbf{x}_1, \ldots, \mathbf{x}_t\}$. Then $dim_E(\mathcal{F}, \epsilon)$ is the length of the longest sequence of elements in $\mathcal{X}$ such that, for some $\epsilon' \ge \epsilon$, every element in the sequence is ϵ'-independent from its predecessors in the sequence.

A number of papers provide bounds on the eluder dimension for various function classes. The original bounds on tabular, linear, and generalized linear functions are found in Russo and Van Roy (2013); Osband and Van Roy (2014). When the function class lies in a reproducing kernel Hilbert space (rkhs), Huang et al. (2021) show that the eluder dimension is equivalent to the notion of information gain (see also Remark 6 below). Further, Dong et al. (2021) prove an exponential lower bound on the eluder dimension for ReLU networks.

As for the relationships between $\mathcal{D}im_T(\mathcal{F})$ and $dim_E(\mathcal{F}, \epsilon)$, we show in Appendix B, Lemma 23 therein, the following bound:[3]

$$\mathcal{D}im_T(\mathcal{F}) \le \bar{C} \inf_{\eta \ge 1} \Big(dim_E(\mathcal{F}, T^{-\frac{1}{2\eta}}) \log^2(T) + T^{1-\frac{1}{\eta}} \Big) \ , \qquad (4)$$

where $\bar{C} > 0$ is a universal constant.

[3] For a weighted version of our complexity measure, Ye et al. (2023) obtained a bound corresponding to $\eta = 1$, which does not enable a trade off on η.

For instance, if $\mathit{dim}_E(\mathcal{F},\epsilon) = O((1/\epsilon)^\beta)$, for some $\beta \geq 0$, then selecting η in (4) that equalizes the two terms in the right-hand side (and disregarding the $\log^2(T)$ factor) gives a bound of the form

$$\mathcal{D}\mathit{im}_T(\mathcal{F}) = \tilde{O}(T^{\frac{\beta}{2+\beta}})\ ,$$

which is achieved for eluder dimension $\mathit{dim}_E(\mathcal{F},\epsilon)$ at scale $\epsilon = (1/T)^{\frac{1}{2+\beta}}$. As a consequence, our bounds will be non-vacuous if the function space $\mathcal{F}$ satisfies

$$\mathit{dim}_E(\mathcal{F},\epsilon) = O((1/\epsilon)^\beta)$$

for some $\beta < \infty$. We stress that this is not possible to achieve if we rely on the typical eluder dimension scale of $\epsilon = 1/T$ found in the non-linear bandit literature in the absence of gap assumptions (e.g., Russo and Van Roy, 2013) or the scale $\epsilon = \Delta$ in the presence of gap $\Delta > 0$ among actions/policies (e.g., Foster et al., 2020).

In the case where $\mathcal{F}$ is a class of linear functions on a d-dimensional space, it is known that $\mathit{dim}_E(\mathcal{F},\epsilon) = O(d\log(1/\epsilon))$, and we can set in (4) $\eta = 1$ to obtain $\mathcal{D}\mathit{im}_T(\mathcal{F}) = O(d\log^3 T)$, which is worse than the usual $d\log T$ bound one is expecting. As a matter of fact, in the linear (or generalized linear) case a more direct argument is possible (see Remark 6 below) giving the desired $\mathcal{D}\mathit{im}_T(\mathcal{F}) = O(d\log T)$ bound.

The algorithm for the non-linear case is given as Algorithm 2. The pseudocode is similar to the one in Algorithm 1, where we replace the diversity measure $\|\mathbf{x}\|_{A^{-1}_{\ell,t-1}}$ by $D(\mathbf{x},\mathcal{Q}_\ell)$, the margin $\langle \mathbf{w}_\ell, \mathbf{x}\rangle$ by $\widehat{f}_\ell(\mathbf{x}) - 1/2$, and define the confidence levels through the quantity $K_T(\delta,\ell,\gamma)$ satisfying

$$K_T(\delta,\ell,\gamma) = \sqrt{8\log\frac{2\ell(\ell+1)\left(\frac{eT}{V\gamma}\right)^V}{\delta} + 2\gamma T\left(8 + \sqrt{8\log\frac{8\ell(\ell+1)T^2}{\delta}}\right)} + 1\ ,$$

where V is the *VC-subgraph dimension* (or Pollard's *pseudo-dimension*) of $\mathcal{F}$, i.e. the VC-dimension of the 0/1-valued class of functions

$$\{(\mathbf{x},\theta) \to \mathbb{1}\{f(\mathbf{x}) \leq \theta\}\ :\ f \in \mathcal{F}, \theta \in [0,1]\}\ .$$

The above expression for $K_T(\delta,\ell,\gamma)$ is itself the result of upper bounding in the parametric case ($V < \infty$) the covering numbers $N_\infty(\mathcal{F},\mathcal{Q}_\ell,\gamma)$ (with $\gamma = 1/T$) occurring in the analysis. See, in particular, Lemma 26 in Appendix C. This upper bound via the VC-subgraph dimension is not strictly necessary, and is only aimed at making the function $K_T(\delta,\ell,1/T)$ occurring in the algorithm's pseudo-code easier to interpret.

5.1 Analysis

The following is the main result of this paper.

Theorem 3 *Let V be the VC-subgraph dimension of $\mathcal{F}$, and $T \geq \max\{\mathcal{D}\mathit{im}(\mathcal{F},\mathcal{P}), V\}$. Then with probability at least $1-\delta$ over the random draw of $(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_T,y_T) \sim \mathcal{D}$ the* *excess risk $\mathcal{L}(\widehat{f}) - \mathcal{L}(f_\star)$, the label complexity $N_T(\mathcal{P})$, and the number of stages L generated by Algorithm 2 with $\gamma = 1/T$ are simultaneously upper bounded as follows:*

$$\begin{aligned}
&\mathcal{L}(\widehat{f}) - \mathcal{L}(f_\star) \\
&\qquad \le \bar{C}\, C(\delta, T, \epsilon_0)\left(\max\left\{ \left(\frac{\mathcal{D}im(\mathcal{F},\mathcal{P})}{T}\right)^{\frac{\alpha+1}{\alpha+2}},\ \frac{\mathcal{D}im(\mathcal{F},\mathcal{P})}{T\epsilon_0} \right\}\right) \\
&\qquad\qquad + \bar{C}\left(\frac{\log\left(\frac{\log T}{\delta}\right) + V\log T}{T}\right)\ ,
\end{aligned}$$

$$N_T(\mathcal{P}) \le \bar{C}\, C(\delta, T, \epsilon_0)\left(\max\left\{ \mathcal{D}im(\mathcal{F},\mathcal{P})^{\frac{\alpha}{\alpha+2}} T^{\frac{2}{\alpha+2}},\ \frac{\mathcal{D}im(\mathcal{F},\mathcal{P})}{\epsilon_0^2} \right\} + \log^2\left(\frac{\log T}{\delta}\right)\right)\ ,$$

$$L \le \bar{C}\left(\max\left\{ \frac{\log\left(\frac{T}{\mathcal{D}im(\mathcal{F},\mathcal{P})}\right)}{\alpha+2},\ \log\left(\frac{1}{\epsilon_0}\right) \right\} + \log\left(\frac{\log T}{\delta}\right)\right)\ ,$$

for an absolute constant $\bar{C}$ and

$$C(\delta, T, \epsilon_0) = V\log^2\left(\frac{T}{\delta}\right)\left(1 + \log^2\left(\frac{1}{\epsilon_0}\right)\right)\ .$$

A few remarks are in order.

Remark 4 *In Algorithm 2 the stopping condition*

$$\mathcal{D}im(\mathcal{F},\mathcal{P})/2^{-\ell+1} > 2^{-\ell+1}|\mathcal{P}_\ell|$$

involves $\mathcal{D}im(\mathcal{F},\mathcal{P})$, which may be hard to compute in practice. In its stead, we can introduce a tuning parameter ω and replace $\mathcal{D}im(\mathcal{F},\mathcal{P})$ therein by ω. As a consequence, the bounds in Theorem 3 (ignoring constant factors and lower order terms) become

$$\mathcal{L}(\widehat{f}) - \mathcal{L}(f_\star) \le \max\left\{ \left(\frac{\mathcal{D}im(\mathcal{F},\mathcal{P})}{T}\right)^{\frac{\alpha+1}{\alpha+2}} \left(\frac{\mathcal{D}im(\mathcal{F},\mathcal{P})}{\omega}\right)^{\frac{1}{\alpha+2}} + \left(\frac{\omega}{T}\right)^{\frac{\alpha+1}{\alpha+2}},\ \frac{\mathcal{D}im(\mathcal{F},\mathcal{P})}{T\epsilon_0} \right\}\ ,$$

$$N_T(\mathcal{P}) \le \max\left\{ \mathcal{D}im(\mathcal{F},\mathcal{P})^{\frac{\alpha}{\alpha+2}}\, T^{\frac{2}{\alpha+2}} \left(\frac{\mathcal{D}im(\mathcal{F},\mathcal{P})}{\omega}\right)^{\frac{2}{\alpha+2}},\ \frac{\mathcal{D}im(\mathcal{F},\mathcal{P})}{\epsilon_0^2} \right\}\ ,$$

$$L \le \max\left\{ \frac{\log\left(\frac{T}{\omega}\right)}{\alpha+2},\ \log\left(\frac{1}{\epsilon_0}\right) \right\}\ .$$

Note that the excess risk is optimized when $\omega = \mathcal{D}im(\mathcal{F},\mathcal{P})$.

Remark 5 *For the sake of illustration, consider the case where $\mathcal{D}im(\mathcal{F},\mathcal{P}) \le \mathcal{D}im_T(\mathcal{F}) = O(T^\zeta)$, for some $\zeta \in [0,1)$, as is the case when $dim_E(\mathcal{F},\epsilon) = O((1/\epsilon)^\beta)$ for some $\beta < \infty$. In this case, one can easily see that with high probability (and disregarding constants and log factors)*

$$\mathcal{L}(\widehat{f}) - \mathcal{L}(f_\star) \approx \frac{1}{(N_T(\mathcal{P}))^{\frac{(1-\zeta)(1+\alpha)}{2+\zeta\alpha}}}\ ,$$

Algorithm 2: Pool-based batch active learning algorithm for general non-linear models.

```
Input: Confidence level $\delta \in (0, 1]$, pool of instances $\mathcal{P} \subseteq \mathbb{R}^d$ of size $|\mathcal{P}| = T$, scale parameter $\gamma$
Initialize: $\mathcal{P}_0 = \mathcal{P}$
for $\ell = 1, 2, \dots,$
    Initialize within stage $\ell$:
        • $\epsilon_\ell = 2^{-\ell}/K_T(\delta, \ell, \gamma)$
        • $t = 0, \;\; \mathcal{Q}_\ell = \emptyset$
    while $\mathcal{P}_{\ell-1} \backslash \mathcal{Q}_\ell \neq \emptyset$ and $\max_{\mathbf{x} \in \mathcal{P}_{\ell-1} \backslash \mathcal{Q}_\ell} D(\mathbf{x}, \mathcal{Q}_\ell) > \epsilon_\ell$
        • $t = t + 1$
        • Pick $\mathbf{x}_{\ell,t} \in \operatorname{argmax}_{\mathbf{x} \in \mathcal{P}_{\ell-1} \backslash \mathcal{Q}_\ell} D(\mathbf{x}, \mathcal{Q}_\ell)$
        • Update $\;\; \mathcal{Q}_\ell = \mathcal{Q}_\ell \cup \{\mathbf{x}_{\ell,t}\}$
    Set $T_\ell = t$, the number of queries made in stage $\ell$
    if $\mathcal{Q}_\ell \neq \emptyset$
        • Query the labels $y_{\ell,1}, \dots, y_{\ell,T_\ell}$ associated with the unlabeled data in $\mathcal{Q}_\ell$, and compute
              $$\widehat{f}_\ell = \operatorname{argmin}_{f \in \mathcal{F}} \sum_{t=1}^{T_\ell} \left( \frac{1 + y_{\ell,t}}{2} - f(\mathbf{x}_{\ell,t}) \right)^2$$
        • Set $\;\mathcal{C}_\ell = \{\mathbf{x} \in \mathcal{P}_{\ell-1} \backslash \mathcal{Q}_\ell : |\widehat{f}_\ell(\mathbf{x}) - 1/2| > 2^{-\ell}\}$
        • Compute pseudo-labels on each $\mathbf{x} \in \mathcal{C}_\ell$ as $\hat{y} = \operatorname{sgn}(\widehat{f}_\ell(\mathbf{x}) - 1/2)$
    else
        $\widehat{f}_\ell = 1/2$ (random guess), $\mathcal{C}_\ell = \emptyset$
    Set $\mathcal{P}_\ell = \mathcal{P}_{\ell-1} \backslash (\mathcal{C}_\ell \cup \mathcal{Q}_\ell)$
    if $\mathcal{D}im(\mathcal{F}, \mathcal{P})/2^{-\ell+1} > 2^{-\ell+1}|\mathcal{P}_\ell|$
        • $L = \ell$
        • Exit the for-loop ($L$ is the total number of stages)
Predict labels in pool $\mathcal{P}$:
    • Find empirical 0–1 loss minimizer $\widehat{f}$ on $\cup_{\ell=1}^{L} \mathcal{C}_\ell$ via the generated pseudo-labels $\hat{y}$
    • Predict on each $\mathbf{x} \in (\cup_{\ell=1}^{L} \mathcal{Q}_\ell) \cup \mathcal{P}_L$ through $\operatorname{sgn}(\widehat{f}(\mathbf{x}) - 1/2)$
```

which are similar to the bounds obtained in Wang et al. (2021) in the streaming setting under assumptions that are similar in spirit.

