Achieving Minimax Rates in Pool-Based Batch Active Learning

Claudio Gentile [1] **Zhilei Wang** [2] **Tong Zhang** [1 3]

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 standard statistical learning settings. Our results also exhibit a mild dependence on the batch size. 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.

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 minimax rates of convergence of their excess risk as a function of the number of requested labels. Interestingly enough, these optimal rates are retained 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 or generalized linear models, 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 & Long, 2013; Zhang & 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

[1]Google Research, New York [2]Citadel Securities, New York [3]The Hong Kong University of Science and Technology, Hong Kong. Correspondence to: Zhilei Wang <zhileiwang92@gmail.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

realistic) batch pool-based active learning scenario. See also the related work contained in Section 3.

1.1. Content and contributions

Our contributions can be described as follows.

1. We 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 4.1) 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 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 4.2), still retaining the above-mentioned optimal rates.

3. We extend in Section 5 our results to the generalized linear case (specifically, the logistic case), and point out that restricting to exponentially small regions of the margin space is also beneficial for obtaining bounds with a milder dependence on the loss curvature.

4. Last but not least, despite we work out the details only for (generalized) linear models, 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, as briefly discussed in Section 6.

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 $h^*(x) = \operatorname{sgn}(f^*(x) - 1/2)$, 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 nowdays' Deep Neural Networks.

As a simple example, we consider 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 & 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

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

Notice, in particular, that $\alpha \to \infty$ gives the so-called *hard margin* condition $\mathbb{P}\big(|f^*(\mathbf{x}) - 1/2| < \epsilon\big) = 0$. 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 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. Notice 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 & Savarese, 2018; Kirsch et al., 2019; Ash et al., 2020; Killamsetty et al., 2020; Kirsch et al., 2021; Citovsky et al., 2021)), the one enabling tight theoretical analyses for our algorithms is a spectral-like diversity measure defined in the finite dimensional case $\mathcal{X} = \mathbb{R}^d$ as $D(\mathbf{x}, S) = \langle \mathbf{x}, \mathbf{x} \rangle_{A_S^{-1}}^{\frac{1}{2}} = ||\mathbf{x}||_{A_S^{-1}} = \sqrt{\mathbf{x}^\top A_S^{-1} \mathbf{x}}$, 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. Notice 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.

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 (Gu et al., 2012; 2014; Sener & Savarese, 2018; Kirsch et al., 2019; Zhdanov, 2019; Shui et al., 2020; Ash et al., 2020; Kim et al., 2020; Killamsetty et al., 2020; 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 subfield of statistical learning. General references in pool-based active learning include (Dasgupta, 2004; 2005; Hanneke, 2014; Nowak, 2011; Tosh & Dasgupta, 2017), and specific algorithms for half-spaces under classes of input distributions are contained, e.g., in (Balcan et al., 2007; Balcan & Long, 2013; Zhang & 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 & 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 ac-

tive 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.

Very recent stream-based active learning papers that are worth mentioning are (Katz-Samuels et al., 2021; Camilleri et al., 2021b)). These papers share similar methods and modeling assumptions as ours in leveraging optimal design, but they do 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.

A learning problem 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., 2020; Borsos et al., 2021).

4. The Linear Case

We start off 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 algorithm 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] $D(\mathbf{x}, \mathcal{Q}_\ell) = ||\mathbf{x}||_{A_{\ell,t-1}^{-1}}$ 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 $\mathcal{Q}_\ell$.

[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 ℓ:

- $\epsilon_\ell = 2^{-\ell}/(\sqrt{2\log\frac{2\ell(\ell+1)T}{\delta}} + 1)$
- $A_{\ell,0} = I,\ t = 0,\ \mathcal{Q}_\ell = \emptyset$

while $\mathcal{P}_{\ell-1}\backslash\mathcal{Q}_\ell \neq \emptyset$ *and* $\max_{\mathbf{x}\in\mathcal{P}_{\ell-1}\backslash\mathcal{Q}_\ell} \|\mathbf{x}\|_{A_{\ell,t}^{-1}} > \epsilon_\ell$

- $t = t + 1$
- Pick $\mathbf{x}_{\ell,t} \in \operatorname{argmax}_{\mathbf{x}\in\mathcal{P}_{\ell-1}\backslash\mathcal{Q}_\ell} \|\mathbf{x}\|_{A_{\ell,t-1}^{-1}}$
- 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 ℓ

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
$$\mathbf{w}_\ell = A_{\ell,T_\ell}^{-1} \sum_{t=1}^{T_\ell} y_{\ell,t}\mathbf{x}_{\ell,t}$$
- 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 $\hat{y} = \mathrm{sgn}\langle \mathbf{w}_\ell, \mathbf{x}\rangle$

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 no. 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 $\mathrm{sgn}(\langle \widehat{\mathbf{w}}, \mathbf{x}\rangle)$

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 $D(\mathbf{x}, \mathcal{Q}_\ell) \leq \epsilon_\ell$. 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., an SVM) 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, 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 $\cup_{\ell=1}^{L}\mathcal{C}_\ell$.

The fact that the algorithm only uses pseudo-labels to train its final predictor may look counter-intuitive at first, but this is due to our proof technique that derives an excess risk bound out of weighted empirical risk bounds — see, e.g., the proof sketch of Theorem 4.1. Algorithmically, the queried labels can be noisy and, in general, we do not know whether they are consistent with the Bayes optimal predictor $\mathbf{w}^*$. In this sense, the process of generating pseudo-labels can be seen as a *label denoising* process. This is made possible by our algorithm, which guarantees (with high probability) that for the selected data points, pseudo-labels generated by the model are consistent with those of the Bayes optimal predictor, while labels of other data (including those in the training data) may not.

Further, notice that the final predictor $\widehat{\mathbf{w}}$ need not be an SVM. Any training algorithm that returns a linear classifier which is consistent with the pseudo-labels will suffice. From our analysis we know that such a linear classifier has to exist (with high probability). Incidentally, this is the main the reason why relying on (denoised) pseudo-labels facilitates our statistical analysis, beyond the involved algorithmics.

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 & 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 $\mathbb{E}[y_t \mid \mathbf{x}_t] = \langle \mathbf{w}^*, \mathbf{x}_t\rangle$. 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 is the main result of this section.[2]

Theorem 4.1. *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

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

Proof sketch. We first derive a high-probability bound on the weighted empirical risk

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

and then turn it into an excess risk bound through a uniform convergence argument. In order to bound $R_T(\mathcal{P})$, we partition the points in $\mathcal{P}$ into the three subsets

$$\cup_{\ell=1}^{L}\mathcal{C}_\ell, \qquad \cup_{\ell=1}^{L}\mathcal{Q}_\ell, \qquad \mathcal{P}_L,$$

and consider the contribution to $R_T(\mathcal{P})$ of each subset separately.

When $\mathbf{x} \in \cup_{\ell=1}^{L}\mathcal{C}_\ell$, we show that the pseudo-labels $\hat{y}$ generated by the algorithm are with high probability consistent with those generated by $\mathbf{w}^*$, that is, $\mathrm{sgn}\langle\widehat{\mathbf{w}}, \mathbf{x}\rangle = \mathrm{sgn}\langle\mathbf{w}^*, \mathbf{x}\rangle$, hence those $\mathbf{x}$ do not contribute to the weighted empirical risk.

Any $\mathbf{x} \in \mathcal{Q}_\ell$, is shown to contribute to $R_T(\mathcal{P})$ by at most $2^{-\ell}$, thus the overall contribution of $\cup_{\ell=1}^{L}\mathcal{Q}_\ell$ can be bounded by $\sum_{\ell=1}^{L} T_\ell/2^\ell$. In turn, by the way points are picked, T_ℓ is roughly bounded by d/ϵ_ℓ^2, allowing us to conclude that the total contribution of $\cup_{\ell=1}^{L}\mathcal{Q}_\ell$ is bounded by

$$\sum_{\ell=1}^{L} 2^{-\ell} d/\epsilon_\ell^2 \approx d/\epsilon_L .$$

Next, for $\mathbf{x} \in \mathcal{P}_L$, we show that (with high probability) it must be $|\langle\mathbf{w}^*, \mathbf{x}\rangle| \leq 2^{-L}$ which, combined with the stopping condition defining L implies an overall contribution of the same form d/ϵ_L.

