Online Active Learning with Surrogate Loss Functions

Giulia DeSalvo
Google Research
New York, NY 10011
giuliad@google.com

Claudio Gentile
Google Research
New York, NY 10011
cgentile@google.com

Tobias Sommer Thune
University of Copenhagen
Copenhagen, DK
tobias.thune@gmail.com

Abstract

We derive a novel active learning algorithm in the streaming setting for binary classification tasks. The algorithm leverages weak labels to minimize the number of label requests, and trains a model to optimize a surrogate loss on a resulting set of labeled and weak-labeled points. Our algorithm jointly admits two crucial properties: theoretical guarantees in the general agnostic setting and a strong empirical performance. Our theoretical analysis shows that the algorithm attains favorable generalization and label complexity bounds, while our empirical study on 18 real-world datasets demonstrate that the algorithm outperforms standard baselines, including the Margin Algorithm, or Uncertainty Sampling, a high-performing active learning algorithm favored by practitioners.

1 Introduction

While unlabeled data is generated at increasing rates, supervised learning often requires large portions of this data to be annotated by human raters, which carry significant costs. Active learning seeks to significantly alleviate this problem through algorithms that attain comparable performance to passive learning (i.e., to full supervision) while using considerably fewer labels. Because of the ever increasing use of labeled data in modern machine learning approaches, this calls for the development of more and more effective active learning techniques.

In this paper, we are specifically interested in binary classification problems in the so-called streaming (or online) setting of active learning. In this setting, learning proceeds in a sequence of rounds, where in each round an unlabeled sample is received, and the learner decides on-the-fly whether or not to observe the associated binary label. Now, it is common practice, in active learning or machine learning in general, that even for binary classification problems, the loss used both for training and for evaluating statistical performance is not the zero-one loss, but some surrogate function thereof. This results in more tractable (e.g. convex) optimization problems in the training phase, but also in more calibrated metrics in the evaluation phase. For instance, we may be compelled to output estimates of class probabilities, say, the success rate of a given set of decisions, and thus deploy a maximum posterior probability estimator for such probabilities. Then, given a training set with labels $y \in \{\pm 1\}$ and if $\hat{p}(x)$ is an estimate of the conditional class probability $\mathbb{P}[y = 1 \,|\, x]$, a standard goal is to minimize the log-loss $\frac{1+y}{2} \log \frac{1}{\hat{p}(x)} + \frac{1-y}{2} \log \frac{1}{1-\hat{p}(x)}$. In these cases, the logistic link function $\hat{p}(x) = \frac{1}{1+e^{-h(x)}}$ is often adopted to model conditional class probabilities where h is some (unknown) function in a given hypothesis class. In this and many other situations that arise in practice, it is thus advisable to resort to surrogate loss functions for both training and evaluation.

The online active learning literature contains a number of proposed algorithms that admit theoretical guarantees for both generalization error and label complexity, that is, the expected number of label requests. The theoretical analyses found in these works are either tailored to the zero-one loss or to general surrogate losses, with much more emphasis historically placed on the former. The papers that

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

analyze zero-one loss introduce novel ideas tackling important theoretical questions in active learning, but the theory does not easily extend to general loss functions – see discussion in Henneke and Yang [2019]. Nevertheless, the algorithmic solutions we propose borrow several ideas from one such work, namely [Dasgupta et al., 2008]. In the agnostic setting, Dasgupta et al. [2008] developed an algorithm for the zero-one loss based on the disagreement between functions, that carefully constructs pseudo-labels, that is, weak labels generated by the algorithm itself. We call this algorithm DHM, after the name of the three authors.

For general surrogate losses, Beygelzimer et al. [2009] designed an algorithm based on constructing importance weighted predictors with theoretical guarantees in the agnostic setting. Cortes et al. [2019a,b, 2020] further developed these importance-weighted algorithms enhancing their guarantees and used them as building blocks for meta-algorithms. We refer to these algorithms as IWAL (Importance Weighted Active Learning).

On a different line of work, Henneke and Yang [2019] presents a theoretical analysis of an active learning algorithm that uses surrogate loss functions for binary classi cation. Among the theoretically-oriented contributions in active learning, this work appears to be the closest reference to our paper when it comes to motivations. Yet, it is fair to say that the authors make the very strong assumption that the Bayes optimal function for the surrogate loss at hand lies in the hypothesis class used for learning, and they explicitly state that this realizability condition cannot be relaxed without resorting to a signi cantly different approach. Moreover, unlike our work and that of IWAL-based algorithms, Henneke and Yang [2019] are only focusing on deriving bounds on the zero-one risk. And nally, their algorithm operates in a pool-based setting of active learning, where the whole set of unlabeled points is available to the learner, which is more exible than the streaming setting analyzed here.