Remark 6 *One may be wondering whether it is generally possible to directly bound $\mathcal{D}im_T(\mathcal{F})$ in relevant cases. The only relevant cases we are currently aware of are: (i) the linear and generalized linear cases we considered in Section 4 of this paper, and Section 5 of Gentile et al. (2022b), where it is easy to show that $\mathcal{D}im_T(\mathcal{F}) = O(d \log T)$; (ii) The case where $\mathcal{F}$ is made up of bounded functions in a rkhs. In this case, $\mathcal{D}im(\mathcal{F}, S) \leq \log |I + G(S)|$, and $\mathcal{D}im_T(\mathcal{F}) \leq \max_{S \subseteq \mathcal{X}\,:\,|S|=T} \log |I + G(S)|$, where $G(S)$ is the kernel Gram matrix build out of the data in pool S. Hence, $\mathcal{D}im_T(\mathcal{F})$ depends on the decay of the eigenvalues of the underlying kernel function. In this case, since $\mathcal{F}$ may not be a VC-class, we also need to resort to known results for bounding covering numbers of function classes within a rkhs (e.g., Zhou, 2002).*

The argument for the linear case is particularly simple. Consider an arbitrary ordering $\langle \mathbf{x}_1, \ldots, \mathbf{x}_T \rangle$ of S, and define

$$
\begin{aligned}
D^2(\mathbf{x}_t, \langle \mathbf{x}_1, \ldots, \mathbf{x}_{t-1} \rangle) &= \sup_{\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^d\,:\,||\mathbf{w}_1||, ||\mathbf{w}_2|| \leq 1} \frac{(\langle \mathbf{w}_1, \mathbf{x}_t \rangle - \langle \mathbf{w}_2, \mathbf{x}_t \rangle)^2}{\sum_{i=1}^{t-1} (\langle \mathbf{w}_1, \mathbf{x}_i \rangle - \langle \mathbf{w}_2, \mathbf{x}_i \rangle)^2 + 1} \\
&= \sup_{\mathbf{w} \in \mathbb{R}^d\,:\,||\mathbf{w}||=2} \frac{(\langle \mathbf{w}, \mathbf{x}_t \rangle)^2}{\sum_{i=1}^{t-1} (\langle \mathbf{w}, \mathbf{x}_i \rangle)^2 + 1} \\
&= \sup_{\mathbf{w} \in \mathbb{R}^d\,:\,||\mathbf{w}||=2} \frac{\mathbf{w}^\top \mathbf{x}_t \mathbf{x}_t^\top \mathbf{w}}{\mathbf{w}^\top A_{t-1} \mathbf{w}} \,,
\end{aligned}
$$

where $A_{t-1} = \sum_{i=1}^{t-1} \mathbf{x}_i \mathbf{x}_i^\top + \frac{1}{2} I$. Since $\mathbf{w}^\top \mathbf{x}_t \mathbf{x}_t^\top \mathbf{w} \leq (\mathbf{w}^\top A_{t-1} \mathbf{w})(\mathbf{x}_t^\top A_{t-1}^{-1} \mathbf{x}_t)$, the above gives

$$
D^2(\mathbf{x}_t, \langle \mathbf{x}_1, \ldots, \mathbf{x}_{t-1} \rangle) \leq \mathbf{x}_t^\top A_{t-1}^{-1} \mathbf{x}_t
$$

and

$$
\mathcal{D}im(\mathcal{F}, S) = \sum_{t=1}^{T} D^2(\mathbf{x}_t, \langle \mathbf{x}_1, \ldots, \mathbf{x}_{t-1} \rangle) = O(d \log T)
$$

by known inequalities (e.g., Azoury and Warmuth, 2001; Cesa-Bianchi et al., 2005; Abbasi-Yadkori et al., 2011). A very similar argument can be carried out for the rkhs case by switching to dual variables, leading to the claimed bound $\mathcal{D}im(\mathcal{F}, S) \leq \log |I + G(S)|$. See, e.g., Cesa-Bianchi et al. (2005), Section 3.2 therein.

Incidentally, the above also shows that the algorithm for the general non-linear case (Algorithm 2) essentially recovers the one for the linear case (Algorithm 1) up to logarithmic factors. To see this, observe that the quantity $K_T(\delta, \ell, \gamma)$ at scale $\gamma = 1/T$ defining ϵ_ℓ in Algorithm 2 is similar to the log factor at the denominator in the expression for ϵ_ℓ in Algorithm 1. More importantly, as argued above, in the linear case we have $D(\mathbf{x}, Q_\ell) \approx ||\mathbf{x}||_{A_{\ell,t}^{-1}}$, and $\mathcal{D}im(\mathcal{F}, S) = \tilde{O}(d)$, which are the corresponding quantities Algorithm 1 relies upon.

Remark 7 *A constant batch size version of Algorithm 2 can also be devised, and the associated properties spelled out. The details are similar to those in Section 4.2 for the linear* *case, but the interplay with the batch size B now also depends on the behavior of $\mathcal{D}im(\mathcal{F}, \mathcal{P})$ or $\mathcal{D}im_T(\mathcal{F})$, as a function of $T = |\mathcal{P}|$. The details are easy extensions of the arguments we carried out for the linear case (comments surrounding Corollary 2), and are therefore omitted.*

6. Conclusions and Open Questions

We have described and analyzed novel batch active learning algorithms in the pool-based setting that in relevant cases achieve minimax rates of convergence of their excess risk as a function of the number of queried labels. The minimax nature of our results is retained also when the batch size B is allowed to scale polynomially ($B \leq T^{\beta}$, for $\beta \leq 1$) with the size T of the training set, the allowed exponent β depending on the actual level of noise in the data. The algorithms have a number of re-training rounds which is at worst logarithmic, and is able to automatically adapt to the noise level.

Our algorithms generate pseudo-labels by restricting to exponentially small regions of the margin space.

The main content of this paper is the extension of the above results to the general non-linear case through a notion of dimension $\mathcal{D}im(\mathcal{F}, S)$ that takes the adaptive nature of active learning sampling into account. The resulting algorithm is a generalization of the one for the linear case, while the corresponding analysis greatly generalizes the linear case analysis, where we replace the greedy version of G-optimal design with a design guided by the diversity measure induced by the function space at hand. We related $\mathcal{D}im(\mathcal{F}, S)$ to the eluder dimension $dim_E(\mathcal{F}, \epsilon)$ of $\mathcal{F}$ at an appropriate scale ϵ, and pointed out that any function space $\mathcal{F}$ such that $dim_E(\mathcal{F}, \epsilon)$ is polynomial in $1/\epsilon$ is actually learnable in our realizable scenario.

It would be nice to see whether it is possible to replace $\mathcal{D}im_T(\mathcal{F})$ by some substantially smaller notion of dimension, in both the algorithm and the analysis. For instance, it might be possible to leverage the fact that our algorithm is selecting points $\mathbf{x}_t$ by maximizing $D(\mathbf{x}, \mathcal{Q}_\ell)$, so that the sequence $\mathbf{x}_1, \mathbf{x}_2, \ldots$ is not completely arbitrary. This would be similar in spirit to the arguments by Brukhim et al. (2023), where the authors introduce a notion of function complexity, called *dissimilarity measure*, that is able to exploit the specific structure of optimistic bandit algorithms, leading to bounds that are in some cases substantially sharper than those based on the eluder dimension.

Acknowledgements

We thank the anonymous reviewers for their insightful comments that helped improve both the content and the presentation of this paper. We thank the action editor for their careful handling of our submission. Tong Zhang is partially supported by an NSF IIS grant No. 2416897.

Appendix A. Proofs for Section 4

Consider Algorithm 1, and denote by T_ℓ the length of stage ℓ.

We denote for any $\epsilon > 0$,

$$\mathcal{T}_\epsilon = \{\mathbf{x} \in \mathcal{P} \,:\, |\langle \mathbf{w}^*, \mathbf{x}\rangle| \leq \epsilon\} \ .$$

Recall that in Algorithm 1 variable L counts the total number of stages (a random quantity), while the size of the original pool $|\mathcal{P}|$ is denoted by T.

We first show that on the confident sets, that is, on sets $\mathcal{C}_\ell$ where pseudo-labels are generated, the learner has with high probability no regret. Before giving our key lemma, it will be useful to define the events

$$\mathcal{E}_\ell = \left\{ \max_{\mathbf{x} \in \mathcal{P}_{\ell-1} \setminus \mathcal{Q}_\ell} |\langle \mathbf{w}_\ell - \mathbf{w}^*, \mathbf{x}\rangle| \leq 2^{-\ell} \right\} \ ,$$

for $\ell = 1, \ldots, L$.

Lemma 8 *For any positive L,*

$$\mathbb{P}\left(\bigcap_{\ell=1}^{L} \mathcal{E}_\ell \right) > 1 - \delta \ .$$

Proof We assume $\mathcal{P}_{\ell-1} \setminus \mathcal{Q}_\ell$ is not empty (it could be empty only in the final stage L). We follow the material contained in Chapters 20 and 21 of the book of Lattimore and Szepesvari (2020). Let $\xi_{\ell,t} = y_{\ell,t} - \langle \mathbf{w}^*, \mathbf{x}_{\ell,t}\rangle$ and notice that $\xi_{\ell,t}$ are independent 1-sub-Gaussian random variables conditioned on $\mathcal{P}_{\ell-1}$. Also, observe that, conditioned on past stages $1, \ldots, \ell - 1$, we are in a *fixed design* scenario, where the $\mathbf{x}_{\ell,t}$ are chosen without knowledge of the corresponding labels $y_{\ell,t}$. We can write, for any $\mathbf{x} \in \mathcal{P}_{\ell-1}$,

$$\begin{aligned}
\langle \mathbf{w}_\ell - \mathbf{w}^*, \mathbf{x}\rangle &= \langle A_{\ell,T_\ell}^{-1} \left(\sum_{t=1}^{T_\ell} y_{\ell,t} \mathbf{x}_{\ell,t} \right) - \mathbf{w}^*, \mathbf{x}\rangle \\
&= \langle A_{\ell,T_\ell}^{-1} \left(\sum_{t=1}^{T_\ell} \mathbf{x}_{\ell,t} \langle \mathbf{w}^*, \mathbf{x}_{\ell,t}\rangle + \xi_{\ell,t} \mathbf{x}_{\ell,t} \right) - \mathbf{w}^*, \mathbf{x}\rangle \\
&= \langle A_{\ell,T_\ell}^{-1} (A_{\ell,T_\ell} - I)\mathbf{w}^* + A_{\ell,T_\ell}^{-1} \left(\sum_{t=1}^{T_\ell} \xi_{\ell,t} \mathbf{x}_{\ell,t} \right) - \mathbf{w}^*, \mathbf{x}\rangle \\
&= -\langle \mathbf{w}^*, \mathbf{x}\rangle_{A_{\ell,T_\ell}^{-1}} + \sum_{t=1}^{T_\ell} \langle \mathbf{x}_{\ell,t}, \mathbf{x}\rangle_{A_{\ell,T_\ell}^{-1}} \xi_{\ell,t} \ .
\end{aligned}$$

Since $\{\xi_{\ell,t}\}_{t=1}^{T_\ell}$ are 1-sub-Gaussian and independent conditioned on $\{\mathbf{x}_{\ell,t}\}$, the variance term $\sum_{t=1}^{T_\ell} \langle \mathbf{x}_{\ell,t}, \mathbf{x}\rangle_{A_{\ell,T_\ell}^{-1}} \xi_{\ell,t}$ is $\sqrt{\sum_{t=1}^{T_\ell} \langle \mathbf{x}_{\ell,t}, \mathbf{x}\rangle_{A_{\ell,T_\ell}^{-1}}^2}$-sub-Gaussian. We apply Lemma 29

$$\mathbb{P}\left(\Big| \sum_{t=1}^{T_\ell} \langle \mathbf{x}_{\ell,t}, \mathbf{x}\rangle_{A_{\ell,T_\ell}^{-1}} \xi_{\ell,t} \Big| \geq \sqrt{2 \sum_{t=1}^{T_\ell} \langle \mathbf{x}_{\ell,t}, \mathbf{x}\rangle_{A_{\ell,T_\ell}^{-1}}^2 \log \frac{2\ell(\ell+1)T}{\delta}} \right) \leq \frac{\delta}{\ell(\ell+1)T} \ .$$

Now observe that

$$\sum_{t=1}^{T_\ell} \langle \mathbf{x}_{\ell,t}, \mathbf{x} \rangle^2_{A^{-1}_{\ell,T_\ell}} = \|\mathbf{x}\|^2_{A^{-1}_{\ell,T_\ell}} - \|A^{-1}_{\ell,T_\ell}\mathbf{x}\|^2 \le \|\mathbf{x}\|^2_{A^{-1}_{\ell,T_\ell}} .$$

We plug back into the previous inequality to obtain

$$\mathbb{P}\left(\Big| \sum_{t=1}^{T_\ell} \langle \mathbf{x}_{\ell,t}, \mathbf{x} \rangle_{A^{-1}_{\ell,T_\ell}} \xi_{\ell,t} \Big| \ge \sqrt{2\|\mathbf{x}\|^2_{A^{-1}_{\ell,T_\ell}} \log \frac{2\ell(\ell+1)T}{\delta}} \right) \le \frac{\delta}{\ell(\ell+1)T} .$$

Using a union bound, we get with probability at least $1 - \frac{\delta}{\ell(\ell+1)}$,

$$\Big| \sum_{t=1}^{T_\ell} \langle \mathbf{x}_{\ell,t}, \mathbf{x} \rangle_{A^{-1}_{\ell,T_\ell}} \xi_{\ell,t} \Big| \le \sqrt{2\|\mathbf{x}\|^2_{A^{-1}_{\ell,T_\ell}} \log \frac{2\ell(\ell+1)T}{\delta}} ,$$

holds uniformly for all $\mathbf{x} \in \mathcal{P}_{\ell-1}$. For the bias term $\langle \mathbf{w}^*, \mathbf{x} \rangle_{A^{-1}_{\ell,T_\ell}}$, notice that $A_{\ell,T_\ell} \succeq I$ implies

$$|\langle \mathbf{w}^*, \mathbf{x} \rangle_{A^{-1}_{\ell,T_\ell}}| \le \|\mathbf{x}\|_{A^{-1}_{\ell,T_\ell}} \|\mathbf{w}^*\|_{A^{-1}_{\ell,T_\ell}} \le \|\mathbf{x}\|_{A^{-1}_{\ell,T_\ell}} .$$

Hence with probability at least $1 - \frac{\delta}{\ell(\ell+1)}$,

$$|\langle \mathbf{w}_\ell - \mathbf{w}^*, \mathbf{x} \rangle| \le \left(\sqrt{2 \log \frac{2\ell(\ell+1)T}{\delta}} + 1 \right) \|\mathbf{x}\|_{A^{-1}_{\ell,T_\ell}} ,$$

holds uniformly for all $\mathbf{x} \in \mathcal{P}_{\ell-1}$.

Notice that by the selection criterion in Algorithm 1, $\max_{\mathbf{x} \in \mathcal{P}_{\ell-1} \setminus \mathcal{Q}_\ell} \|\mathbf{x}\|_{A^{-1}_{\ell,T_\ell}} \le \epsilon^2_\ell$. As a consequence, with probability at least $1 - \frac{\delta}{\ell(\ell+1)}$,

$$\max_{\mathbf{x} \in \mathcal{P}_{\ell-1} \setminus \mathcal{Q}_\ell} |\langle \mathbf{w}_\ell - \mathbf{w}^*, \mathbf{x} \rangle| \le \left(\sqrt{2 \log \frac{2\ell(\ell+1)T}{\delta}} + 1 \right) \epsilon_\ell .$$

Recalling the definition of ϵ_ℓ in Algorithm 1 and using an union bound over ℓ, we get the desired result. ■

As a simple consequence, we have the following lemma.