Finally, since L is itself a random variable, we need to devise high probability upper bounds on it. We rely on the low noise assumption (2) to conclude that L is with high probability of the form

$$\max\left\{\frac{\log(T/d)}{\alpha+2},\ \log\left(\frac{4}{\epsilon_0}\right)\right\} ,$$

which we replace back into the previous bounds yielding a guarantee of the form

$$R_T(\mathcal{P}) \lesssim \max\left\{d^{\frac{\alpha+1}{\alpha+2}}\, T^{\frac{1}{\alpha+2}},\ \frac{d}{\epsilon_0}\right\} ,$$

hence an excess risk bound of the form

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

The analysis of the label complexity $N_T(\mathcal{P}) = \sum_{\ell=1}^{L} T_\ell$ follows a similar pattern, but it does not require uniform convergence. □

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 4.1, 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 overcount 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 4.1, 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 4.2. *With the same assumptions and notation as in Theorem 4.1, 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:*

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

where $\bar{C}, C(\delta, T, \epsilon_0)$ are the same as in Theorem 4.1.

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 4.1 to the label complexity bound in Corollary 4.2, one can see that when $B = O(T^{\frac{2}{\alpha+2}})$ we have with high probability

$$\mathcal{L}(\widehat{\mathbf{w}}) - \mathcal{L}(\mathbf{w}^*) \approx \frac{1}{(N_{T,B}(\mathcal{P}))^{\frac{1+\alpha}{2}}} , \qquad (3)$$

   which is the minimax rate one can achieve for VC classes[3] under the low-noise condition (2) with exponent α (e.g., (Castro & 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 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 & 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 & 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).

[3] Notice that if we disregard the dependence on the VC-dimension, and only focus on the dependence on the label complexity $N_{T,B}(\mathcal{P})$, all these rates have the same form (3).

5. The Logistic Case

We now discuss how to extend the result of the previous section to the logistic case (the generalized linear model (1) with $\sigma(z) = \frac{e^z}{1+e^z}$).

Algorithm 2 is the adaptation to the logistic case of the algorithm of Section 4, the main difference being that we now assume the comparison vector $\mathbf{w}^*$ to lie in a Euclidean ball of (known) radius R, and compute estimators $\mathbf{w}_\ell$ as regularized logistic regressors:

$$\mathbf{w}_\ell = \operatorname*{argmin}_{\mathbf{w}\,:\,\max_{\mathbf{x}\in\mathcal{Q}_\ell} |\langle \mathbf{w},\mathbf{x}\rangle| \le 2R_{\ell-1}} \left[\sum_{t=1}^{T_\ell} Loss(y_{\ell,t}\langle \mathbf{w}, \mathbf{x}_{\ell,t}\rangle) + \frac{1}{8} e^{-4R_\ell} \|\mathbf{w}\|^2 \right] , \qquad (4)$$

where $Loss(\cdot)$ is the logistic function

$$Loss(a) = \log(1 + e^{-a}) .$$

One of the main concerns in the logistic case is to investigate how excess risk and label complexity bounds depend on the complexity R of the comparison class. The following is the logistic counterpart to Theorem 4.1.

Theorem 5.1. *Let $T \ge d$ and assume that $\|\mathbf{x}\|_2 \le 1$ 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 2 are simultaneously upper*

Algorithm 2: Pool-based batch active learning algorithm for logistic models.

Input: Confidence level $\delta \in (0,1]$, pool of instances $\mathcal{P} \subseteq \mathbb{R}^d$ of size $|\mathcal{P}| = T$, upper bound $R > 0$ on $||\mathbf{w}^*||$

Initialize: $\mathcal{P}_0 = \mathcal{P}$

for $\ell = 1, 2, \ldots,$

Initialize within stage ℓ:

- $R_\ell = R\, 2^{-\ell}$
- $\epsilon_\ell = R_\ell / \left(16 e^{8R_\ell} \sqrt{d \log \frac{2d\ell(\ell+1)}{\delta}} + 4Re^{4R_\ell}\right)$
- $A_{\ell,0} = I$, $t = 0$, $\mathcal{Q}_\ell = \emptyset$

while $\mathcal{P}_{\ell-1} \backslash \mathcal{Q}_\ell \neq \emptyset$ *and* $\max_{\mathbf{x} \in \mathcal{P}_{\ell-1} \backslash \mathcal{Q}_\ell} \|\mathbf{x}\|_{A_{\ell,t}^{-1}} > \epsilon_\ell$

- $t = t + 1$
- Pick $\mathbf{x}_{\ell,t} \in \operatorname{argmax}_{\mathbf{x} \in \mathcal{P}_{\ell-1} \backslash \mathcal{Q}_\ell} \|\mathbf{x}\|_{A_{\ell,t-1}^{-1}}$
- 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 ℓ

if $\mathcal{Q}_\ell \neq \emptyset$

- Query the labels $y_{\ell,1}, \ldots, y_{\ell,T_\ell}$ associated with the unlabeled data in $\mathcal{Q}_\ell$
- Compute $\mathbf{w}_\ell$ as in (4)
- Set $\mathcal{C}_\ell = \{\mathbf{x} \in \mathcal{P}_{\ell-1} \backslash \mathcal{Q}_\ell : |\langle \mathbf{w}_\ell, \mathbf{x} \rangle| > R_\ell\}$
- 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 $d/(2R_\ell) > 2R_\ell |\mathcal{P}_\ell|$

- $L = \ell$
- Exit the for-loop (L is the total no. 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 $\operatorname{sgn}(\langle \widehat{\mathbf{w}}, \mathbf{x} \rangle)$

bounded as follows:

$$\mathcal{L}(\widehat{\mathbf{w}}) - \mathcal{L}(\mathbf{w}^*) = \bar{C} C_{d,R}(\delta, T, \epsilon_0) \left(\max\left\{ \left(\frac{d}{T}\right)^{\frac{\alpha+1}{\alpha+2}}, \frac{d}{T\epsilon_0} \right\} + \frac{\log\left(\frac{\log T}{\delta}\right) + de^{8R}\lceil \log_2 R \rceil}{T} \right),$$

$$N_T(\mathcal{P}) = \bar{C} C_{d,R}(\delta, T, \epsilon_0) \left(\max\left\{ d^{\frac{\alpha}{\alpha+2}} T^{\frac{2}{\alpha+2}}, \frac{d}{\epsilon_0^2} \right\} + \frac{\log^2\left(\frac{\log T}{\delta}\right) + de^{8R}\lceil \log_2 R \rceil}{T} \right),$$

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

where $\bar{C}$ is an absolute constant and

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

In the above bounds, the complexity term R is meant to be a constant. Notice that the dependence on e^R is common to many logistic bounds, specifically in the bandits literature. This is due to the nonlinear shape of $\sigma(\cdot)$ (see, e.g., (Filippi et al., 2010; Gentile & Orabona, 2012; Li et al., 2017; Faury et al., 2020), where it takes the form of an upper bound on $1/\sigma'(\cdot)$). In fact, a closer look at the multiplicative dependence on e^R above reveals that this factor multiplies only *logarithmic* terms in T. This is akin to the more refined self-concordant analysis of logistic models contained in (Faury et al., 2020). Since our algorithm is focusing attention to exponentially shrinking regions of margin values $\langle \mathbf{w}^*, \mathbf{x} \rangle$ around the origin, we have obtained here similar guarantees without resorting to a self-concordant analysis.

A constant batch size version of Algorithm 2 can also be devised, and the associated properties spelled out. The details are very similar to those in Section 4.2, and are therefore omitted.

6. Conclusions and Ongoing Research

We have described and analyzed novel batch active learning algorithms in the pool-based setting that 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. In the logistic case, this has the side benefit of delivering performance bounds where the classical exponential dependence on the complexity of the comparator $\mathbf{w}^*$ occurs as a multiplicative factor only in logarithmic terms.

The logistic algorithm we presented in Section 5 has a suboptimal dependence on the input dimension d (notice the extra factor d contained in C_1 in the excess risk bound of Theorem 5.1), and we are currently trying to see if it is possible to achieve the same result as in the linear case. For the logistic case, a more computationally efficient algorithm actually exists that is based on the online Newton step-like analysis in (Gentile & Orabona, 2012). Yet, this algorithm will have a similar suboptimal dependence on d.

Related to the above, we are currently investigating to what extent it is possible to improve the logistic analysis so as to turn the constrained minimization problem therein into an unconstrained one. Analyses we are aware of in the contiguous field of contextual bandits in generalized linear scenarios (e.g., (Li et al., 2017)) do not seem to help, given the strong assumptions on the context distribution they formulate to achieve the optimal dependence on d.

The methods we have presented here are instances of a more general approach to batch active learning in realizable settings where, given a diversity measure $D(\mathbf{x}, S)$, an estimator $\widehat{f} = \widehat{f}(S)$ in fixed design scenarios exists for which we can guarantee L_∞ approximation bounds of the form

$$|\widehat{f}_S(\mathbf{x}) - f^*(x)| \leq D(\mathbf{x}, S) \qquad \forall \mathbf{x}\,. \tag{5}$$

For instance, our approach can be seamlessly extended to the case where f^* belongs to a RKHS, the algorithmic aspects simply requiring a dual variable formulation of Algorithm 1, and the statistical ones simply resorting to covering number bounds (e.g., (Zhou, 2002)) or empirical versions thereof. As another relevant example, (5) can be shown to hold for known plug-in estimators, like local polynomial estimators (e.g., Sect. 1.6.1. in (Tsybakov, 2009)). Hence our general approach may be extended to those cases as well.

References

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

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

Balcan, M.-F., Broder, A., and Zhang, T. Margin based active learning. In *COLT*, 2007.

Balcan, N. and Long, P. Active and passive learning of linear separators under log-concave distributions. In *Colt 2013*, 2013.

Borsos, Z., Mutny, M., Tagliasacchi, M., and Krause, A. Data summarization via bilevel optimization, 2021.

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

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

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

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

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

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

Citovsky, G., DeSalvo, G., Gentile, C., Karydas, L., Rajagopalan, A., Rostamizadeh, A., and Kumar, S. Batch active learning at scale. In *Neurips 2021*, 2021.

Dasgupta, S. Analysis of a greedy active learning strategy. In *NIPS*, 2004.

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

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

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

Faury, L., Abeille, M., Calauzènes, C., and Fercoq, O. Improved optimistic algorithms for logistic bandits. In *37th ICML*, 2020.

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

Filippi, S., Cappe', O., Garivier, A., and Szepesvari, C. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, pp. 586–594, 2010.

Gentile, C. and Orabona, F. On multilabel classification and ranking with partial feedback. In *Advances in Neural Information Processing Systems*, volume 25, pp. 1151–1159. Curran Associates, Inc., 2012.

Ghorbani, A., Zou, J., and Esteva, A. Data shapley valuation for efficient batch active learning. In *arXiv:2104.08312v1*, 2021.

Golovin, D. and Krause, A. Adaptive submodularity: A new approach to active learning and stochastic optimization. In *arXiv:1003.3967*, 2017.

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

Gu, Q., Zhang, T., and Han, J. Batch-mode active learning via error bound minimization. In *UAI*, pp. 300–309. Citeseer, 2014.

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

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

Hoi, S., Jin, R., Zhu, J., and Lyu, M. R. Batch mode active learning and its application to medical image classification. In *ICML*, 2006.

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

Killamsetty, K., Sivasubramanian, D., Ramakrishnan, G., and Iyer, R. Glister: Generalization based data subset selection for efficient and robust learning. *arXiv preprint arXiv:2012.10630*, 2020.

Kim, K., Park, D. Kim, K., and Chun, S. Task-aware variational adversarial active learning. In *arXiv:2002.04709v2*, 2020.

Kirsch, A., van Amersfoort, J., and Gal, Y. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *arXiv:1906.08158v2*, 2019.

Kirsch, A., Farquhar, S., and Gal, Y. A simple baseline for batch active learning with stochastic acquisition functions. *arXiv preprint arXiv:2106.12059*, 2021.

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

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

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

Li, L., Lu, Y., and Zhou, D. Provably optimal algorithms for generalized linear contextual bandits. In *International Conference on Machine Learning*, pp. 2071–2080. PMLR, 2017.

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

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

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

Sener, O. and Savarese, S. 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.

Shui, C., Zhou, F. Gagne, C., and Wang, B. Deep active learning: Unified and principled method for query and training. In *Proc. AiSTATS 2020*, 2020.

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

Tsybakov, A. *Introduction to Nonparametric Estimation*. Springer, 2009.

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

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

Zhang, C. and Li, Y. Improved algorithms for efficient active learning halfspaces with massart and tsybakov noise. In *COLT*, 2021.

Zhdanov, F. Diverse mini-batch active learning. In *arXiv:1901.05954v1*, 2019.

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

A. Proofs for Section 4

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

We denote for any $\epsilon > 0$,

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

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

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

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

Lemma A.1. *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 Lattimore & 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 $1, \dots, \ell-1$, 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 $\sqrt{\sum_{t=1}^{T_\ell} \langle \mathbf{x}_{\ell,t}, \mathbf{x}\rangle^2_{A_{\ell,T_\ell}^{-1}}}$-sub-Gaussian. We apply lemma C.5

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

Now observe that

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

We plug back into the previous inequality to obtain

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

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

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

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

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

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

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

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

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

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

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 A.2. *Assume $\bigcap_{\ell=1}^L\mathcal{E}_\ell$ holds. Then Algorithm 1 generates pseudo-labels such that, on all points $\mathbf{x}\in\cup_{\ell=1}^L\mathcal{C}_\ell$, $\operatorname{sgn}(\langle \mathbf{w}_\ell,\mathbf{x}\rangle)=\operatorname{sgn}(\langle \mathbf{w}^*,\mathbf{x}\rangle)$.*

Proof. Simply observe that if $\mathbf{x}\in\cup_{\ell=1}^L\mathcal{C}_\ell$ is such that $\operatorname{sgn}(\langle \mathbf{w}_\ell,\mathbf{x}\rangle)=1$ 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, $\operatorname{sgn}(\langle \mathbf{w}_\ell,\mathbf{x}\rangle)=-1$ implies $\langle \mathbf{w}^*,\mathbf{x}\rangle<0$. □

Lemma A.3. *The length T_ℓ of stage ℓ in Algorithm 1 is (deterministically) upper bounded as*

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

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

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

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

$$|A_{\ell,t+1}|=|A_{\ell,t}+\mathbf{x}_{\ell,t+1}\mathbf{x}_{\ell,t+1}^\top|=|A_{\ell,t}|\cdot|I+A_{\ell,t}^{-1}\mathbf{x}_{\ell,t+1}\mathbf{x}_{\ell,t+1}^\top|=(1+||\mathbf{x}_{\ell,t+1}||^2_{A_{\ell,t}^{-1}})|A_{\ell,t}|\leq 2|A_{\ell,t}|\,,$$