Lemma 9 *Assume $\bigcap_{\ell=1}^L \mathcal{E}_\ell$ holds. Then Algorithm 1 generates pseudo-labels such that, on all points $\mathbf{x} \in \cup_{\ell=1}^L \mathcal{C}_\ell$,* $\mathrm{sgn}(\langle \mathbf{w}_\ell, \mathbf{x} \rangle) = \mathrm{sgn}(\langle \mathbf{w}^*, \mathbf{x} \rangle)$.

Proof Simply observe that if $\mathbf{x} \in \cup_{\ell=1}^L \mathcal{C}_\ell$ is such that $\mathrm{sgn}(\langle \mathbf{w}_\ell, \mathbf{x} \rangle) = 1$ then $\langle \mathbf{w}_\ell, \mathbf{x} \rangle > 2^{-\ell}$, which implies $\langle \mathbf{w}^*, \mathbf{x} \rangle > 0$ by the assumption that $\mathcal{E}_\ell$ holds. Similarly, $\mathrm{sgn}(\langle \mathbf{w}_\ell, \mathbf{x} \rangle) = -1$ implies $\langle \mathbf{w}^*, \mathbf{x} \rangle < 0$. ■

Lemma 10 *The length T_ℓ of stage ℓ in Algorithm 1 is (deterministically) upper bounded as*

$$T_\ell \le \frac{8d}{\epsilon_\ell^2} \log\left(\frac{1}{\epsilon_\ell}\right) \ .$$

Proof Since in stage ℓ the algorithm terminates at T_ℓ, any round $t < T_\ell$ is such that

$$||\mathbf{x}_{\ell,t+1}||^2_{A^{-1}_{\ell,t}} > \epsilon_\ell^2 \ .$$

We denote $|\cdot|$ as the determinant of the matrix at argument and have the known identity

$$\begin{aligned} |A_{\ell,t+1}| &= |A_{\ell,t} + \mathbf{x}_{\ell,t+1}\mathbf{x}_{\ell,t+1}^\top| \\ &= |A_{\ell,t}| \cdot |I + A_{\ell,t}^{-1}\mathbf{x}_{\ell,t+1}\mathbf{x}_{\ell,t+1}^\top| \\ &= (1 + ||\mathbf{x}_{\ell,t+1}||^2_{A^{-1}_{\ell,t}})|A_{\ell,t}| \\ &\le 2|A_{\ell,t}| \ , \end{aligned}$$

where the third equality holds since $I + A_{\ell,t}^{-1/2}\mathbf{x}_{\ell,t+1}\mathbf{x}_{\ell,t+1}^\top A_{\ell,t}^{-1/2}$ has $d-1$ eigenvalues 1 and one eigenvalue $1 + ||\mathbf{x}_{\ell,t+1}||^2_{A^{-1}_{\ell,t}}$.

Combining the above equality with the fact that $\log(1+x) \ge \frac{x}{1+x} \ge \frac{x}{2}$ for $0 \le x \le 1$, we get

$$||\mathbf{x}_{\ell,t+1}||^2_{A^{-1}_{\ell,t}} = \frac{|A_{\ell,t+1}|}{|A_{\ell,t}|} - 1 \le 2(\log|A_{\ell,t+1}| - \log|A_{\ell,t}|) \ .$$

Therefore,

$$2(\log|A_{\ell,t+1}| - \log|A_{\ell,t}|) > \epsilon_\ell^2 \ .$$

Summing over $t = 0, \ldots, T_\ell - 1$ yields,

$$2 \ \log\frac{|A_{\ell,T_\ell}|}{|A_{\ell,0}|} \ge \epsilon_\ell^2 T_\ell \ .$$

Now, $A_{\ell,0} = I$, so that $|A_{\ell,0}| = 1$, and

$$\log|A_{\ell,T_\ell}| \le \log\left(\mathrm{trace}(A_{\ell,T_\ell})/d\right)^d \le d\log\left(1 + \frac{T_\ell}{d}\right) \ ,$$

yields

$$\frac{T_\ell}{d} \le \frac{2}{\epsilon_\ell^2} \log\left(1 + \frac{T_\ell}{d}\right) \ .$$

Let $G(x) = \frac{x}{\log(1+x)}$, and notice that $G(x)$ is increasing for $x > 0$. We have

$$G\left(\frac{T_\ell}{d}\right) \le \frac{2}{\epsilon_\ell^2} < G\left(\frac{4}{\epsilon_\ell^2}\log\frac{1}{\epsilon_\ell^2}\right) \ ,$$

where the second inequality holds since $\epsilon_\ell \le \epsilon_1 < \frac{1}{4}$.

As a consequence,

$$T_\ell \le \frac{8d}{\epsilon_\ell^2} \log\left(\frac{1}{\epsilon_\ell}\right) ,$$

as claimed. ■

The proof then proceeds by bounding two relevant quantities associated with the behavior of Algorithm 1: the *label complexity*

$$N_T(\mathcal{P}) = \sum_{\ell=1}^{L} |\mathcal{Q}_\ell| ,$$

and the *weighted cumulative regret* over pool $\mathcal{P}$ of size T, defined as

$$R_T(\mathcal{P}) = \sum_{\mathbf{x}\in\mathcal{P}} \mathbb{1}\{\mathrm{sgn}\langle\widehat{\mathbf{w}},\mathbf{x}\rangle \ne \mathrm{sgn}\langle\mathbf{w}^*,\mathbf{x}\rangle\}|\langle\mathbf{w}^*,\mathbf{x}\rangle| .$$

We will first present intermediate bounds on $R_T(\mathcal{P})$ and $N_T(\mathcal{P})$ as a function of L, and then rely on the properties of the noise (hence the randomness on $\mathcal{P}$) to complete the proofs.

To simplify the math display we denote

$$K_T(\delta,\ell) = \sqrt{2\log\frac{2\ell(\ell+1)T}{\delta}} + 1 ,$$

so that $\epsilon_\ell = \frac{1}{2^\ell K_T(\delta,\ell)}$.

Lemma 10 immediately delivers the following bound on $N_T(\mathcal{P})$:

Theorem 11 *For any pool realization $\mathcal{P}$, the label complexity $N_T(\mathcal{P})$ of Algorithm 1 operating on a pool $\mathcal{P}$ of size T is bounded deterministically as*

$$N_T(\mathcal{P}) \le \frac{32}{3} d \log\left(2^L K_T(\delta,L)\right) K_T^2(\delta,L)4^L$$

Proof By definition

$$\begin{aligned} N_T(\mathcal{P}) = \sum_{\ell=1}^{L} T_\ell &\le \sum_{\ell=1}^{L} \frac{8d}{\epsilon_\ell^2} \log\left(\frac{1}{\epsilon_\ell}\right) \\ &\le 8d\log\left(\frac{1}{\epsilon_L}\right) K_T^2(\delta,L) \sum_{\ell=1}^{L} 4^\ell \\ &\le \frac{8}{3} d \log\left(2^L K_T(\delta,L)\right) K_T^2(\delta,L)4^{L+1} , \end{aligned}$$

where the second inequality follows from the fact that both $\frac{1}{\epsilon_\ell}$ and $K_T(\delta,\ell)$ increase with ℓ, and the last inequality follows from $\sum_{\ell=1}^{L} 4^\ell < \frac{4}{3}4^L$. ■

As for the regret $R_T(\mathcal{P})$, we have the following high probability result.

Theorem 12 *For any pool realization $\mathcal{P}$, the weighted cumulative regret $R_T(\mathcal{P})$ of Algorithm 1 operating on a pool $\mathcal{P}$ of size T is bounded as*

$$R_T(\mathcal{P}) \le 64d \log\left(2^L K_T(\delta, L)\right) K_T^2(\delta, L) 2^L + d\, 2^{L-1} \ ,$$

assuming $\bigcap_{\ell=1}^L \mathcal{E}_\ell$ holds.

Proof We decompose the pool $\mathcal{P}$ as the union of following disjoint sets

$$\mathcal{P} = \left(\cup_{l=1}^L \mathcal{C}_\ell\right) \cup \left(\cup_{l=1}^L \mathcal{Q}_\ell\right) \cup \mathcal{P}_L$$

and, correspondingly, the weighted cumulative regret $R_T(\mathcal{P})$ as the sum of the three components

$$R_T(\mathcal{P}) = R(\cup_{l=1}^L \mathcal{C}_\ell) + R(\cup_{l=1}^L \mathcal{Q}_\ell) + R(\mathcal{P}_L) \ .$$

Assume $\bigcap_{\ell=1}^L \mathcal{E}_\ell$ holds. First, notice that on $\mathcal{C}_\ell$,

$$\mathrm{sgn}\langle \widehat{\mathbf{w}}, \mathbf{x}\rangle = \mathrm{sgn}\langle \mathbf{w}_\ell, \mathbf{x}\rangle = \mathrm{sgn}\langle \mathbf{w}^*, \mathbf{x}\rangle$$

under the assumption that $\mathcal{E}_\ell$ holds, thus points in $\cup_{\ell=1}^L \mathcal{C}_\ell$ do not contribute weighted regret for $\widehat{\mathbf{w}}$, i.e.,

$$R(\cup_{l=1}^L \mathcal{C}_\ell) = 0 \ .$$

Next, on $\mathcal{P}_L$, we have $|\langle \mathbf{w}_L, \mathbf{x}\rangle| \le 2^{-L}$. Combining this with the assumption that $\mathcal{E}_L$ holds, we get $|\langle \mathbf{w}^*, \mathbf{x}\rangle| \le 2^{-L+1}$, which implies that the weighted cumulative regret on $\mathcal{P}_L$ is bounded as

$$R(\mathcal{P}_L) \le 2^{-L+1}|\mathcal{P}_L| < d\, 2^{L-1} \ ,$$

the second inequality deriving from the stopping condition defining L in Algorithm 1.

Finally, on the queried points $\cup_{l=1}^L \mathcal{Q}_\ell$, it is unclear whether $\mathrm{sgn}\langle \widehat{\mathbf{w}}, \mathbf{x}\rangle = \mathrm{sgn}\langle \mathbf{w}^*, \mathbf{x}\rangle$ or not, so we bound the weighted cumulative regret contribution of each data item $\mathbf{x}$ therein by $|\langle \mathbf{w}^*, \mathbf{x}\rangle|$. Now, by construction, $\mathbf{x} \in \mathcal{Q}_\ell \subset \mathcal{P}_{\ell-1}$, so that $|\langle \mathbf{w}_{\ell-1}, \mathbf{x}\rangle| \le 2^{-\ell+1}$ which, combined with the assumption that $\mathcal{E}_{\ell-1}$ holds, yields $|\langle \mathbf{w}^*, \mathbf{x}\rangle| \le 2^{-\ell+2}$. Since $|\mathcal{Q}_\ell| = T_\ell$, we have

$$R(\cup_{\ell=1}^L \mathcal{Q}_\ell) \le 4 \sum_{\ell=1}^L T_\ell\, 2^{-\ell}$$

and Lemma 10 allows us to write

$$R(\cup_{l=1}^L \mathcal{Q}_\ell) \le 32d \sum_{l=1}^L \frac{2^{-\ell}}{\epsilon_\ell^2} \log\left(\frac{1}{\epsilon_\ell}\right) \le 64d \log\left(2^L K_T(\delta, L)\right) K_T^2(\delta, L) 2^L \ ,$$

the last inequality following from a reasoning similar to the one that lead us to Theorem 11. ■

Given any pool realization $\mathcal{P}$, both the label complexity and weighted regret are bounded by a function of L. Adding the ingredient of the low noise condition (2) helps us leverage the randomness in $\mathcal{P}$ and further bound from above the number of stages L.

Specifically, assume the low noise condition (2) holds for $f^*(\mathbf{x}) = \frac{1+\langle \mathbf{w}^*, \mathbf{x} \rangle}{2}$, for some unknown exponent $\alpha \geq 0$ and unknown constant $\epsilon_0 \in (0, 1]$. Using a multiplicative Chernoff bound, it is easy to see that for any fixed ϵ_*, with probability at least $1 - \delta$,

$$|\mathcal{T}_{\epsilon_*}| \leq \frac{3}{2} \left(T\epsilon_*^\alpha + \log(1/\delta) \right) \ ,$$

the probability being over the random draw of the initial pool $\mathcal{P}$. Now, since ϵ_L is itself a random variable (since so is L), we need to resort to a covering argument. For any positive number M, consider the following set of fixed ϵ values

$$\mathcal{K}_M = \left\{ \frac{\epsilon_0}{2^{i/\alpha}} : i = 0, \ldots, M \right\} \ .$$

Then with probability at least $1 - \delta$,

$$|\mathcal{T}_\epsilon| \leq \frac{3}{2} \left(T\epsilon^\alpha + \log\left(\frac{M}{\delta}\right) \right) \ ,$$

holds simultaneously over $\epsilon \in \mathcal{K}_M$. Set $M = \log_2 T$ and assume ϵ is the smallest value in $\mathcal{K}_M$ that is bigger than or equal to ϵ_*. If ϵ is not the smallest value in $\mathcal{K}_M$, then by construction we have $\epsilon_*^\alpha \leq \epsilon^\alpha < 2\epsilon_*^\alpha$ so that, for all $\epsilon_* > \frac{\epsilon_0}{2^{M/\alpha}}$,

$$|\mathcal{T}_{\epsilon_*}| \leq |\mathcal{T}_\epsilon| \leq \frac{3}{2} \left(T\epsilon^\alpha + \log\left(\frac{M}{\delta}\right) \right) < 3 \left(T\epsilon_*^\alpha + \log\left(\frac{M}{\delta}\right) \right) \ . \tag{5}$$

On the other hand if $\epsilon_* \leq \frac{\epsilon_0}{2^{M/\alpha}}$ we can write

$$|\mathcal{T}_{\epsilon_*}| \leq \left| \mathcal{T}_{\frac{\epsilon_0}{2^{M/\alpha}}} \right| \leq \frac{3}{2} \left(\frac{T\epsilon_0^\alpha}{2^M} + \log\left(\frac{M}{\delta}\right) \right) \leq \frac{3}{2} \left(1 + \log\left(\frac{M}{\delta}\right) \right) < 3 \log\left(\frac{M}{\delta}\right) \ ,$$

making Eq. (5) hold in this case as well.

We define the event

$$\bar{\mathcal{E}} = \bigcap_{\epsilon_* \in (0, \epsilon_0]} \left\{ |T_{\epsilon_*}| < 3 \left(T\epsilon_*^\alpha + \log\left(\frac{M}{\delta}\right) \right) \right\} \ .$$

Then

$$\mathbb{P}\left(\bar{\mathcal{E}}\right) \geq 1 - \delta \ , \tag{6}$$

for $M = \log_2 T$.

We set ϵ^* to be the unique solution of the equation[4]

$$d/\epsilon_* = 3 \left(T\epsilon_*^{\alpha+1} + \epsilon_* \log\left(\frac{M}{\delta}\right) \right) \ . \tag{7}$$

Eq. (5) will be applied, in particular, to the margin value 2^{-L+2} when $2^{-L+2} \leq \epsilon_0$.

Armed with Eqs. (5) and (7) with $M = \log_2 T$, we prove a lemma that upper bounds the number of stages L.

[4] We need to further assume $T > \frac{2}{3}d$ so as to make sure the solution exists.

Lemma 13 *Let ϵ_* be defined through (7), with $T > \frac{2}{3}d$. Assume both $\bar{\mathcal{E}}$ and $\bigcap_{\ell=1}^{L} \mathcal{E}_\ell$ hold. Then the number of stages L of Algorithm 1 is upper bounded as*

$$
\begin{aligned}
L &\le \max\left(\log_2\left(\frac{1}{\epsilon_*}\right),\ \log_2\left(\frac{1}{\epsilon_0}\right)\right) + 2 \\
&\le \max\left(\log_2\left[\left(\frac{3T}{d}\right)^{\frac{1}{\alpha+2}} + 3\left(\frac{1}{d}\right)^{\frac{\alpha+1}{\alpha+2}}\left(\frac{1}{3T}\right)^{\frac{1}{\alpha+2}}\log(\frac{\log_2 T}{\delta})\right],\ \log_2\left(\frac{1}{\epsilon_0}\right)\right) + 2 \\
&= \max\left(O\left(\frac{1}{\alpha+2}\log\left(\frac{T}{d}\right) + \log\left(\frac{\log T}{\delta}\right)\right),\ \log\left(\frac{4}{\epsilon_0}\right)\right)\ .
\end{aligned}
$$

Here the O-notation only omits absolute constants.

Proof If at stage $L-1$ the algorithm has not stopped, then we must have

$$
d/2^{-L+2} \le 2^{-L+2}|\mathcal{P}_{L-1}|\ .
$$

Notice that if $\mathbf{x} \in \mathcal{P}_{L-1}$ then $|\langle \mathbf{w}_{L-1}, \mathbf{x}\rangle| \le 2^{-L+1}$. Combining it with the assumption that $\mathcal{E}_{L-1}$ holds, we have $|\langle \mathbf{w}^*, \mathbf{x}\rangle| \le 2^{-L+2}$ which implies $|\mathcal{P}_{L-1}| \le |\mathcal{T}_{2^{-L+2}}|$.

We split the analysis into two cases. On one hand, when $2^{-L+2} > \epsilon_0$, this condition gives us directly

$$
L \le \log_2(\frac{1}{\epsilon_0}) + 2\ .
$$

On the other hand if $2^{-L+2} \le \epsilon_0$, then given $\bar{\mathcal{E}}$ holds, $|\mathcal{T}_{2^{-L+2}}|$ is upper bounded as

$$
|\mathcal{T}_{2^{-L+2}}| \le 3\left(T\, 2^{(-L+2)\alpha} + \log\left(\frac{M}{\delta}\right)\right)\ ,
$$

with $M = \log_2 T$. Plugging into the first display results in

$$
d/2^{-L+2} \le 3\left(T 2^{(-L+2)(\alpha+1)} + 2^{-L+2}\log(\frac{M}{\delta})\right)\ ,
$$

which resembles (7) with 2^{-L+2} here playing the role of ϵ^* therein. Then, from the definition of ϵ^* in (7) we immediately obtain $2^{-L+2} \ge \epsilon_*$, thus $L \le \log_2(\frac{1}{\epsilon_*}) + 2$. Moreover, from (7) we see that $d/\epsilon_* \ge 3T\epsilon_*^{\alpha+1}$, which is equivalent to $\epsilon_* \le (\frac{d}{3T})^{\frac{1}{\alpha+2}}$. Replacing this upper bound on ϵ^* back into the right-hand side of (7) and dividing by d yields

$$
\frac{1}{\epsilon_*} \le \left(\frac{3T}{d}\right)^{\frac{1}{\alpha+2}} + 3\left(\frac{1}{d}\right)^{\frac{\alpha+1}{\alpha+2}}\left(\frac{1}{3T}\right)^{\frac{1}{\alpha+2}}\log(\frac{M}{\delta})\ ,
$$

which gives the claimed upper bound on L through $L \le \log_2(\frac{1}{\epsilon_*}) + 2$. ■

Corollary 14 *Let $T > d$. Then with probability at least $1-2\delta$ over the random draw of $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T) \sim \mathcal{D}$ the label complexity $N_T(\mathcal{P})$ and the weighted cumulative regret*

$R_T(\mathcal{P})$ *of Algorithm 1 simultaneously satisfy the following:*

$$N_T(\mathcal{P}) = \log^2\left(\frac{T}{\delta}\right)\left(1+\log^2\left(\frac{1}{\epsilon_0}\right)\right) O\left(\max\left\{d^{\frac{\alpha}{\alpha+2}}T^{\frac{2}{\alpha+2}},\ \frac{d}{\epsilon_0^2}\right\} + \log^2\left(\frac{\log T}{\delta}\right)\right)$$
$$R_T(\mathcal{P}) = \log^2\left(\frac{T}{\delta}\right)\left(1+\log^2\left(\frac{1}{\epsilon_0}\right)\right) O\left(\max\left\{d^{\frac{\alpha+1}{\alpha+2}}T^{\frac{1}{\alpha+2}},\ \frac{d}{\epsilon_0}\right\} + \log\left(\frac{\log T}{\delta}\right)\right) \ .$$

where the O-notation only omits absolute constants .

Proof Assume both $\bar{\mathcal{E}}$ and $\bigcap_{\ell=1}^{L}\mathcal{E}_\ell$ hold. Recalling the definition of $K_T(\delta, L)$, we have

$$K_T(\delta, L) = O\left(\sqrt{\log\left(\frac{T}{\delta}\right) + \log L}\right) = O\left(\sqrt{\log\left(\frac{T}{\delta}\right) + L}\right) \ .$$

Similar to lemma 13, we split the analysis into two cases depending on whether or not 2^{-L+2} is bigger than ϵ_0. If $2^{-L+2} \le \epsilon_0$, we have

$$L \le \log_2\left[\left(\frac{3T}{d}\right)^{\frac{1}{\alpha+2}} + 3\left(\frac{1}{d}\right)^{\frac{\alpha+1}{\alpha+2}}\left(\frac{1}{3T}\right)^{\frac{1}{\alpha+2}}\log(\frac{\log_2 T}{\delta})\right] \ ,$$

therefore,

$$2^L = O\left(\left(\frac{T}{d}\right)^{\frac{1}{\alpha+2}} + \left(\frac{1}{d}\right)^{\frac{\alpha+1}{\alpha+2}}\left(\frac{1}{T}\right)^{\frac{1}{\alpha+2}}\log\left(\frac{\log T}{\delta}\right)\right) \ .$$

Plugging the above bounds into Theorem 11 gives

$$\begin{aligned} N_T(\mathcal{P}) &= O\left(d\left(L + \log K_T^2(\delta, L)\right)K_T^2(\delta, L)4^L\right) \\ &= O\left(\left(L + K_T^2(\delta, L)\right)K_T^2(\delta, L)\left(d^{\frac{\alpha}{\alpha+2}}T^{\frac{2}{\alpha+2}} + \log^2\left(\frac{\log T}{\delta}\right)\right)\right) \\ &= O\left(\left(L + \log\left(\frac{T}{\delta}\right)\right)^2\left(d^{\frac{\alpha}{\alpha+2}}T^{\frac{2}{\alpha+2}} + \log^2\left(\frac{\log T}{\delta}\right)\right)\right) \\ &= \log^2\left(\frac{T}{\delta}\right) O\left(d^{\frac{\alpha}{\alpha+2}}T^{\frac{2}{\alpha+2}} + \log^2\left(\frac{\log T}{\delta}\right)\right) \ . \end{aligned}$$

Similarly applying them to Theorem 12,

$$\begin{aligned} R_T(\mathcal{P}) &= O\left(d\left(L + \log K_T^2(\delta, L)\right)K_T^2(\delta, L)2^L\right) \\ &= O\left(\left(L + K_T^2(\delta, L)\right)K_T^2(\delta, L)\left(d^{\frac{\alpha+1}{\alpha+2}}T^{\frac{1}{\alpha+2}} + \log\left(\frac{\log T}{\delta}\right)\right)\right) \\ &= O\left(\left(L + \log\left(\frac{T}{\delta}\right)\right)^2\left(d^{\frac{\alpha+1}{\alpha+2}}T^{\frac{1}{\alpha+2}} + \log\left(\frac{\log T}{\delta}\right)\right)\right) \\ &= \log^2\left(\frac{T}{\delta}\right) O\left(d^{\frac{\alpha+1}{\alpha+2}}T^{\frac{1}{\alpha+2}} + \log\left(\frac{\log T}{\delta}\right)\right) \ , \end{aligned}$$

where in the second equality we used the assumption that $d < T$.

If $2^{-L+2} > \epsilon_0$, then $2^L \le \frac{4}{\epsilon_0}$. Plugging these bounds into Theorem 11 and Theorem 12 gives

$$N_T(\mathcal{P}) = O\left(\log^2\left(\frac{T}{\delta\epsilon_0}\right)\frac{d}{\epsilon_0^2}\right) = \log^2\left(\frac{T}{\delta}\right)\left(1+\log^2\left(\frac{1}{\epsilon_0}\right)\right)O\left(\frac{d}{\epsilon_0^2}\right)$$
$$R_T(\mathcal{P}) = O\left(\log^2\left(\frac{T}{\delta\epsilon_0}\right)\frac{d}{\epsilon_0}\right) = \log^2\left(\frac{T}{\delta}\right)\left(1+\log^2\left(\frac{1}{\epsilon_0}\right)\right)O\left(\frac{d}{\epsilon_0}\right) .$$

Lastly, (6) and lemma 8 together yield

$$\mathbb{P}\left(\bar{\mathcal{E}}\bigcap\left(\bigcap_{\ell=1}^{L}\mathcal{E}_\ell\right)\right) \ge 1-2\delta ,$$

which concludes the proof. ■

We now turn the bound on the weighted cumulative regret $R_T(\mathcal{P})$ in the previous corollary into a bound on the excess risk. We can write

$$\begin{aligned}\mathcal{L}(\widehat{\mathbf{w}}) - \mathcal{L}(\mathbf{w}^*) &= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\Big[\,\mathbb{1}\{y \neq \mathrm{sgn}(\langle\widehat{\mathbf{w}},\mathbf{x}\rangle)\} - \mathbb{1}\{y \neq \mathrm{sgn}(\langle\mathbf{w}^*,\mathbf{x}\rangle)\}\Big]\\ &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathcal{X}}\Big[\mathbb{E}_{y\sim\mathcal{D}_{\mathcal{Y}|\mathcal{X}}}\big[\,\mathbb{1}\{y \neq \mathrm{sgn}(\langle\widehat{\mathbf{w}},\mathbf{x}\rangle)\} - \mathbb{1}\{y \neq \mathrm{sgn}(\langle\mathbf{w}^*,\mathbf{x}\rangle)\}\big]\Big]\\ &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathcal{X}}\Big[\,\mathbb{1}\{\mathrm{sgn}(\langle\widehat{\mathbf{w}},\mathbf{x}\rangle) \neq \mathrm{sgn}(\langle\mathbf{w}^*,\mathbf{x}\rangle)\}\,|\langle\mathbf{w}^*,\mathbf{x}\rangle|\Big] ,\end{aligned}$$

where $\widehat{\mathbf{w}}$ is the hypothesis returned by Algorithm 1. Now, simply observe that

$$\mathbb{1}\{\mathrm{sgn}(\langle\widehat{\mathbf{w}},\mathbf{x}\rangle) \neq \mathrm{sgn}(\langle\mathbf{w}^*,\mathbf{x}\rangle)\}\,|\langle\mathbf{w}^*,\mathbf{x}\rangle|$$

has the same form as the function $\phi(\widehat{\mathbf{w}},\mathbf{x})$ in Appendix C on which the uniform convergence result of Theorem 28 applies, with $\widehat{\epsilon}(\delta)$ therein replaced by the bound on $R_T(\mathcal{P})$ borrowed from Corollary 14. This allows us to conclude that with probability at least $1-\delta$

$$\mathcal{L}(\widehat{\mathbf{w}}) - \mathcal{L}(\mathbf{w}^*) = \log^2\left(\frac{T}{\delta}\right)\left(1+\log^2\left(\frac{1}{\epsilon_0}\right)\right)O\left(\max\left\{\left(\frac{d}{T}\right)^{\frac{\alpha+1}{\alpha+2}}, \frac{d}{T\epsilon_0}\right\} + \frac{\log\left(\frac{\log T}{\delta}\right)}{T}\right) ,$$

as claimed in Theorem 1 in the main body of the paper.

Appendix B. Proofs for Section 5

The proof has the very same structure as the one in Section A. Hence we only emphasize the main differences.

The starting point of the analysis is Proposition 2 in Russo and Van Roy (2013) which, with our assumptions and notation, reads as follows.

Lemma 15 *Let $\widehat{f}_\ell$ be the estimator computed by Algorithm 2 at the end of stage ℓ, and $\mathcal{Q}_\ell = \{\mathbf{x}_{\ell,1}, \ldots, \mathbf{x}_{\ell,T_\ell}\}$ be the queried points in stage ℓ. Then with probability at least $1 - \frac{\delta}{\ell(\ell+1)}$ we have*

$$\sum_{t=1}^{T_\ell} (f_\star(\mathbf{x}_{\ell,t}) - \widehat{f}_\ell(\mathbf{x}_{\ell,t}))^2 \le 8 \log \frac{2\ell(\ell+1)\, N_\infty(\mathcal{F}, \mathcal{Q}_\ell, \gamma)}{\delta} + 2\gamma T_\ell \left(8 + \sqrt{8 \log \frac{8\ell(\ell+1)T_\ell^2}{\delta}} \right),$$

where $N_\infty(\mathcal{F}, \mathcal{Q}_\ell, \gamma)$ is the size of a γ-cover of function class $\mathcal{F}$ with respect to the infinity norm.