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

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

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

Therefore,

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

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

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

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

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

yields

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

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

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

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

As a consequence,

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

□

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

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

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 A.3 immediately delivers the following bound on $N_T(\mathcal{P})$:

Theorem A.4. *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*

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

Proof. By definition

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

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 A.5. *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*

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

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 $\mathcal{C}_\ell$,

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

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

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

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 $\operatorname{sgn}\langle \widehat{\mathbf{w}}, \mathbf{x} \rangle = \operatorname{sgn}\langle \mathbf{w}^*, \mathbf{x} \rangle$ 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 $|\langle \mathbf{w}_{\ell-1}, \mathbf{x} \rangle| \leq 2^{-\ell+1}$ which, combined with the assumption that $\mathcal{E}_{\ell-1}$ holds, yields $|\langle \mathbf{w}^*, \mathbf{x} \rangle| \leq 2^{-\ell+2}$. Since $|\mathcal{Q}_\ell| = T_\ell$, we have

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

and Lemma A.3 allows us to write

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

the last inequality following from a reasoning similar to the one that lead us to theorem A.4. □

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 $\epsilon_* > \frac{\epsilon_0}{2^{M/\alpha}}$,

$$|\mathcal{T}_{\epsilon_*}| \leq |\mathcal{T}_{\epsilon}| \leq \frac{3}{2}\left(T\epsilon^\alpha + \log\left(\frac{M}{\delta}\right)\right) < 3\left(T\epsilon_*^\alpha + \log\left(\frac{M}{\delta}\right)\right) . \tag{6}$$

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. (6) 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

$$\mathbb{P}\left(\bar{\mathcal{E}}\right) \geq 1 - \delta , \tag{7}$$

for $M = \log_2 T$.

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

$$d/\epsilon_* = 3\left(T\epsilon_*^{\alpha+1} + \epsilon_* \log\left(\frac{M}{\delta}\right)\right) . \tag{8}$$

Eq. (6) will be applied, in particular, to the margin value 2^{-L+2} when $2^{-L+2} \leq \epsilon_0$.

Armed with Eqs. (6) and (8) with $M = \log_2 T$, we prove a lemma that upper bounds the number of stages L.

Lemma A.6. *Let ϵ_* be defined through (8), 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*

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

Here the O-notation only omits absolute constants.

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

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

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

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

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

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

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

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

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

$$d/2^{-L+2} \leq 3\left(T2^{(-L+2)(\alpha+1)} + 2^{-L+2}\log(\frac{M}{\delta})\right) \,,$$

which resembles (8) with 2^{-L+2} here playing the role of ϵ^* therein. Then, from the definition of ϵ^* in (8) we immediately obtain $2^{-L+2} \geq \epsilon_*$, thus $L \leq \log_2(\frac{1}{\epsilon_*}) + 2$. Moreover, from (8) we see that $d/\epsilon_* \geq 3T\epsilon_*^{\alpha+1}$, which is equivalent to $\epsilon_* \leq (\frac{d}{3T})^{\frac{1}{\alpha+2}}$. Replacing this upper bound on ϵ^* back into the right-hand side of (8) and dividing by d yields

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

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

□

Corollary A.7. *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 A.6, we split the analysis into two cases depending on whether or not 2^{-L+2} is bigger than ϵ_0. If $2^{-L+2} \leq \epsilon_0$, we have

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

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 A.4 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 A.5,

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

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 A.4 and Theorem A.5 gives

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

Lastly, (7) and lemma A.1 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

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

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

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

has the same form as the function $\phi(\widehat{\mathbf{w}},\mathbf{x})$ in Appendix C on which the uniform convergence result of Theorem C.4 applies, with $\widehat{\epsilon}(\delta)$ therein replaced by the bound on $R_T(\mathcal{P})$ borrowed from Corollary A.7. 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 4.1 in the main body of the paper.

B. Proofs for Section 5

We adopt the same notation as in Section A and follow the same proof structure.

Define the loss function

$$Loss(a) = \log(1 + e^{-a}) \,,$$

and the sigmoidal function

$$\sigma(a) = \frac{1}{1 + e^{-a}} \,.$$

The noise model in the main body of the paper can be re-formulated as follows: there exists an unknown vector $\mathbf{w}^*$ belonging to a Euclidean ball of radius $R \geq 1$ such that for any instance $\mathbf{x}$ of Euclidean norm at most 1,

$$\mathbb{P}(y = 1 \,|\, \mathbf{x}) = \sigma(\langle \mathbf{w}^*, \mathbf{x} \rangle) \,.$$

Therefore we have

$$\mathbb{E}\left[y \mid \mathbf{x}\right] = \sigma(\langle \mathbf{w}^*, \mathbf{x} \rangle) - \sigma(-\langle \mathbf{w}^*, \mathbf{x} \rangle) = 2\sigma(\langle \mathbf{w}^*, \mathbf{x} \rangle) - 1 \,,$$

and the noise variable ξ can be written as

$$\xi := y - \mathbb{E}\left[y \mid \mathbf{x}\right] = \frac{2y}{1 + e^{y\langle \mathbf{w}^*, \mathbf{x} \rangle}} \,.$$

Similar to the linear case, we denote for any $\epsilon > 0$,

$$\mathcal{T}_\epsilon^\sigma = \{\mathbf{x} \in \mathcal{P} \,:\, |2\sigma\left(\langle \mathbf{w}^*, \mathbf{x} \rangle\right) - 1| \leq \epsilon\} \,.$$

Now, recall the notation in Algorithm 2. Similar to $\mathcal{E}_\ell$ defined in linear case, it will be useful to define the events

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

where $R_\ell = R2^{-\ell}$ for $\ell = 0, \ldots, L$.

Lemma B.1. *For any positive L,*

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

Proof. We decompose the above quantity as

$$\mathbb{P}\left(\bigcap_{\ell=1}^{L} \mathcal{E}_\ell \right) = \mathbb{P}\left(\mathcal{E}_L \,\middle|\, \bigcap_{\ell=1}^{L-1} \mathcal{E}_\ell \right) \mathbb{P}\left(\mathcal{E}_{L-1} \,\middle|\, \bigcap_{\ell=1}^{L-2} \mathcal{E}_\ell \right) \ldots \mathbb{P}(\mathcal{E}_2 | \mathcal{E}_1) \mathbb{P}(\mathcal{E}_1) \,,$$

and bound each factor individually.

At the beginning of the stage ℓ, the remaining pool is $\mathcal{P}_{\ell-1}$, and $\sup_{x \in \mathcal{P}_{\ell-1}} |\langle \mathbf{w}_{l-1}, \mathbf{x} \rangle| \leq R_{\ell-1}$.

For $\ell \geq 2$, if $\mathcal{E}_{\ell-1}$ holds then

$$\sup_{x \in \mathcal{P}_{\ell-1}} |\langle \mathbf{w}^*, \mathbf{x} \rangle| \leq 2R_{\ell-1} \,. \tag{9}$$

Note that (9) also holds for $\ell = 1$ since $\|\mathbf{w}^*\| \leq R$ and $\|\mathbf{x}\| \leq 1$.

Now, for any positive number b, let

$$\Omega_\ell(b) = \{\mathbf{w} \in \mathbb{R}^d \,:\, \max_{\mathbf{x} \in \mathcal{Q}_\ell} |\langle \mathbf{w}, \mathbf{x} \rangle| \leq b\},$$

which is a convex compact set of $\mathbf{w}$'s.