Note that from Lemma 26 we can further bound $N_\infty(\mathcal{F}, \mathcal{Q}_\ell, \gamma)$ by $\left(\frac{eT}{V\gamma}\right)^V$, where V is the VC-subgraph dimension of $\mathcal{F}$. To simplify the math display we denote

$$K_T(\delta, \ell, \gamma) = \sqrt{8 \log \frac{2\ell(\ell+1)\,\left(\frac{eT}{V\gamma}\right)^V}{\delta} + 2\gamma T \left(8 + \sqrt{8 \log \frac{8\ell(\ell+1)T^2}{\delta}} \right) + 1}\ ,$$

so that $\epsilon_\ell = \frac{1}{2^\ell K_T(\delta,\ell,\gamma)}$.

Moreover, we define, for any $\epsilon > 0$,

$$\mathcal{T}_\epsilon = \{\mathbf{x} \in \mathcal{P} \,:\, |f_*(\mathbf{x}) - 1/2| \le \epsilon\}\ .$$

Recall that in Algorithm 2 variable L counts the total number of stages, while T denotes the size of the original pool $\mathcal{P}$.

Similar to Appendix A, we define the events

$$\mathcal{E}_\ell = \left\{ \max_{\mathbf{x} \in \mathcal{P}_{\ell-1} \setminus \mathcal{Q}_\ell} |f_*(\mathbf{x}) - \widehat{f}_\ell(\mathbf{x})| \le 2^{-\ell} \right\}\ ,$$

for $\ell = 1, \ldots, L$.

Lemma 16 *For any positive L, we have*

$$\mathbb{P}\left(\bigcap_{\ell=1}^{L} \mathcal{E}_\ell \right) > 1 - \delta\ .$$

Proof We assume $\mathcal{P}_{\ell-1} \setminus \mathcal{Q}_\ell$ is not empty (it could be empty only in the final stage L). Then for any ℓ and $\mathbf{x} \in \mathcal{P}_{\ell-1} \setminus \mathcal{Q}_\ell$ we can write

$$\begin{aligned}
(f_\star(\mathbf{x}) \quad & - \quad \widehat{f}_\ell(\mathbf{x}))^2 \\
&= \frac{(f_\star(\mathbf{x}) - \widehat{f}_\ell(\mathbf{x}))^2}{\sum_{t=1}^{T_\ell} (f_\star(\mathbf{x}_{\ell,t}) - \widehat{f}_\ell(\mathbf{x}_{\ell,t}))^2 + 1} \left(\sum_{t=1}^{T_\ell} (f_\star(\mathbf{x}_{\ell,t}) - \widehat{f}_\ell(\mathbf{x}_{\ell,t}))^2 + 1 \right) \\
&\le \sup_{f,g \in \mathcal{F}} \frac{(f(\mathbf{x}) - g(\mathbf{x}))^2}{\sum_{t=1}^{T_\ell} (f(\mathbf{x}_{\ell,t}) - g(\mathbf{x}_{\ell,t}))^2 + 1} \left(\sum_{t=1}^{T_\ell} (f_\star(\mathbf{x}_{\ell,t}) - \widehat{f}_\ell(\mathbf{x}_{\ell,t}))^2 + 1 \right) \\
&= D^2(\mathbf{x}; \mathcal{Q}_\ell) \left(\sum_{t=1}^{T_\ell} (f_\star(\mathbf{x}_{\ell,t}) - \widehat{f}_\ell(\mathbf{x}_{\ell,t}))^2 + 1 \right) \\
&\le \epsilon_\ell^2 K_T^2\,(\delta, \ell, \gamma)\ ,
\end{aligned}$$

holds with probability at least $1 - \frac{\delta}{\ell(\ell+1)}$. This is due to Lemma 15 and the fact that $\max\limits_{\mathbf{x}\in\mathcal{P}_{\ell-1}\setminus\mathcal{Q}_\ell} D(\mathbf{x}, \mathcal{Q}_\ell) \le \epsilon_\ell$.

Using the definition of ϵ_ℓ and union bound we get the desired result. ■

Similar to Appendix A, this yields the following consequence.

Lemma 17 *Assume $\bigcap_{\ell=1}^L \mathcal{E}_\ell$ holds. Then Algorithm 2 generates pseudo-labels such that, on all points $\mathbf{x} \in \cup_{\ell=1}^L \mathcal{C}_\ell$, $\mathrm{sgn}(\widehat{f}_\ell(\mathbf{x}) - 1/2) = \mathrm{sgn}(f_*(\mathbf{x}) - 1/2)$.*

Lemma 18 *The length T_ℓ of stage ℓ in Algorithm 2 is (deterministically) upper bounded as*

$$T_\ell \le \frac{\mathcal{D}im(\mathcal{F}, \mathcal{Q}_\ell)}{\epsilon_\ell^2} \le \frac{\mathcal{D}im(\mathcal{F}, \mathcal{P})}{\epsilon_\ell^2} \ .$$

Proof Since in stage ℓ the algorithm terminates at T_ℓ, any round $t < T_\ell$ is such that

$$D^2\left(\mathbf{x}_{\ell,t+1}; \langle \mathbf{x}_{\ell,1}, \ldots, \mathbf{x}_{\ell,t}\rangle\right) > \epsilon_\ell^2 \ .$$

Summing this inequality up we get

$$\mathcal{D}im(\mathcal{F}, \mathcal{Q}_\ell) \ge \epsilon_\ell^2 T_\ell \ ,$$

as a consequence

$$T_\ell \le \frac{\mathcal{D}im(\mathcal{F}, \mathcal{Q}_\ell)}{\epsilon_\ell^2} \le \frac{\mathcal{D}im(\mathcal{F}, \mathcal{P})}{\epsilon_\ell^2} \ ,$$

as claimed. ■

We then bound the *label complexity*

$$N_T(\mathcal{P}) = \sum_{\ell=1}^L |\mathcal{Q}_\ell| \ ,$$

and the *weighted cumulative regret* over pool $\mathcal{P}$ of size T,

$$R_T(\mathcal{P}) = \sum_{\mathbf{x}\in\mathcal{P}} \mathbb{1}\Big\{\mathrm{sgn}(\widehat{f}(\mathbf{x}) - 1/2) \ne \mathrm{sgn}(f_*(\mathbf{x}) - 1/2)\Big\} |f_*(\mathbf{x}) - 1/2| \ .$$

We will first present intermediate bounds on $R_T(\mathcal{P})$ and $N_T(\mathcal{P})$ as a function of L, and then rely on the properties of the noise (hence the randomness on $\mathcal{P}$) to complete the proofs.

Lemma 18 immediately delivers the following bound on $N_T(\mathcal{P})$:

Theorem 19 *For any pool realization $\mathcal{P}$, the label complexity $N_T(\mathcal{P})$ of Algorithm 2 operating on a pool $\mathcal{P}$ of size T is bounded deterministically as*

$$N_T(\mathcal{P}) \le \frac{4^{L+1}}{3} K_T^2(\delta, L, \gamma) \mathcal{D}im(\mathcal{F}, \mathcal{P}) \ .$$

As for the regret $R_T(\mathcal{P})$, we have the following high probability result.

Theorem 20 *For any pool realization $\mathcal{P}$, the weighted cumulative regret $R_T(\mathcal{P})$ of Algorithm 1 operating on a pool $\mathcal{P}$ of size T is bounded as*

$$R_T(\mathcal{P}) \le 2^{L+4} K_T^2(\delta, L, \gamma) \mathcal{D}im(\mathcal{F}, \mathcal{P}) \ ,$$

assuming $\bigcap_{\ell=1}^{L} \mathcal{E}_\ell$ holds.

Proof We decompose the pool $\mathcal{P}$ as the union of following disjoint sets

$$\mathcal{P} = \left(\cup_{l=1}^{L} \mathcal{C}_\ell\right) \cup \left(\cup_{l=1}^{L} \mathcal{Q}_\ell\right) \cup \mathcal{P}_L$$

and, correspondingly, the weighted cumulative regret $R_T(\mathcal{P})$ as the sum of the three components

$$R_T(\mathcal{P}) = R(\cup_{l=1}^{L} \mathcal{C}_\ell) + R(\cup_{l=1}^{L} \mathcal{Q}_\ell) + R(\mathcal{P}_L) \ .$$

Assume $\bigcap_{\ell=1}^{L} \mathcal{E}_\ell$ holds. First, notice that on $\mathcal{C}_\ell$,

$$\mathrm{sgn}(\widehat{f}(\mathbf{x}) - 1/2) = \mathrm{sgn}(\widehat{f}_\ell(\mathbf{x}) - 1/2) = \mathrm{sgn}(f_*(\mathbf{x}) - 1/2) \ ,$$

where the second equality is due to Lemma 17 and the first equality follows from the fact that $\hat{f}$ minimize the 0–1 loss on the generated pseudo-labels, thus incurs no loss since f_* already incurs zero loss. As a consequence points in $\cup_{\ell=1}^{L} \mathcal{C}_\ell$ do not contribute weighted regret for $\widehat{f}$, i.e.,

$$R(\cup_{l=1}^{L} \mathcal{C}_\ell) = 0 \ .$$

Next, on $\mathcal{P}_L$, we have $|\widehat{f}_L(\mathbf{x}) - 1/2| \le 2^{-L}$. Combining this with the assumption that $\mathcal{E}_L$ holds, we get $|f_*(\mathbf{x}) - 1/2| \le 2^{-L+1}$, which implies that the weighted cumulative regret on $\mathcal{P}_L$ is bounded as

$$R(\mathcal{P}_L) \le 2^{-L+1} |\mathcal{P}_L| < \mathcal{D}im(\mathcal{F}, \mathcal{P}) \, 2^{L-1} \ ,$$

the second inequality deriving from the stopping condition defining L in Algorithm 2.

Finally, on the queried points $\cup_{l=1}^{L} \mathcal{Q}_\ell$, it is unclear whether $\mathrm{sgn}(\widehat{f}(\mathbf{x}) - 1/2) = \mathrm{sgn}(f_*(\mathbf{x}) - 1/2)$ or not, so we bound the weighted cumulative regret contribution of each data item $\mathbf{x}$ therein by $|f_*(\mathbf{x}) - 1/2|$. Now, by construction, $\mathbf{x} \in \mathcal{Q}_\ell \subset \mathcal{P}_{\ell-1}$, so that $|f_{\ell-1}(\mathbf{x}) - 1/2| \le 2^{-\ell+1}$ which, combined with the assumption that $\mathcal{E}_{\ell-1}$ holds, yields $|f_*(\mathbf{x}) - 1/2| \le 2^{-\ell+2}$. Since $|\mathcal{Q}_\ell| = T_\ell$, we have

$$R(\cup_{\ell=1}^{L} \mathcal{Q}_\ell) \le 4 \sum_{\ell=1}^{L} T_\ell \, 2^{-\ell}$$

and Lemma 18 allows us to write

$$R(\cup_{l=1}^{L} \mathcal{Q}_\ell) \le 4 \sum_{l=1}^{L} \frac{2^{-\ell} \mathcal{D}im(\mathcal{F}, \mathcal{P})}{\epsilon_\ell^2} \le 2^{L+3} K_T^2(\delta, L, \gamma) \mathcal{D}im(\mathcal{F}, \mathcal{P}) \ ,$$

the last inequality following from a reasoning similar to the one that lead us to Theorem 19. ■

Given any pool realization $\mathcal{P}$, both the label complexity and weighted regret are bounded by a function of L. Adding the ingredient of the low noise condition (2) helps us leverage the randomness in $\mathcal{P}$ and further bound from above the number of stages L.

Specifically, assume the low noise condition (2) holds for $f^*(\mathbf{x})$, for some unknown exponent $\alpha \geq 0$ and unknown constant $\epsilon_0 \in (0, 1]$. We set ϵ^* to be the unique solution of the equation[5]

$$\mathcal{D}im(\mathcal{F}, \mathcal{P})/\epsilon_* = 3\left(T\epsilon_*^{\alpha+1} + \epsilon_* \log\left(\frac{M}{\delta}\right)\right) \ . \tag{8}$$

Eq. (5) holds in this case by low noise condition and will be applied, in particular, to the margin value 2^{-L+2} when $2^{-L+2} \leq \epsilon_0$.

Armed with Eqs. (5) and (8) with $M = \log_2 T$, we prove a lemma that upper bounds the number of stages L. Recall the definition of event $\bar{\mathcal{E}}$:

$$\bar{\mathcal{E}} = \bigcap_{\epsilon_* \in (0,\epsilon_0]} \left\{ |T_{\epsilon_*}| < 3\left(T\epsilon_*^{\alpha} + \log\left(\frac{M}{\delta}\right)\right)\right\} \ .$$

Lemma 21 *Let ϵ_* be defined through (7), with $T > \frac{2}{3}\mathcal{D}im(\mathcal{F}, \mathcal{P})$ and $\gamma = 1/T$. Assume both $\bar{\mathcal{E}}$ and $\bigcap_{\ell=1}^{L} \mathcal{E}_\ell$ hold. Then the number of stages L of Algorithm 2 is upper bounded as*

$$\begin{aligned}
L &\leq \max\left\{\log_2\left(\frac{1}{\epsilon_*}\right),\ \log_2\left(\frac{1}{\epsilon_0}\right)\right\} + 2 \\
&\leq \max\Bigg\{\log_2\left[\left(\frac{3T}{\mathcal{D}im(\mathcal{F}, \mathcal{P})}\right)^{\frac{1}{\alpha+2}} + 3\left(\frac{1}{\mathcal{D}im(\mathcal{F}, \mathcal{P})}\right)^{\frac{\alpha+1}{\alpha+2}}\left(\frac{1}{3T}\right)^{\frac{1}{\alpha+2}} \log(\frac{\log_2 T}{\delta})\right], \\
&\qquad\qquad \log_2\left(\frac{1}{\epsilon_0}\right)\Bigg\} + 2 \\
&= \max\left\{O\left(\frac{1}{\alpha+2}\log\left(\frac{T}{\mathcal{D}im(\mathcal{F}, \mathcal{P})}\right) + \log\left(\frac{\log T}{\delta}\right)\right),\ \log\left(\frac{4}{\epsilon_0}\right)\right\} \ .
\end{aligned}$$

Here the O-notation only omits absolute constants.

Proof If at stage $L - 1$ the algorithm has not stopped, then we must have

$$\mathcal{D}im(\mathcal{F}, \mathcal{P})/2^{-L+2} \leq 2^{-L+2}|\mathcal{P}_{L-1}| \ .$$

Notice that if $\mathbf{x} \in \mathcal{P}_{L-1}$ then $|\widehat{f}_{L-1}(\mathbf{x}) - 1/2| \leq 2^{-L+1}$. Combining it with the assumption that $\mathcal{E}_{L-1}$ holds, we have $|f_*(\mathbf{x}) - 1/2| \leq 2^{-L+2}$ which implies $|\mathcal{P}_{L-1}| \leq |\mathcal{T}_{2^{-L+2}}|$.

The rest of the proof remains unchanged compared to Lemma 13, except that we replace d by $\mathcal{D}im(\mathcal{F}, \mathcal{P})$. ■

[5] We need to further assume $T > \frac{2}{3}\mathcal{D}im(\mathcal{F}, \mathcal{P})$ so as to make sure the solution exists.

Corollary 22 *Let $T > \mathcal{D}im(\mathcal{F},\mathcal{P})$ and $\gamma = 1/T$. Then with probability at least $1-2\delta$ over the random draw of $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T) \sim \mathcal{D}$ the label complexity $N_T(\mathcal{P})$ and the weighted cumulative regret $R_T(\mathcal{P})$ of Algorithm 2 simultaneously satisfy the following:*

$$\begin{aligned} N_T(\mathcal{P}) &= V\log^2\left(\frac{T}{\delta}\right)\left(1+\log^2\left(\frac{1}{\epsilon_0}\right)\right) \\ &\qquad \times O\left(\max\left\{\mathcal{D}im(\mathcal{F},\mathcal{P})^{\frac{\alpha}{\alpha+2}}T^{\frac{2}{\alpha+2}},\ \frac{\mathcal{D}im(\mathcal{F},\mathcal{P})}{\epsilon_0^2}\right\}+\log^2\left(\frac{\log T}{\delta}\right)\right) \\ R_T(\mathcal{P}) &= V\log^2\left(\frac{T}{\delta}\right)\left(1+\log^2\left(\frac{1}{\epsilon_0}\right)\right) \\ &\qquad \times O\left(\max\left\{\mathcal{D}im(\mathcal{F},\mathcal{P})^{\frac{\alpha+1}{\alpha+2}}T^{\frac{1}{\alpha+2}},\ \frac{\mathcal{D}im(\mathcal{F},\mathcal{P})}{\epsilon_0}\right\}+\log\left(\frac{\log T}{\delta}\right)\right)\ , \end{aligned}$$

where the O-notation only omits absolute constants .

Proof Assume both $\bar{\mathcal{E}}$ and $\bigcap_{\ell=1}^{L}\mathcal{E}_\ell$ hold. Recalling the definition of $K_T(\delta, L, \gamma)$, we have

$$\begin{aligned} K_T(\delta,L,\gamma) &= \sqrt{8\log\frac{2\ell(\ell+1)\left(\frac{eT}{V\gamma}\right)^V}{\delta}+2\gamma T\left(8+\sqrt{8\log\frac{8\ell(\ell+1)T^2}{\delta}}\right)+1} \\ &= O\left(\sqrt{\log\left(\frac{\left(\frac{eT}{V\gamma}\right)^V}{\delta}\right)+\gamma T\sqrt{\log L+\log\left(\frac{T}{\delta}\right)}+\log L}\right)\ . \end{aligned}$$

We choose $\gamma = \frac{1}{T}$ and simplify the above display as

$$K_T(\delta, L, \frac{1}{T}) = O\left(\sqrt{V\log\left(\frac{T}{\delta}\right)+\log L}\right) = O\left(\sqrt{V\log\left(\frac{T}{\delta}\right)+L}\right)\ .$$

Similar to Lemma 21, we split the analysis into two cases depending on whether or not 2^{-L+2} is bigger than ϵ_0. If $2^{-L+2} \le \epsilon_0$, we have

$$L \le \log_2\left[\left(\frac{3T}{\mathcal{D}im(\mathcal{F},\mathcal{P})}\right)^{\frac{1}{\alpha+2}}+3\left(\frac{1}{\mathcal{D}im(\mathcal{F},\mathcal{P})}\right)^{\frac{\alpha+1}{\alpha+2}}\left(\frac{1}{3T}\right)^{\frac{1}{\alpha+2}}\log(\frac{\log_2 T}{\delta})\right]\ ,$$

therefore,

$$2^L = O\left(\left(\frac{T}{\mathcal{D}im(\mathcal{F},\mathcal{P})}\right)^{\frac{1}{\alpha+2}}+\left(\frac{1}{\mathcal{D}im(\mathcal{F},\mathcal{P})}\right)^{\frac{\alpha+1}{\alpha+2}}\left(\frac{1}{T}\right)^{\frac{1}{\alpha+2}}\log\left(\frac{\log T}{\delta}\right)\right)\ .$$

Plugging the above bounds into Theorem 19 gives

$$\begin{aligned}
N_T(\mathcal{P}) &= O\left(\mathcal{D}im(\mathcal{F},\mathcal{P})\left(L+\log K_T\left(\delta,L,\frac{1}{T}\right)\right)K_T^2(\delta,L,\frac{1}{T})4^L\right)\\
&= O\left(\left(L+\log K_T\left(\delta,L,\frac{1}{T}\right)\right)K_T^2(\delta,L,\frac{1}{T})\left(\mathcal{D}im(\mathcal{F},\mathcal{P})^{\frac{\alpha}{\alpha+2}}T^{\frac{2}{\alpha+2}}+\log^2\left(\frac{\log T}{\delta}\right)\right)\right)\\
&= O\left(\left(L+V\log\left(\frac{T}{\delta}\right)\right)\left(L+\log\left(\frac{T}{\delta}\right)\right)\left(\mathcal{D}im(\mathcal{F},\mathcal{P})^{\frac{\alpha}{\alpha+2}}T^{\frac{2}{\alpha+2}}+\log^2\left(\frac{\log T}{\delta}\right)\right)\right)\\
&= \left(L+V\log\left(\frac{T}{\delta}\right)\right)\left(L+\log\left(\frac{T}{\delta}\right)\right)O\left(\mathcal{D}im(\mathcal{F},\mathcal{P})^{\frac{\alpha}{\alpha+2}}T^{\frac{2}{\alpha+2}}+\log^2\left(\frac{\log T}{\delta}\right)\right)\ .
\end{aligned}$$

Similarly applying them to Theorem 20,

$$\begin{aligned}
R_T(\mathcal{P}) &= O\left(\mathcal{D}im(\mathcal{F},\mathcal{P})\left(L+\log K_T\left(\delta,L,\frac{1}{T}\right)\right)K_T^2(\delta,L,\frac{1}{T})2^L\right)\\
&= O\left(\left(L+K_T\left(\delta,L,\frac{1}{T}\right)\right)K_T^2(\delta,L,\frac{1}{T})\left(\mathcal{D}im(\mathcal{F},\mathcal{P})^{\frac{\alpha+1}{\alpha+2}}T^{\frac{1}{\alpha+2}}+\log\left(\frac{\log T}{\delta}\right)\right)\right)\\
&= O\left(\left(L+V\log\left(\frac{T}{\delta}\right)\right)\left(L+\log\left(\frac{T}{\delta}\right)\right)\left(\mathcal{D}im(\mathcal{F},\mathcal{P})^{\frac{\alpha+1}{\alpha+2}}T^{\frac{1}{\alpha+2}}+\log\left(\frac{\log T}{\delta}\right)\right)\right)\\
&= V\log^2\left(\frac{T}{\delta}\right)O\left(\mathcal{D}im(\mathcal{F},\mathcal{P})^{\frac{\alpha+1}{\alpha+2}}T^{\frac{1}{\alpha+2}}+\log\left(\frac{\log T}{\delta}\right)\right)\ ,
\end{aligned}$$

where in the second equality we used the assumption that $\mathcal{D}im(\mathcal{F},\mathcal{P}) < T$.

If $2^{-L+2} > \epsilon_0$, then $2^L \le \frac{4}{\epsilon_0}$. Plugging these bounds into Theorem 19 and Theorem 20 gives

$$\begin{aligned}
N_T(\mathcal{P}) =& O\left(V\log^2\left(\frac{T}{\delta\epsilon_0}\right)\frac{\mathcal{D}im(\mathcal{F},\mathcal{P})}{\epsilon_0^2}\right) = V\log^2\left(\frac{T}{\delta}\right)\left(1+\log^2\left(\frac{1}{\epsilon_0}\right)\right)O\left(\frac{\mathcal{D}im(\mathcal{F},\mathcal{P})}{\epsilon_0^2}\right)\\
R_T(\mathcal{P}) =& O\left(V\log^2\left(\frac{T}{\delta\epsilon_0}\right)\frac{\mathcal{D}im(\mathcal{F},\mathcal{P})}{\epsilon_0}\right) = V\log^2\left(\frac{T}{\delta}\right)\left(1+\log^2\left(\frac{1}{\epsilon_0}\right)\right)O\left(\frac{\mathcal{D}im(\mathcal{F},\mathcal{P})}{\epsilon_0}\right)
\end{aligned}$$

Lastly, (6) and lemma 16 together yield

$$\mathbb{P}\left(\bar{\mathcal{E}}\bigcap\left(\bigcap_{\ell=1}^{L}\mathcal{E}_\ell\right)\right) \ge 1-2\delta\ ,$$

which concludes the proof. ■

We now turn the bound on the weighted cumulative regret $R_T(\mathcal{P})$ in the previous corollary into a bound on the excess risk. We can write

$$\begin{aligned}
\mathcal{L}(\widehat{f})-\mathcal{L}(f_\star) &= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\Big[\,\mathbb{1}\Big\{y\neq \mathrm{sgn}(\widehat{f}(\mathbf{x})-1/2)\Big\}-\ \mathbb{1}\{y\neq\mathrm{sgn}(f_\star(\mathbf{x})-1/2)\}\Big]\\
&= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathcal{X}}\Big[\mathbb{E}_{y\sim\mathcal{D}_{\mathcal{Y}|\mathcal{X}}}\Big[\,\mathbb{1}\Big\{y\neq\mathrm{sgn}(\widehat{f}(\mathbf{x})-1/2)\Big\}-\ \mathbb{1}\{y\neq\mathrm{sgn}(f_\star(\mathbf{x})-1/2)\}\Big]\Big]\\
&= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathcal{X}}\Big[\,\mathbb{1}\Big\{\mathrm{sgn}(\widehat{f}(\mathbf{x})-1/2)\neq\mathrm{sgn}(f_\star(\mathbf{x})-1/2)\Big\}\,|f_\star(\mathbf{x})-1/2|\Big]\ ,
\end{aligned}$$

where $\widehat{f}$ is the hypothesis returned by Algorithm 2. Now, simply observe that

$$\mathbb{1}\Big\{\mathrm{sgn}(\widehat{f}(\mathbf{x})-1/2)\neq\mathrm{sgn}(f_\star(\mathbf{x})-1/2)\Big\}\,|f_\star(\mathbf{x})-1/2|$$

has the same form as the function $\phi(\widehat{f},\mathbf{x})$ in Appendix C on which the uniform convergence result of Theorem 28 (with $\langle\widehat{\mathbf{w}},\mathbf{x}\rangle$ replaced by $\widehat{f}(\mathbf{x})-1/2$) applies, with $\widehat{\epsilon}(\delta)$ therein replaced by the bound on $R_T(\mathcal{P})$ borrowed from Corollary 22. This allows us to conclude that, with probability at least $1-\delta$,

$$\begin{aligned}\mathcal{L}(\widehat{f})&-\mathcal{L}(f_\star)\\ &=V\log^2\left(\frac{T}{\delta}\right)\left(1+\log^2\left(\frac{1}{\epsilon_0}\right)\right)O\left(\max\left\{\left(\frac{\mathcal{D}im(\mathcal{F},\mathcal{P})}{T}\right)^{\frac{\alpha+1}{\alpha+2}},\ \frac{\mathcal{D}im(\mathcal{F},\mathcal{P})}{T\epsilon_0}\right\}\right)\\ &\qquad+O\left(\frac{\log\left(\frac{\log T}{\delta}\right)+V\log\left(2eT\right)}{T}\right),\end{aligned}$$

as claimed in Theorem 3 in the main body of the paper.

Lastly we show that our notion of complexity is upper bounded by the *eluder dimension* at an appropriate scale.

Lemma 23 *There exists a positive constant $\bar{C}$ such that, for any $T>0$,*

$$\sup_{S\subset\mathcal{X}\,:\,|S|=T}\mathcal{D}im(\mathcal{F},S)\leq\bar{C}\,\inf_{\eta\geq1}\left(dim_E(\mathcal{F},T^{-\frac{1}{2\eta}})\log^2(T)+T^{1-\frac{1}{\eta}}\right)\ .$$

Proof We only need to prove for any $\eta\geq1$ and a fixed sequence $\langle\mathbf{x}_1,\ldots,\mathbf{x}_T\rangle$,

$$\sum_{t=1}^{T}D^2\Big(\mathbf{x}_t;\langle\mathbf{x}_1,\ldots,\mathbf{x}_{t-1}\rangle\Big)\leq C\left(dim_E(\mathcal{F},T^{-\frac{1}{2\eta}})\log^2(T)+T^{1-\frac{1}{\eta}}\right)\ .$$

Recall that

$$D^2\Big(\mathbf{x}_t;\langle\mathbf{x}_1,\ldots,\mathbf{x}_{t-1}\rangle\Big)=\sup_{f,g\in\mathcal{F}}\frac{(f(\mathbf{x}_t)-g(\mathbf{x}_t))^2}{\sum_{i=1}^{t-1}(f(\mathbf{x}_i)-g(\mathbf{x}_i))^2+1}\ .$$

Without loss of generality we can assume there exist f_t and g_t such that

$$\sup_{f,g\in\mathcal{F}}\frac{(f(\mathbf{x}_t)-g(\mathbf{x}_t))^2}{\sum_{i=1}^{t-1}(f(\mathbf{x}_i)-g(\mathbf{x}_i))^2+1}=\frac{(f_t(\mathbf{x}_t)-g_t(\mathbf{x}_t))^2}{\sum_{i=1}^{t-1}(f_t(\mathbf{x}_i)-g_t(\mathbf{x}_i))^2+1}\ .$$

Given $\eta\geq1$, we split the $[0,1]$ interval into the $\lfloor\log_2(T)\rfloor+1$ intervals

$$[0,2^{-\lfloor\log_2(T)\rfloor/\eta}],\ (2^{-\lfloor\log_2(T)\rfloor/\eta},2^{-(\lfloor\log_2(T)\rfloor-1)/\eta}],\ \ldots,\ (2^{-1/\eta},1]\ .$$