The predictor $\mathbf{w}_\ell$ in Eq. (4) in the main body is defined as the solution of the following constraint minimization problem:

$$\mathbf{w}_\ell = \operatorname*{argmin}_{\mathbf{w}\in\Omega_\ell(2R_{\ell-1})} \left[\sum_{t=1}^{T_\ell} Loss(y_{\ell,t}\langle \mathbf{w}, \mathbf{x}_{\ell,t}\rangle) + \frac{1}{8}e^{-4R_\ell}\|\mathbf{w}\|^2\right],$$

For simplicity, from now on we omit the stage index ℓ from the subscripts of $\mathbf{x}_{\ell,t}$ and $y_{\ell,t}$ and denote A_{ℓ,T_ℓ} as A_ℓ. For $t = 1, \ldots, T_\ell$, denote

$$g_t(\mathbf{w})\mathbf{x}_t = \nabla_{\mathbf{w}} Loss(y_t\langle \mathbf{w}, \mathbf{x}_t\rangle) = -\frac{y_t}{1+\exp(y_t\langle \mathbf{w}, \mathbf{x}_t\rangle)}\mathbf{x}_t$$
$$h_t(\mathbf{w})\mathbf{x}_t\mathbf{x}_t^\top = \nabla^2_{\mathbf{w}} Loss(y_t\langle \mathbf{w}, \mathbf{x}_t\rangle) = \frac{1}{2\big(1+\cosh(y_t\langle \mathbf{w}, \mathbf{x}_t\rangle)\big)}\mathbf{x}_t\mathbf{x}_t^\top.$$

Notice that by definition

$$g_t(\mathbf{w}^*) = -\frac{1}{2}\xi_t\,,$$

where ξ_t is the noise term $\xi_t = y_t - \mathbb{E}[y_t \,|\, \mathbf{x}_t]$. Since $\cosh(\cdot)$ is an even function,

$$h_t(\mathbf{w}) = \frac{1}{2\big(1+\cosh(\langle \mathbf{w}, \mathbf{x}_t\rangle)\big)}$$

does not depend on y_t.

Since $\mathbf{w}^* \in \Omega_\ell(2R_{\ell-1})$ (as a consequence of (9)), the assumption that $\mathcal{E}_{\ell-1}$ holds and the optimality of $\mathbf{w}_\ell$ in $\Omega_\ell(2R_{\ell-1})$ allow us to write

$$\langle \mathbf{g}(\mathbf{w}_\ell) + \frac{1}{4}e^{-4R_\ell}\mathbf{w}_\ell, \mathbf{w}^* - \mathbf{w}_\ell\rangle \geq 0\,,$$

where

$$\mathbf{g}(\mathbf{w}) = \sum_{t=1}^{T_\ell} g_t(\mathbf{w})\mathbf{x}_t\,.$$

It follows that

$$\langle \mathbf{g}(\mathbf{w}^*) - \mathbf{g}(\mathbf{w}_\ell), \mathbf{w}^* - \mathbf{w}_\ell\rangle \leq \langle \mathbf{g}(\mathbf{w}^*), \mathbf{w}^* - \mathbf{w}_\ell\rangle + \frac{1}{4}e^{-4R_\ell}\langle \mathbf{w}_\ell, \mathbf{w}^* - \mathbf{w}_\ell\rangle\,. \tag{10}$$

For each $t = 1, \ldots, T_\ell$, the mean-value theorem insures the existence of a constant $\mu_\ell^t \in [0, 1]$ such that for

$$\mathbf{w}_\ell^t = (1-\mu_\ell^t)\mathbf{w}_\ell + \mu_\ell^t\mathbf{w}^*\,,$$

we have

$$g_t(\mathbf{w}^*) - g_t(\mathbf{w}_\ell) = h_t(\mathbf{w}_\ell^t)\langle \mathbf{w}^* - \mathbf{w}_\ell, \mathbf{x}_t\rangle\,.$$

Since

$$|\langle \mathbf{w}_\ell^t, \mathbf{x}_t\rangle| \leq (1-\mu_\ell^t)|\langle \mathbf{w}_\ell, \mathbf{x}_t\rangle| + \mu_\ell^t|\langle \mathbf{w}^*, \mathbf{x}_t\rangle| \leq 2R_{\ell-1} = 4R_\ell\,,$$

we have

$$h_t(\mathbf{w}_\ell^t) = \frac{1}{2\big(1+\cosh(\langle \mathbf{w}_\ell^t, \mathbf{x}_t\rangle)\big)} \geq \frac{1}{4}e^{-|\langle \mathbf{w}_\ell^t, \mathbf{x}_t\rangle|} \geq \frac{1}{4}e^{-4R_\ell}\,.$$

Introduce now the matrix

$$H_\ell := \sum_{t=1}^{T_\ell} h_t(\mathbf{w}_\ell^t)\mathbf{x}_t\mathbf{x}_t^\top + \frac{1}{4}e^{-4R_\ell}I\,,$$

where I is the $d \times d$ identity matrix. We can write

$$\mathbf{g}(\mathbf{w}^*) - \mathbf{g}(\mathbf{w}_\ell) = (H_\ell - \frac{1}{4}e^{-4R_\ell}I)(\mathbf{w}^* - \mathbf{w}_\ell)\,.$$

As a consequence, (10) implies

$$\begin{aligned}\langle H_\ell(\mathbf{w}^*-\mathbf{w}_\ell),\mathbf{w}^*-\mathbf{w}_\ell\rangle &\le \langle \mathbf{g}(\mathbf{w}^*),\mathbf{w}^*-\mathbf{w}_\ell\rangle + \frac{1}{4}e^{-4R_\ell}\|\mathbf{w}^*-\mathbf{w}_\ell\|^2 + \frac{1}{4}e^{-4R_\ell}\langle \mathbf{w}_\ell,\mathbf{w}^*-\mathbf{w}_\ell\rangle \\ &= \langle \mathbf{g}(\mathbf{w}^*) + \frac{1}{4}e^{-4R_\ell}\mathbf{w}^*,\mathbf{w}^*-\mathbf{w}_\ell\rangle \\ &\le \left(\|\mathbf{g}(\mathbf{w}^*)\|_{H_\ell^{-1}} + \frac{1}{4}e^{-4R_\ell}\|\mathbf{w}^*\|_{H_\ell^{-1}}\right)\|\mathbf{w}^*-\mathbf{w}_\ell\|_{H_\ell}\,.\end{aligned}$$

We thus obtain

$$\|\mathbf{w}^*-\mathbf{w}_\ell\|_{H_\ell} \le \|\mathbf{g}(\mathbf{w}^*)\|_{H_\ell^{-1}} + \frac{1}{4}e^{-4R_\ell}\|\mathbf{w}^*\|_{H_\ell^{-1}} \le 4e^{4R_\ell}\|\mathbf{g}(\mathbf{w}^*)\|_{A_\ell^{-1}} + R\,,$$

where in the second inequality we used $H_\ell \succeq \frac{1}{4}e^{-4R_\ell}A_\ell$.

To bound $\|\mathbf{g}(\mathbf{w}^*)\|_{A_\ell^{-1}}$, note that

$$\|\mathbf{g}(\mathbf{w}^*)\|^2_{A_\ell^{-1}} = \|\sum_{t=1}^{T_\ell} g_t(\mathbf{w}^*)A_\ell^{-1/2}\mathbf{x}_t\|_2^2 = \frac{1}{2}\|\sum_{t=1}^{T_\ell}\xi_t A_\ell^{-1/2}\mathbf{x}_t\|_2^2\,.$$

We plug in $A=[A_\ell^{-1/2}\mathbf{x}_1,\dots,A_\ell^{-1/2}\mathbf{x}_{T_\ell}]$, $\xi=(\xi_1,\dots,\xi_{T_\ell})$ into lemma C.6 and get with probability at least $1-\frac{\delta}{\ell(\ell+1)}$,

$$\|\mathbf{g}(\mathbf{w}^*)\|^2_{A_\ell^{-1}} \le \log\frac{2d\ell(\ell+1)}{\delta}\,\mathrm{tr}\left(A_\ell^{-1/2}\sum_{t=1}^{T_\ell}\mathbf{x}_t\mathbf{x}_t^\top A_\ell^{-1/2}\right) = \left(d-\mathrm{tr}(A_\ell^{-1})\right)\log\frac{2d\ell(\ell+1)}{\delta} < d\log\frac{2d\ell(\ell+1)}{\delta}\,.$$

Thus for any $\mathbf{x}\in\mathcal{P}_{\ell-1}\backslash\mathcal{Q}_\ell$, we obtain that with probability at least $1-\frac{\delta}{\ell(\ell+1)}$:

$$\begin{aligned}|\langle \mathbf{w}^*-\mathbf{w}_\ell,\mathbf{x}\rangle| &\le \|\mathbf{x}\|_{H_\ell^{-1}}\|\mathbf{w}^*-\mathbf{w}_\ell\|_{H_\ell} \\ &\le 4e^{4R_\ell}\epsilon_\ell\left(4e^{4R_\ell}\|\mathbf{g}(\mathbf{w}^*)\|_{A_\ell^{-1}}+R\right) \\ &\le \epsilon_\ell\left(16e^{8R_\ell}\sqrt{d\log\frac{2d\ell(\ell+1)}{\delta}}+4e^{4R_\ell}R\right)\,.\end{aligned}$$

Recalling the definition of ϵ_ℓ in Algorithm 2, we have, with probability at least $1-\frac{\delta}{\ell(\ell+1)}$,

$$\max_{\mathbf{x}\in\mathcal{P}_{\ell-1}\backslash\mathcal{Q}_\ell}|\langle \mathbf{w}_\ell-\mathbf{w}^*,\mathbf{x}\rangle| \le R_\ell\,,$$

that is, $\mathbb{P}(\mathcal{E}_\ell|\bigcap_{s=1}^{\ell-1}\mathcal{E}_s^\sigma) \ge 1-\frac{\delta}{\ell(\ell+1)}$ (for $\ell=1$ the above analysis gives $\mathbb{P}(\mathcal{E}_1^\sigma)\ge 1-\frac{\delta}{2}$). Hence

$$\mathbb{P}\left(\bigcap_{\ell=1}^{L}\mathcal{E}_\ell\right) \ge \prod_{\ell=1}^{L}\left(1-\frac{\delta}{\ell(\ell+1)}\right) \ge 1-\delta\sum_{\ell=1}^{L}\frac{1}{\ell(\ell+1)} > 1-\delta\,,$$

thereby concluding the proof. □

Similar to linear case, Lemma A.2 and Lemma A.3 also hold for logistic case.

We define the weighted cumulative regret for the logistic case as

$$R_T(\mathcal{P}) = \sum_{\mathbf{x}\in\mathcal{P}}\mathbb{1}\{\mathrm{sgn}\langle\widehat{\mathbf{w}},\mathbf{x}\rangle \ne \mathrm{sgn}\langle\mathbf{w}^*,\mathbf{x}\rangle\}|2\sigma(\langle\mathbf{w}^*,\mathbf{x}\rangle)-1|\,,$$

where $\widehat{\mathbf{w}}$ is the model output by Algorithm 2. Notice that since $|2\sigma(x)-1|\le|x|/2$ for all x, we alternatively upper bound

$$R_T(\mathcal{P}) = \sum_{\mathbf{x}\in\mathcal{P}}\mathbb{1}\{\mathrm{sgn}\langle\widehat{\mathbf{w}},\mathbf{x}\rangle \ne \mathrm{sgn}\langle\mathbf{w}^*,\mathbf{x}\rangle\}|\langle\mathbf{w}^*,\mathbf{x}\rangle|/2\,,$$

To simplify the math display we denote

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

then $\epsilon_\ell = \frac{R_\ell}{16e^{8R_\ell} K_d(\delta,\ell) + 4Re^{4R_\ell}}$. Note that here the factor $K_d(\delta, \ell)$ doesn't depend on T but has a $\sqrt{d}$ dependence.

To bound the number of queries note that lemma A.3 still holds, we use this to prove the following result.

Theorem B.2. *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*

$$N_T(\mathcal{P}) = d \max_{\ell \in [L]} \log\left(\frac{1}{\epsilon_\ell}\right) O\left(K_d^2(\delta, L) e^{8R\lceil \log_2 R \rceil} + e^{4R\lceil \log_2 R \rceil} R^2 + \frac{K_d^2(\delta, L) + R^2}{R_L^2}\right) \,,$$

where the O-notation only omits absolute constants.

Proof. By lemma A.3 and the fact that $K_d(\delta, \ell)$ is an increasing function of ℓ, we get

$$\begin{aligned} T_\ell &\le \frac{8d}{\epsilon_\ell^2} \log\left(\frac{1}{\epsilon_\ell}\right) \\ &\le 16d \, \frac{256 e^{16R_\ell} K_d^2(\delta, L) + 16R^2 e^{8R_\ell}}{R_\ell^2} \max_{\ell \in [L]} \log\left(\frac{1}{\epsilon_\ell}\right) \\ &= d \max_{\ell \in [L]} \log\left(\frac{1}{\epsilon_\ell}\right) O\left(\frac{e^{16R_\ell} K_d^2(\delta, L)}{R_\ell^2} + \frac{R^2 e^{8R_\ell}}{R_\ell^2}\right) \,. \end{aligned}$$

where the second inequality uses $(a+b)^2 \le 2a^2 + 2b^2$.

For the terms within the big-oh, once we sum over ℓ we can write

$$\begin{aligned} \sum_{\ell=1}^{L} \frac{e^{16R_\ell}}{R_\ell^2} &= \sum_{R_\ell > 1} \frac{e^{16R_\ell}}{R_\ell^2} + \sum_{R_\ell \le 1} \frac{e^{16R_\ell}}{R_\ell^2} \\ &\le e^{8R\lceil \log_2 R \rceil} + \frac{e^{16}}{R_L} \sum_{R_\ell \le 1} \frac{1}{R_\ell} \\ &\le e^{8R\lceil \log_2 R \rceil} + \frac{2e^{16}}{R_L^2} \,. \end{aligned}$$

And similarly

$$\sum_{\ell=1}^{L} \frac{e^{8R_\ell}}{R_\ell^2} \le e^{4R\lceil \log_2 R \rceil} + \frac{2e^8}{R_L^2} \,.$$

Putting them together gives

$$N_T(\mathcal{P}) = \sum_{\ell=1}^{L} T_\ell = d \max_{\ell \in [L]} \log\left(\frac{1}{\epsilon_\ell}\right) O\left(K_d^2(\delta, L) e^{8R\lceil \log_2 R \rceil} + e^{4R\lceil \log_2 R \rceil} R^2 + \frac{K_d^2(\delta, L) + R^2}{R_L^2}\right) \,,$$

as claimed. □

The following bound on the weighted cumulative regret is the logistic counterpart to Theorem A.5.