Let

$$U_i=\{\mathbf{x}_t\in S\mid(f_t(\mathbf{x}_t)-g_t(\mathbf{x}_t))^2\in(2^{-i/\eta},2^{-(i-1)/\eta}]\}$$

and

$$U_{\lfloor\log_2(T)\rfloor+1}=\{\mathbf{x}_t\in S\mid(f_t(\mathbf{x}_t)-g_t(\mathbf{x}_t))^2\leq2^{-\lfloor\log_2(T)\rfloor/\eta}\}\ .$$

Then

$$
\begin{aligned}
\sum_{t=1}^{T} D^2\Big(\mathbf{x}_t;\langle \mathbf{x}_1,\ldots,\mathbf{x}_{t-1}\rangle\Big) &= \sum_{i=1}^{\lfloor \log_2(T)\rfloor} \sum_{\mathbf{x}_t\in U_i} \frac{(f_t(\mathbf{x}_t)-g_t(\mathbf{x}_t))^2}{\sum_{i=1}^{t-1}(f_t(\mathbf{x}_i)-g_t(\mathbf{x}_i))^2+1} \\
&\quad + \sum_{\mathbf{x}_t\in U_{\lfloor \log_2(T)\rfloor+1}} \frac{(f_t(\mathbf{x}_t)-g_t(\mathbf{x}_t))^2}{\sum_{i=1}^{t-1}(f_t(\mathbf{x}_i)-g_t(\mathbf{x}_i))^2+1} \\
&\le \sum_{i=1}^{\lfloor \log_2(T)\rfloor} \sum_{\mathbf{x}_t\in U_i} \frac{(f_t(\mathbf{x}_t)-g_t(\mathbf{x}_t))^2}{\sum_{i=1}^{t-1}(f_t(\mathbf{x}_i)-g_t(\mathbf{x}_i))^2+1} + 2^{\frac{1}{\eta}}T^{1-\frac{1}{\eta}}\ .
\end{aligned}
$$

Each U_i we further decompose into $K_i = \lceil |U_i|/dim_E(\mathcal{F}, 2^{-\frac{i}{2\eta}})\rceil + 1$ disjoint sets $\Omega_i^1,\ldots,\Omega_i^{K_i}$ in the following way. Originally all Ω_i^j $(j=1,\ldots,K_i)$ are empty. We sort the $\mathbf{x}_t$'s in U_i according to their index and assign them one by one. Specifically, we assign $\mathbf{x}_t$ to $\Omega_i^{j(t)}$ where $j(t)$ is the smallest index among $1,\ldots,K_i-1$ such that $\mathbf{x}_t$ is $2^{-\frac{i}{2\eta}}$-independent to elements in $\Omega_i^{j(t)}$; if $\mathbf{x}_t$ is $2^{-\frac{i}{2\eta}}$-dependent to $\Omega_i^1,\ldots,\Omega_i^{K_i-1}$, assign it to $\Omega_i^{K_i}$. As a consequence,

$$
\frac{(f_t(\mathbf{x}_t)-g_t(\mathbf{x}_t))^2}{\sum_{i=1}^{t-1}(f_t(\mathbf{x}_i)-g_t(\mathbf{x}_i))^2+1} \le \frac{2^{-\frac{i-1}{\eta}}}{(j(t)-1)2^{-\frac{i}{\eta}}+1} = \frac{2^{\frac{1}{\eta}}}{j(t)-1+2^{\frac{i}{\eta}}}\ ,
$$

where the inequality follows from $\mathbf{x}_t$'s $2^{-\frac{i}{2\eta}}$-dependency on $\Omega_i^1,\ldots,\Omega_i^{j(t)-1}$.

With the help of the above inequality,

$$
\begin{aligned}
\sum_{\mathbf{x}_t\in U_i} \frac{(f_t(\mathbf{x}_t)-g_t(\mathbf{x}_t))^2}{\sum_{i=1}^{t-1}(f_t(\mathbf{x}_i)-g_t(\mathbf{x}_i))^2+1} &= \sum_{j=1}^{K_i-1} \sum_{\mathbf{x}_t\in\Omega_i^j} \frac{(f_t(\mathbf{x}_t)-g_t(\mathbf{x}_t))^2}{\sum_{i=1}^{t-1}(f_t(\mathbf{x}_i)-g_t(\mathbf{x}_i))^2+1} \\
&\quad + \sum_{\mathbf{x}_t\in\Omega_i^{K_i}} \frac{(f_t(\mathbf{x}_t)-g_t(\mathbf{x}_t))^2}{\sum_{i=1}^{t-1}(f_t(\mathbf{x}_i)-g_t(\mathbf{x}_i))^2+1} \\
&\le \sum_{j=1}^{K_i-1} \frac{2^{\frac{1}{\eta}}|\Omega_i^j|}{j-1+2^{\frac{i}{\eta}}} + \frac{2|\Omega_i^{K_i}|}{K_i-1+2^{\frac{i}{\eta}}} \\
&\le 2\,dim_E(\mathcal{F}, 2^{-\frac{i}{2\eta}}) \sum_{j=1}^{K_i-1} \frac{1}{j-1+2^{\frac{i}{\eta}}} + 2\frac{|U_i|}{K_i-1} \\
&\le 2\,dim_E(\mathcal{F}, 2^{-\frac{i}{2\eta}})(\log(K_i)+1)\ ,
\end{aligned}
$$

where in the second inequalty we used the fact that $|\Omega_i^j| \le dim_E(\mathcal{F}, 2^{-\frac{i}{\eta}})$, for $j=1,\ldots,K_i-1$.

Finally combining all the equations and using the fact that $dim_E(\mathcal{F},\cdot)$ is monotone

$$
\begin{aligned}
\sum_{t=1}^{T} D^2\Big(\mathbf{x}_t;\langle \mathbf{x}_1,\ldots,\mathbf{x}_{t-1}\rangle\Big) &\le 2dim_E(\mathcal{F}, T^{-\frac{1}{2\eta}}) \sum_{i=1}^{\lfloor \log_2(T)\rfloor} (\log(K_i)+1) + 2^{\frac{1}{\eta}}T^{1-\frac{1}{\eta}} \\
&\le C\left(dim_E(\mathcal{F}, T^{-\frac{1}{2\eta}})\log^2(T) + T^{1-\frac{1}{\eta}}\right)\ ,
\end{aligned}
$$

as claimed. ■

Appendix C. Ancillary technical results

This section collects ancillary technical results that are used throughout the appendix.

We first recall the following version of the Hoeffding's bound.

Lemma 24 *Let $a_1, \dots, a_T$ be T arbitrary real numbers, and $\{\sigma_1, \dots, \sigma_T\}$ be T i.i.d. Rademacher variables, each taking values ± 1 with equal probability. Then for any $\epsilon \geq 0$*

$$\mathbb{P}\left(\sum_{t=1}^{T} \sigma_t a_t \geq \epsilon\right) \leq \exp\left(-\frac{\epsilon^2}{2\sum_{t=1}^{T} a_t^2}\right) ,$$

where the probability is with respect to $\{\sigma_1, \dots, \sigma_T\}$.

Let us consider the linear case first. Define the function $\phi : \mathbb{R}^d \times \mathcal{P} \to [0,1]$ as

$$\phi(\widehat{\mathbf{w}}, \mathbf{x}) = \mathbb{1}\{\mathrm{sgn}\langle \widehat{\mathbf{w}}, \mathbf{x}\rangle \neq \mathrm{sgn}\langle \mathbf{w}^*, \mathbf{x}\rangle\}\rho(\langle \mathbf{w}^*, \mathbf{x}\rangle) ,$$

where $\rho(\cdot)$ has range in $[0,1]$, and does not depend on $\widehat{\mathbf{w}}$.

We have the following standard covering result, which is a direct consequence of Sauer-Shelah lemma (e.g., Sauer, 1972).

Lemma 25 *Consider any given $S_T = \{\mathbf{x}_1, \dots, \mathbf{x}_T\} \in \mathbb{R}^d$, and let*

$$\Phi(S_T) = \left|\{[\phi(\widehat{\mathbf{w}}, \mathbf{x}_1), \dots, \phi(\widehat{\mathbf{w}}, \mathbf{x}_T)] \,:\, \widehat{\mathbf{w}} \in \mathbb{R}^d\}\right| .$$

We have, when $T \geq d$,

$$\Phi(S_T) \leq \left(\frac{eT}{d}\right)^d .$$

The following result gives a bound on L_∞ covering number for VC subgraph class.

Lemma 26 *Let $f \in \mathcal{F}$ be a $[0,1]$-valued function of $\mathbf{x} \in \mathcal{X}$. Assume the binary function $\mathbb{1}\{z \leq f(\mathbf{x})\}$ defined on $\mathbb{R} \times \mathcal{X}$ has VC-dimension V, i.e. $\mathcal{F}$ has VC-subgraph dimension V. Then given $S_T = \{\mathbf{x}_1, \dots, \mathbf{x}_T\} \subset \mathcal{X}$, with $T \geq V$,*

$$N(\epsilon, \mathcal{F}, L_\infty(S_T)) \leq \left(\frac{eT}{V\epsilon}\right)^V .$$

Proof As $N(\cdot, \mathcal{F}, L_\infty(S_T))$ is a decreasing function of ϵ, we can assume that $1/\epsilon$ is an integer. Consider discretization of $[0,1]$ by $m = 1/\epsilon$ points $U_1, \dots, U_m$ so that $\forall U \in [0,1]$, $\min_i |U - U_i| \leq \epsilon$. Consider mT points $\{(U_j, x_i) : i, j\}$, by Sauer's lemma, the cardinality of $\{(\mathbb{1}\{U_j \leq f(\mathbf{x}_i)\})_{i,j} \,:\, f \in \mathcal{F}\}$ is at most $\left(\frac{eT}{V\epsilon}\right)^V$. Let $f_1, \dots, f_N$ attain all distinctive values of $\{(\mathbb{1}\{U_j \leq f(\mathbf{x}_i)\})_{i,j} \,:\, f \in \mathcal{F}\}$. Given any $f \in \mathcal{F}$, there exists f_k such that $\mathbb{1}\{U_j \leq f(\mathbf{x}_i)\} = \mathbb{1}\{U_j \leq f_k(\mathbf{x}_i)\}$ for all i and j. It is easy to check that this implies that $|f(\mathbf{x}_i) - f_k(\mathbf{x}_i)| \leq \epsilon$ for all $\mathbf{x}_i \in S_T$. ■

The next result follows from a standard symmetrization argument.

Lemma 27 *Let $\mathcal{X} = \mathbb{R}^d$, $S_T = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ be a sample drawn i.i.d. according to $\mathcal{D}_\mathcal{X}$ and $S'_T = (\mathbf{x}'_1, \ldots, \mathbf{x}'_T)$ be another sample drawn according to $\mathcal{D}_\mathcal{X}$, with $T \geq d$. Then with probability at least $1-\delta$*

$$\sum_{t=1}^T \phi(\widehat{\mathbf{w}}, \mathbf{x}'_t) \leq 3 \sum_{t=1}^T \phi(\widehat{\mathbf{w}}, \mathbf{x}_t) + 8 \log(1/\delta) + 8d \log(2eT/d) \ ,$$

uniformly over $\widehat{\mathbf{w}} \in \mathbb{R}^d$.

Proof Let $\{\sigma_1, \ldots, \sigma_T\}$ be independent Rademacher variables as in Lemma 24. We can write, for any $\epsilon \geq 0$,

$$\begin{aligned}
&\mathbb{P}\left(\exists \widehat{\mathbf{w}} \in \mathbb{R}^d : \sum_{t=1}^T \phi(\widehat{\mathbf{w}}, \mathbf{x}'_t) \geq 3 \sum_{t=1}^T \phi(\widehat{\mathbf{w}}, \mathbf{x}_t) + 2\epsilon \right)\\
&= \mathbb{P}\left(\exists \widehat{\mathbf{w}} \in \mathbb{R}^d : \sum_{t=1}^T \Big[\phi(\widehat{\mathbf{w}}, \mathbf{x}'_t) - \phi(\widehat{\mathbf{w}}, \mathbf{x}_t)\Big] \geq \frac{1}{2} \sum_{t=1}^T \Big[\phi(\widehat{\mathbf{w}}, \mathbf{x}_t) + \phi(\widehat{\mathbf{w}}, \mathbf{x}'_t)\Big] + \epsilon \right)\\
&= \mathbb{P}\left(\exists \widehat{\mathbf{w}} \in \mathbb{R}^d : \sum_{t=1}^T \sigma_t \Big[\phi(\widehat{\mathbf{w}}, \mathbf{x}'_t) - \phi(\widehat{\mathbf{w}}, \mathbf{x}_t)\Big] \geq \frac{1}{2} \sum_{t=1}^T \Big[\phi(\widehat{\mathbf{w}}, \mathbf{x}_t) + \phi(\widehat{\mathbf{w}}, \mathbf{x}'_t)\Big] + \epsilon \right)\\
&\leq \mathbb{P}\left(\exists \widehat{\mathbf{w}} \in \mathbb{R}^d : \sum_{t=1}^T \sigma_t \phi(\widehat{\mathbf{w}}, \mathbf{x}'_t) \geq \frac{1}{2} \sum_{t=1}^T \phi(\widehat{\mathbf{w}}, \mathbf{x}'_t) + \frac{\epsilon}{2} \right)\\
&\qquad + \mathbb{P}\left(\exists \widehat{\mathbf{w}} \in \mathbb{R}^d : \sum_{t=1}^T -\sigma_t \phi(\widehat{\mathbf{w}}, \mathbf{x}_t) \geq \frac{1}{2} \sum_{t=1}^T \phi(\widehat{\mathbf{w}}, \mathbf{x}_t) + \frac{\epsilon}{2} \right)\\
&\leq 2 \sup_{S_T} \mathbb{P}\left(\exists \widehat{\mathbf{w}} \in \mathbb{R}^d : \sum_{t=1}^T \sigma_t \phi(\widehat{\mathbf{w}}, \mathbf{x}_t) \geq \frac{1}{2} \sum_{t=1}^T \phi(\widehat{\mathbf{w}}, \mathbf{x}_t) + \frac{1}{2}\epsilon \,\Big|\, S_T \right)\\
&\leq 2 \left(\frac{eT}{d}\right)^d \sup_{S_T, \widehat{\mathbf{w}} \in \mathbb{R}^d} \mathbb{P}\left(\sum_{t=1}^T \sigma_t \phi(\widehat{\mathbf{w}}, \mathbf{x}_t) \geq \frac{1}{2} \sum_{t=1}^T \phi(\widehat{\mathbf{w}}, \mathbf{x}_t) + \frac{1}{2}\epsilon \,\Big|\, S_T \right)\\
&\text{(from the union bound and Lemma 25)}\\
&\leq 2 \left(\frac{eT}{d}\right)^d \sup_{S_T, \widehat{\mathbf{w}} \in \mathbb{R}^d} \exp\left(-\frac{(\sum_{t=1}^T \phi(\widehat{\mathbf{w}}, \mathbf{x}_t) + \epsilon)^2}{8 \sum_{t=1}^T \phi(\widehat{\mathbf{w}}, \mathbf{x}_t)^2}\right)\\
&\text{(from Lemma 24)}\\
&\leq 2 \left(\frac{eT}{d}\right)^d \exp(-\epsilon/4) \ ,
\end{aligned}$$

the last inequality deriving from the fact that, since $\phi(\widehat{\mathbf{w}}, \mathbf{x}_t) \in [0,1]$,

$$\frac{(\sum_{t=1}^T \phi(\widehat{\mathbf{w}}, \mathbf{x}_t) + \epsilon)^2}{\sum_{t=1}^T \phi(\widehat{\mathbf{w}}, \mathbf{x}_t)^2} \geq \frac{(\sum_{t=1}^T \phi(\widehat{\mathbf{w}}, \mathbf{x}_t) + \epsilon)^2}{\sum_{t=1}^T \phi(\widehat{\mathbf{w}}, \mathbf{x}_t)} \geq 2\epsilon \ .$$

Take ϵ such that $\delta = 2\left(\frac{eT}{d}\right)^d \exp(-\epsilon/4)$, to obtain the claimed bound. ■

Theorem 28 *With the same notation and assumptions as in Lemma 27, let $\widehat{\mathbf{w}} \in \mathbb{R}^d$ be a function of S_T such that*

$$\frac{1}{T}\sum_{t=1}^{T}\phi(\widehat{\mathbf{w}},\mathbf{x}_t) \le \widehat{\epsilon}(\delta)$$

holds with probability at least $1-\delta$, for some $\widehat{\epsilon}(\delta) \in [0,1]$. Then with probability at least $1-3\delta$:

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\phi(\widehat{\mathbf{w}},\mathbf{x}) \le 4\widehat{\epsilon}(\delta) + \frac{22\log\left(\frac{1}{\delta}\right) + 11d\log\left(\frac{2eT}{d}\right)}{T} .$$

Proof Use the multiplicative Chernoff bound

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\phi(\widehat{\mathbf{w}},\mathbf{x}) \le \frac{4}{3T}\sum_{t=1}^{T}\phi(\widehat{\mathbf{w}},\mathbf{x}'_t) + \frac{32}{3T}\log(1/\delta) ,$$

and then apply Lemma 27 to further bound the right-hand side. ■

To control noise terms, which are 1-subgaussian random variables, we provide the following lemma which is a direct implication of Chernoff bound.

Lemma 29 *Suppose ξ is a σ-subgaussian random variable, then for any $\delta > 0$,*

$$\mathbb{P}\left(|\xi| \ge \sqrt{2\sigma^2\log(2/\delta)}\right) \le \delta .$$

Let us now consider the nonlinear case analyzed in Section B. Similar to the linear case, define the function $\phi : \mathcal{F}\times\mathcal{P} \to [0,1]$ as

$$\phi(\widehat{f},\mathbf{x}) = \mathbb{1}\left\{\operatorname{sgn}(\widehat{f}(\mathbf{x})-1/2) \ne \operatorname{sgn}(f_\star(\mathbf{x})-1/2)\right\}\rho(f_\star(\mathbf{x})) ,$$

and $\rho(\cdot)$ has range in $[0,1]$.

As for the counterparts to Theorem 28, simply observe that Lemma 25 and Lemma 27 hold in the more general case with d replaced by V, where $\mathcal{F}$ is a function class having finite VC-subgraph dimension V.

In particular, given any $S_T = \{\mathbf{x}_1,\ldots,\mathbf{x}_T\} \in \mathcal{X}$, define

$$\Phi(S_T) = \left|\{[\phi(\widehat{f},\mathbf{x}_1),\ldots,\phi(\widehat{f},\mathbf{x}_T)] : \widehat{f}\in\mathcal{F}\}\right| .$$

We then have, when $T \ge V'$,

$$\Phi(S_T) \le \sum_{i=0}^{V'}\binom{T}{i}$$

where V' is the VC dimension of the binary-valued class

$$\left\{\operatorname{sgn}(\widehat{f}(\mathbf{x})-1/2) : \widehat{f}\in\mathcal{F}\right\} .$$

Now it is easy to see that $V' \le V$, so that we also have

$$\Phi(S_T) \le \sum_{i=0}^{V} \binom{T}{i} \le \left(\frac{eT}{V}\right)^V .$$

With this modification, Theorem 28 holds with factor $d\log\left(\frac{2eT}{d}\right)$ therein replaced by $V\log\left(\frac{2eT}{V}\right)$, once we also replace $\langle \widehat{\mathbf{w}}, \mathbf{x}\rangle$ by $\widehat{f}(\mathbf{x}) - 1/2$.

References

Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011*, pages 2312–2320, 2011.

K. Amin, C. Cortes, G. DeSalvo, and A. Rostamizadeh. Understanding the effects of batching in online active learning. In *Proc. AISTATS*, 2020.

J. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations (ICLR)*, 2020.

K. S. Azoury and M. K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43:211–246, 2001.

M.-F. Balcan, A. Broder, and T. Zhang. Margin based active learning. In *20th Annual Conference on Learning Theory (COLT)*, 2007.

N. Balcan and P. Long. Active and passive learning of linear separators under log-concave distributions. In *Proceedings of the 26th annual conference on Learning Theory (COLT)*, 2013.

Z. Borsos, M. Mutny, M. Tagliasacchi, and A. Krause. Data summarization via bilevel optimization, 2021. URL `https://arxiv.org/abs/2109.12534`.

N. Brukhim, M. Dudik, A. Pacchiano, and R. Schapire. A unified model and dimension for interactive estimation, 2023. URL `https://arxiv.org/abs/2306.06184`.

R. Camilleri, J. Katz-Samuels, and K. Jamieson. High-dimensional experimental design and kernel bandits. In *Proc. 38th International Conference on Machine Learning, PMLR 139*, 2021a.

R. Camilleri, Z. Xiong, M. Fazel, L. Jain, and K. Jamieson. Selective sampling for online best-arm identification. In *Advances in Neural Information Processing Systems*, 2021b.

R. Castro and R. Nowak. Minimax bounds for active learning. *IEEE Transactions on Information Theory*, 54(5):2339–2353, 2008.

N. Cesa-Bianchi, A. Conconi, and C. Gentile. A second-order perceptron algorithm. *SIAM J. Comput.*, 34(3):640–668, 2005.

K. Chaudhuri, S. M. Kakade, P. Netrapalli, and S. Sanghavi. Convergence rates of active learning for maximum likelihood estimation. *Advances in Neural Information Processing Systems*, 28, 2015.

Y. Chen and A. Krause. Near-optimal batch mode active learning and adaptive submodular optimization. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28(1), pages 160–168. PMLR, 2013.

Y. Chen, S. H. Hassani, A. Karbasi, and A. Krause. Sequential information maximization: When is greedy near-optimal? In *Proc. 28th Conference on Learning Theory, PMLR 40*, pages 338–363, 2015.

Y. Chen, S. H. Hassani, and A. Krause. Near-optimal bayesian active learning with correlated and noisy tests. In *Proc. 20th International Conference on Artificial Intelligence and Statistics*, 2017.

G. Citovsky, G. DeSalvo, C. Gentile, L. Karydas, A. Rajagopalan, A. Rostamizadeh, and S. Kumar. Batch active learning at scale. In *35th Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

C. Dann, M. Mohri, T. Zhang, and J. Zimmert. A provably efficient model-free posterior sampling method for episodic reinforcement learning. In *35th Conference on Neural Information Processing System*, 2021.

S. Dasgupta. Analysis of a greedy active learning strategy. In *Advances in Neural Information Processing Systems*, volume 17, 2004.

S. Dasgupta. Coarse sample complexity bounds for active learning. In *Advances in neural information processing systems*, pages 235–242, 2005.

O. Dekel, C. Gentile, and K. Sridharan. Selective sampling and active learning from single and multiple teachers. *J. Mach. Learn. Res.*, 13(1), 2012.

K. Dong, J. Yang, and T. Ma. Provable model-based nonlinear bandit and reinforcement learning: Shelve optimism, embrace virtual curvature, 2021. URL `https://arxiv.org/abs/2102.04168`.

H. Esfandiari, A. Karbasi, and V. Mirrokni. Adaptivity in adaptive submodularity. In *Proc. 34th Annual Conference on Learning Theory*, volume 134, pages 1–24. PMLR, 2021.

T. Fiez, L. Jain, K. G. Jamieson, and L. Ratliff. Sequential experimental design for transductive linear bandits. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

D. Foster, A. Rakhlin, D. Simchi-Levi, and Y. Xu. Instance-dependent complexity of contextual bandits and reinforcement learning: A disagreement-based perspective, 2020. URL `https://arxiv.org/abs/2010.03104`.

C. Gentile, Z. Wang, and T. Zhang. Fast rates in pool-based batch active learning, 2022a. URL `https://arxiv.org/abs/2202.05448`.

C. Gentile, Z. Wang, and T. Zhang. Achieving minimax rates in pool-based batch active learning. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 2022b.

A. Ghorbani, J. Zou, and A. Esteva. Data shapley valuation for efficient batch active learning, 2021. URL `https://arxiv.org/abs/2104.08312v1`.

D. Golovin and A. Krause. Adaptive submodularity: A new approach to active learning and stochastic optimization, 2017. URL `https://arxiv.org/abs/1003.3967`.

Q. Gu, T. Zhang, J. Han, and C. Ding. Selective labeling via error bound minimization. *Advances in neural information processing systems*, 25, 2012.

Q. Gu, T. Zhang, and J. Han. Batch-mode active learning via error bound minimization. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 300–309, 2014.

S. Hanneke. Adaptive rates of convergence in active learning. In *Proc. of the 22th Annual Conference on Learning Theory*, 2009.

S. Hanneke. Theory of disagreement-based active learning. *Foundations and Trends in Machine Learning*, 7(2–3):131–309, 2014.

S. Hoi, R. Jin, J. Zhu, and M. R. Lyu. Batch mode active learning and its application to medical image classification. In *23rd International Conference on Machine Learning (ICML)*, 2006.

K. Huang, S. Kakade, J. Lee, and Q. Lei. A short note on the relationship of information gain and eluder dimension, 2021. URL `https://arxiv.org/abs/2107.02377`.

J. Katz-Samuels, J. Zhang, L. Jain, and K. Jamieson. Improved algorithms for agnostic pool-based active classification. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 5334–5344. PMLR, 2021.

K. Killamsetty, D. Sivasubramanian, G. Ramakrishnan, and R. Iyer. Glister: Generalization based data subset selection for efficient and robust learning, 2021. URL `https://arxiv.org/abs/2012.10630`.

K. Kim, K. Park, D. Kim, and S. Chun. Task-aware variational adversarial active learning, 2020. URL `https://arxiv.org/abs/2002.04709v2`.

A. Kirsch, J. van Amersfoort, and Y. Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning, 2019. URL `https://arxiv.org/abs/1906.08158v2`.

A. Kirsch, S. Farquhar, and Y. Gal. A simple baseline for batch active learning with stochastic acquisition functions, 2021. URL `https://arxiv.org/abs/2106.12059`.

V. Koltchinskii. Rademacher complexities and bounding the excess risk of active learning. *Journal of Machine Learning Research*, 11:2457–2485, 2010.

S. Kothawade, N. Beck, K. Killamsetty, and R. Iyer. Similar: Submodular information measures based active learning in realistic scenarios. In *Advances in Neural Information Processing Systems*, 2021.

T. Lattimore and C. Szepesvari. *Bandit Algorithms.* Cambridge University Press, 2020.

E. Mammen and A. Tsybakov. Smooth discrimination analysis. *The Annals of Statistics*, 27(6):1808–1829, 1999.

S. Mukherjee, A. S. Tripathy, and R. Nowak. Chernoff sampling for active testing and extension to active regression. In *International Conference on Artificial Intelligence and Statistics*, pages 7384–7432. PMLR, 2022.

R. D. Nowak. The geometry of generalized binary search. *IEEE Transactions on Information Theory*, 57(12):7893–7906, 2011.

I. Osband and B. Van Roy. Model-based reinforcement learning and the eluder dimension. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

L. Pronzato and A. Pázman. Design of experiments in nonlinear models. *Lecture notes in statistics*, 212(1), 2013.

D. Russo and B. Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13:145–147, 1972.

A. Sekhari, K. Sridharan, W. Sun, and R. Wu. Selective sampling and imitation learning via online regression. In *Advances in Neural Information Processing Systems*, 2023.

O. Sener and S. Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=H1aIuk-RW`.

C. Shui, C. Zhou, F. Gagne, and B. Wang. Deep active learning: Unified and principled method for query and training. In *23rd International Conference on Artificial Intelligence and Statistics (AiSTATS)*, 2020.

C. Tosh and S. Dasgupta. Diameter-based active learning. In *34th International Conference on Machine Learning (ICML)*, 2017.

Z. Wang, P. Awasthi, C. Dann, and C. Sekhari, A. Gentile. Neural active learning with performance guarantees. In *Advances in Neural Information Processing Systems 34*, 2021.

K. Wei, R. Iyer, and J. Bilmes. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, pages 1954–1963. PMLR, 2015.

C. Ye, W. Xiong, Q. Gu, and T. Zhang. Corruption-robust algorithms with uncertainty weighting for nonlinear contextual bandits and Markov decision processes. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 39834–39863. PMLR, 2023.

C. Zhang and Y. Li. Improved algorithms for efficient active learning halfspaces with massart and tsybakov noise. In *34th Annual Conference on Learning Theory (COLT)*, 2021.

F. Zhdanov. Diverse mini-batch active learning, 2019. URL `https://arxiv.org/abs/1901.05954v1`.

D.-X. Zhou. The covering number in learning theory. *Journal of Complexity*, 18:739–767, 2002.

Y. Zhu and R. Nowak. Efficient active learning with abstention. In *Advances in Neural Information Processing Systems*, volume 35, pages 35379–35391. Curran Associates, Inc., 2022.