Theorem B.3. *For any pool realization $\mathcal{P}$, the weighted cumulative regret $R_T(\mathcal{P})$ of Algorithm 2 operating on a pool $\mathcal{P}$ of size T is bounded as*

$$R_T(\mathcal{P}) = d \max_{\ell \in [L]} \log\left(\frac{1}{\epsilon_\ell}\right) O\left(K_d^2(\delta, L) e^{8R\lceil \log_2 R \rceil} + e^{4R\lceil \log_2 R \rceil} R^2 + \frac{K_d^2(\delta, L) + R^2}{R_L}\right) \,.$$

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

Proof. We follow the same reasoning as in Theorem A.5. 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 study the weighted cumulative regret components

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

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

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

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

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

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

$$R_T(\mathcal{P}_L) \leq R_L|\mathcal{P}_L| < \frac{d}{4R_L} \ ,$$

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 $\operatorname{sgn}\langle\widehat{\mathbf{w}}, \mathbf{x}\rangle = \operatorname{sgn}\langle\mathbf{w}^*, \mathbf{x}\rangle$ 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 $|\langle\mathbf{w}_{\ell-1}, \mathbf{x}\rangle| \leq R_{\ell-1}$ which, combined with the assumption that $\mathcal{E}_{\ell-1}^\sigma$ holds, yields $|\langle\mathbf{w}^*, \mathbf{x}\rangle| \leq 2R_{\ell-1}$. Since $|\mathcal{Q}_\ell| = T_\ell$, we have

$$R_T(\cup_{\ell=1}^{L}\mathcal{Q}_\ell) \leq 2\sum_{\ell=1}^{L} T_\ell R_\ell$$

and Lemma A.3 allows us to write

$$R_T(\cup_{l=1}^{L}\mathcal{Q}_\ell) \leq 16d\sum_{l=1}^{L}\frac{R_\ell}{\epsilon_\ell^2}\log\left(\frac{1}{\epsilon_\ell}\right) = d\max_{\ell\in[L]}\log\left(\frac{1}{\epsilon_\ell}\right) O\left(\frac{e^{16R_\ell}K_d^2(\delta, L)}{R_\ell} + \frac{R^2e^{8R_\ell}}{R_\ell}\right) \ .$$

Similar to the argument in theorem A.4, we have

$$\begin{aligned}\sum_{\ell=1}^{L}\frac{e^{16R_\ell}}{R_\ell} &= \sum_{R_\ell>1}\frac{e^{16R_\ell}}{R_\ell} + \sum_{R_\ell\leq 1}\frac{e^{16R_\ell}}{R_\ell} \\ &\leq e^{8R}\lceil\log_2 R\rceil + e^{16}\sum_{R_\ell\leq 1}\frac{1}{R_\ell} \\ &\leq e^{8R}\lceil\log_2 R\rceil + \frac{2e^{16}}{R_L} \ .\end{aligned}$$

and

$$\sum_{\ell=1}^{L}\frac{e^{8R_\ell}}{R_\ell} \leq e^{4R}\lceil\log_2 R\rceil + \frac{2e^8}{R_L} \ .$$

Piecing together, we conclude that the total regret is bounded as

$$R_T(\mathcal{P}) = d\max_{\ell\in[L]}\log\left(\frac{1}{\epsilon_\ell}\right) O\left(K_d^2(\delta, L)e^{8R}\lceil\log_2 R\rceil + e^{4R}\lceil\log_2 R\rceil R^2 + \frac{K_d^2(\delta, L) + R^2}{R_L}\right) \ ,$$

thereby concluding the proof. □

As in the linear case, adding the ingredient of the low noise condition (2) helps us exploit the randomness in $\mathcal{P}$ to further bound from above the number of stages L in the logistic case.

Specifically, assume the low noise condition (2) holds for $f^*(\mathbf{x}) = \sigma(\langle \mathbf{w}^*, \mathbf{x} \rangle)$, for some unknown exponent $\alpha \geq 0$ and unknown constant $\epsilon_0 \in (0, 1]$. Similar to linear case we define the event

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

Then

$$\mathbb{P}\left(\bar{\mathcal{E}}^\sigma\right) \geq 1 - \delta \,, \tag{11}$$

for $M = \log_2 T$.

Lemma B.4. *Let ϵ_* be defined through (8), with $T > \frac{2}{3}d$. Assume both $\bar{\mathcal{E}}^\sigma$ and $\bigcap_{\ell=1}^L \mathcal{E}_\ell$ hold. Then the number of stages L of Algorithm 2 is upper bounded as*

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

where the O-notation only hides absolute constants.

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

$$d/2R_{L-1} \leq 2R_{L-1}|\mathcal{P}_{L-1}| \,.$$

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

We split the analysis into two cases. On one hand, when $2R_{L-1} > \epsilon_0$, this condition gives us directly

$$L \leq \log_2(\frac{R}{\epsilon_0}) + 2 \,.$$

On the other hand if $2R_{L-1} \leq \epsilon_0$, then given that $\bar{\mathcal{E}}^\sigma$ holds, $|\mathcal{T}^\sigma_{2R_{L-1}}|$ is upper bounded as

$$|\mathcal{T}^\sigma_{2R_{L-1}}| \leq 3\left(T(2R_{L-1})^\alpha + \log\left(\frac{M}{\delta}\right)\right) \,,$$

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

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

which resembles (8) with $2R_{L-1}$ here playing the role of ϵ^* therein. Then, from the definition of ϵ^* in (8) we immediately obtain $2R_{L-1} \geq \epsilon_*$, thus $L \leq \log_2(\frac{R}{\epsilon_*}) + 2$. Moreover, from (8) we see that $d/\epsilon_* \geq 3T\epsilon_*^{\alpha+1}$, which is equivalent to $\epsilon_* \leq (\frac{d}{3T})^{\frac{1}{\alpha+2}}$. Replacing this upper bound on ϵ^* back into the right-hand side of (8), dividing by d and multiply by R yields

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

which gives the claimed upper bound on L through $L \leq \log_2(\frac{R}{\epsilon_*}) + 2$. □

Corollary B.5. *Let $T > \frac{2}{3}d$. 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:*

$$N_T(\mathcal{P}) = C_{d,R}(\delta, T, \epsilon_0) O\left(\max\left\{d^{\frac{\alpha}{\alpha+2}} T^{\frac{2}{\alpha+2}}, \frac{d}{\epsilon_0^2}\right\} + \log^2\left(\frac{\log T}{\delta}\right) + de^{8R}\lceil \log_2 R\rceil\right),$$
$$R_T(\mathcal{P}) = C_{d,R}(\delta, T, \epsilon_0) O\left(\max\left\{d^{\frac{\alpha+1}{\alpha+2}} T^{\frac{1}{\alpha+2}}, \frac{d}{\epsilon_0}\right\} + \log\left(\frac{\log T}{\delta}\right) + de^{8R}\lceil \log_2 R\rceil\right),$$

where the O-notation hiding absolute constants and

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

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

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

Similar to Lemma B.4, we split the analysis into two cases depending on whether or not $2R_{L-1}$ is bigger than ϵ_0. If $2R_{L-1} \le \epsilon_0$, we have

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

therefore

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

Moreover, we have

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

and

$$\begin{aligned}\max_{\ell\in[L]} \log\left(\frac{1}{\epsilon_\ell}\right) &\le \log\left(\frac{16e^{4R}K_d(\delta, L) + 4Re^{2R}}{R_L}\right)\\ &= O\left(R + \log K_d(\delta, L) + \log\left(\frac{T}{d}\right) + \log\left(\frac{R\log T}{\delta}\right)\right)\\ &= O\left(R + \log\left(\frac{T}{\delta}\right)\right),\end{aligned}$$

where the last equality is because $T > \frac{2}{3}d$.

Plugging these bounds together back into factor

$$K_d^2(\delta, L)e^{8R}\lceil\log_2 R\rceil + e^{4R}\lceil\log_2 R\rceil R^2 + \frac{K_d^2(\delta, L) + R^2}{R_L^2}$$

of Theorem B.2 yields

$$\begin{aligned}&K_d^2(\delta, L)e^{8R}\lceil\log_2 R\rceil + e^{4R}\lceil\log_2 R\rceil R^2 + \frac{K_d^2(\delta, L) + R^2}{R_L^2}\\ &\quad= O\left(\left(K_d^2(\delta, L) + R^2\right)\left(e^{8R}\lceil\log_2 R\rceil + \frac{1}{R_L^2}\right)\right)\\ &\quad= O\left(\left(d\log\left(\frac{T}{\delta}\right) + R^2\right)\left(\left(\frac{T}{d}\right)^{\frac{2}{\alpha+2}} + \left(\frac{1}{d}\right)^{\frac{2\alpha+2}{\alpha+2}}\left(\frac{1}{T}\right)^{\frac{2}{\alpha+2}}\log^2\left(\frac{\log T}{\delta}\right) + e^{8R}\lceil\log_2 R\rceil\right)\right),\end{aligned}$$

where the last equality is due to the assumption that $T > \frac{2}{3}d$. Combining the above estimates gives

$$N_T(\mathcal{P}) = O\left(\left(d\log\left(\frac{T}{\delta}\right) + R^2\right)\left(R + \log\left(\frac{T}{\delta}\right)\right)\left(d^{\frac{\alpha}{\alpha+2}}T^{\frac{2}{\alpha+2}} + \log^2\left(\frac{\log T}{\delta}\right) + de^{8R}\lceil\log_2 R\rceil\right)\right)$$

A similar argument gives

$$R_T(\mathcal{P}) = O\left(\left(d\log\left(\frac{T}{\delta}\right) + R^2\right)\left(R + \log\left(\frac{T}{\delta}\right)\right)\left(d^{\frac{\alpha+1}{\alpha+2}}T^{\frac{1}{\alpha+2}} + \log\left(\frac{\log T}{\delta}\right) + de^{8R}\lceil\log_2 R\rceil\right)\right) .$$

If $2R_{L-1} > \epsilon_0$, then $\frac{1}{R_L} \leq \frac{4}{\epsilon_0}$. Applying these bounds into Theorem B.2 we get

$$\begin{aligned} N_T(\mathcal{P}) =& O\left(\left(d\log\left(\frac{T}{\delta}\right) + R^2 + \log\left(\frac{1}{\epsilon_0}\right)\right)\left(R + \log\left(\frac{T}{\delta\epsilon_0}\right)\right)\left(\frac{d}{\epsilon_0^2} + de^{8R}\lceil\log_2 R\rceil\right)\right) \\ =& \left(d\log\left(\frac{T}{\delta}\right) + R^2\right)\left(R + \log\left(\frac{T}{\delta}\right)\right)\left(1 + \log^2\left(\frac{1}{\epsilon_0}\right)\right) O\left(\frac{d}{\epsilon_0^2} + de^{8R}\lceil\log_2 R\rceil\right) . \end{aligned}$$

Similarly

$$R_T(\mathcal{P}) = \left(d\log\left(\frac{T}{\delta}\right) + R^2\right)\left(R + \log\left(\frac{T}{\delta}\right)\right)\left(1 + \log^2\left(\frac{1}{\epsilon_0}\right)\right) O\left(\frac{d}{\epsilon_0} + de^{8R}\lceil\log_2 R\rceil\right) .$$

Lastly, (11) and Lemma B.1 together yield

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

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. As in the linear case, we have

$$\begin{aligned} \mathcal{L}(\widehat{\mathbf{w}}) - \mathcal{L}(\mathbf{w}^*) &= \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\Big[\mathbb{1}\{y \neq \operatorname{sgn}(\langle\widehat{\mathbf{w}},\mathbf{x}\rangle)\} - \mathbb{1}\{y \neq \operatorname{sgn}(\langle\mathbf{w}^*,\mathbf{x}\rangle)\}\Big] \\ &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathcal{X}}\Big[\mathbb{E}_{y\sim\mathcal{D}_{\mathcal{Y}|\mathcal{X}}}\big[\mathbb{1}\{y \neq \operatorname{sgn}(\langle\widehat{\mathbf{w}},\mathbf{x}\rangle)\} - \mathbb{1}\{y \neq \operatorname{sgn}(\langle\mathbf{w}^*,\mathbf{x}\rangle)\}\big]\Big] \\ &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_\mathcal{X}}\Big[\mathbb{1}\{\operatorname{sgn}(\langle\widehat{\mathbf{w}},\mathbf{x}\rangle) \neq \operatorname{sgn}(\langle\mathbf{w}^*,\mathbf{x}\rangle)\}\,|2\sigma(\langle\mathbf{w}^*,\mathbf{x}\rangle) - 1|\Big] , \end{aligned}$$

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

$$\mathbb{1}\{\operatorname{sgn}(\langle\widehat{\mathbf{w}},\mathbf{x}\rangle) \neq \operatorname{sgn}(\langle\mathbf{w}^*,\mathbf{x}\rangle)\}\,|2\sigma(\langle\mathbf{w}^*,\mathbf{x}\rangle) - 1|$$

has the same form as the function $\phi(\widehat{\mathbf{w}},\mathbf{x})$ in Appendix C on which Theorem C.4 applies, with $\widehat{\epsilon}(\delta)$ therein replaced by the bound on $R_T(\mathcal{P})$ deriving from Corollary B.5. This allows us to conclude that with probability at least $1-\delta$

$$\mathcal{L}(\widehat{\mathbf{w}}) - \mathcal{L}(\mathbf{w}^*) = C_{d,R}(\delta,T,\epsilon_0)O\left(\max\left\{\left(\frac{d}{T}\right)^{\frac{\alpha+1}{\alpha+2}}, \frac{d}{T\epsilon_0}\right\} + \frac{\log\left(\frac{\log T}{\delta}\right) + de^{8R}\lceil\log_2 R\rceil}{T}\right) ,$$

as claimed in Theorem 5.1 in the main body of the paper.

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 C.1. *Let $a_1, \ldots, a_T$ be T arbitrary real numbers, and $\{\sigma_1, \ldots, \sigma_T\}$ 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 $\{\sigma_1, \ldots, \sigma_T\}$.

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 C.2. *Consider any given $S_T = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\} \in \mathbb{R}^d$, and let*

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

We have, when $T \geq d$,

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

The next result follows from a standard symmetrization argument.

Lemma C.3. *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 C.1. 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 C.2)} \\
&\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 C.1)} \\
&\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 C.4. *With the same notation and assumptions as in Lemma C.3, let $\widehat{\mathbf{w}} \in \mathbb{R}^d$ be a function of S_T such that*

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

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

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

Proof. Use the multiplicative Chernoff bound

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

and then apply Lemma C.3 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 C.5. *Suppose ξ is a σ-subgaussian random variable, then for any $\delta > 0$,*

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

Lemma C.6. *Let $A = [a_{ij}] \in \mathbb{R}^{m\times n}$ be a matrix. Suppose $\xi_1, \ldots, \xi_n$ are independent σ-subgaussian random variables. Then for any $\delta > 0$,*

$$\mathbb{P}\left(\|A\xi\|^2 > 2\sigma^2 \log\frac{2m}{\delta}\mathrm{tr}(AA^\top)\right) \leq \delta \,,$$

where $\xi = (\xi_1, \ldots, \xi_n)^\top$.

Proof. Consider

$$(A\xi)_i = \sum_{j=1}^n a_{ij}\xi_j \,,$$

the ith component of vector $A\xi$. Note that $(A\xi)_i$ is a $\sigma\sqrt{\sum_{j=1}^n a_{ij}^2}$-subgaussian random variable, by lemma C.5 we have

$$\mathbb{P}\left(\left|\sum_{j=1}^n a_{ij}\xi_j\right| \geq \sqrt{2\sigma^2 \sum_{j=1}^n a_{ij}^2 \log\frac{2m}{\delta}}\right) \leq \frac{\delta}{m} \,.$$

A union bound over i gives, with probability at least $1-\delta$,

$$\left(\sum_{j=1}^n a_{ij}\xi_j\right)^2 \leq 2\sigma^2 \sum_{j=1}^n a_{ij}^2 \log\frac{2m}{\delta} \,,$$

uniformly over $i = 1, \ldots, m$. Therefore, with probability at least $1-\delta$,

$$\|A\xi\|^2 = \sum_{i=1}^m \left(\sum_{j=1}^n a_{ij}\xi_j\right)^2 \leq 2\sigma^2 \log\frac{2m}{\delta}\sum_{i,j} a_{ij}^2 = 2\sigma^2 \log\frac{2m}{\delta}\mathrm{tr}(AA^\top) \,,$$

as claimed. □