In practice, the Margin Algorithm, or Uncertainty Sampling, of Lewis and Gale [1994] attains a substantial performance improvement compared to passive learning as well as other active learning algorithms, and is thus often favored by practitioners (e.g. Schohn and Cohn [2000], Tong and Koller [2001], Brinker [2003], Culotta and McCallum [2005], Joshi et al. [2009], Mussmann and Liang [2018]). Moreover, the margin algorithm is exible in that it can be used in both the pool and streaming setting. In the pool setting, several papers have recently tailored active learning algorithms to neural networks, always comparing their algorithm's empirical performance to that of the margin algorithm [Geifman and El-Yaniv, 2017, Gal et al., 2017, Savarese et al., 2018, Ducoffe and Precioso, 2018, Ash et al., 2020, Moon et al., 2020, Schroder and Niekler, 2020]. Overall, the margin algorithm is often found to be a competitive contender to this new collection of practical active learning algorithms. Additionally, two extensive empirical studies found that the margin algorithm outperforms many recent active learning algorithms [Yang and Loog, 2016, Chuang et al., 2019]. On the theoretical side, several authors developed active learning algorithms with theoretical guarantees for speci c classes of functions (e.g., linear separators) based on sampling along the margin [Dasgupta et al., 2005, Balcan et al., 2007, Balcan and Long, 2013, Awasthi et al., 2014, 2015, Zhang, 2018, Zhang et al., 2020], but the guarantees of these margin-based algorithms only hold under strong assumptions on the realizability of the problem. Our goal is thus to derive an algorithm that admits theoretical guarantees without relying on realizability assumptions and that, at the same time, attains in practice a performance that is better than that of the margin algorithm in the streaming setting.

1.1 Contributions

In this paper, we introduce a novel active learning algorithm, called ALPS (Actively Learning over Pseudo-labels for Surrogate losses), in the streaming setting for binary classi cation tasks that trains a model by optimizing a surrogate loss on the joint set of labeled and self-constructed pseudo-labeled points (Section 3). Crucially, we prove that ALPS admits theoretical guarantees with respect to the surrogate loss in the general agnostic setting while at the same time surpassing the training performance of the margin algorithm as well as other baselines.

More concretely, we prove theoretical guarantees for ALPS in terms of both generalization and label complexity, under assumptions on the interplay between the function class and the data distribution which are not captured by traditional low noise assumptions often formulated in active learning (Section 4). We show that ALPS is able to leverage pseudo-labels in a principled manner even though the loss function of interest is not the zero-one loss, but a surrogate loss thereof. We then complement our theoretical ndings with a thorough experimental investigation on 18 real-world

datasets using a class of neural networks, Multi-layer Perceptrons. We test our algorithm against the IWAL algorithm of Beygelzimer et al. [2009] since it admits theoretical guarantees for general loss functions, and against passive learning since we want to quantify the improvement over supervised methods. Additionally, we generate as competitors a pool of margin-based uncertainty samplers by varying their label request threshold, and show that on almost all datasets our algorithm outperforms the ex-post best among such algorithms (Section 5).

This paper follows the general research agenda of reducing the gap between theory and practice, which has been widening at a faster rate in the recent active learning literature. As outlined above, we try to bridge this disconnect by deriving an algorithm that not only works well in practice, but also admits solid theoretical guarantees. In doing so, we also introduce novel algorithmic ideas, which could independently lead to new directions of research.

1.2 Main Ideas

The main mechanism behind ALPS is to query the label of the current instance based on the disagreement between hypotheses and to construct pseudo-labels for the non-queried instances. The algorithm then trains a model on the joint set of labeled and pseudo-labeled data. The sample used for training will be unbiased with respect to marginal distribution on the instance space, but label noise will be introduced whenever the pseudo-label differs from the true (unrevealed) label. This noise affects both the generalization ability of the algorithm and its label complexity. Controlling this noise is not only crucial, but also quite challenging, for it relies on assessing the expected reliability of the pseudo-labels.

In order to control this noise, we introduce the novel idea of using requester functions: ALPS learns to query labels on instances with unreliable (that is, potentially noisy) pseudo-labels by using a requester function chosen to minimize an importance-weighted estimate of the noise. For the sake of our results, the requester functions are assumed to lie in some generic function class. One natural example of this class is that of margin-based functions, meaning functions that request along the margin of a given prediction model. In this case, our algorithm is related to the margin algorithm but, unlike the theoretical guarantees for the margin algorithm (e.g. Balcan et al. [2007]), we do not rely on speci c distributional assumptions on the instance space. The requester functions are reminiscent of the abstention functions introduced in Cortes et al. [2016], but serve an entirely different purpose in active learning.

ALPS can be seen as an extension of the DHM algorithm [Dasgupta et al., 2008] to surrogate loss functions, which seeks to address the open question posed by Dasgupta et al. [2008] of whether there are active learning algorithms that only require solving tractable optimization problems in agnostic scenarios. At the same time, these two algorithms differ on how they treat the label noise introduced by pseudo-labeling. In DHM, the label noise can be easily bypassed by a simple property of the zero-one loss. In contrast, for general surrogate loss functions, this noise is unavoidable, and if not controlled as is done in the ALPS algorithm, the learned hypothesis can be arbitrarily far from the best-in-class.

2 Preliminaries and Notation

We consider an active learning framework in the streaming setting for binary classi cation. Learning proceeds in a sequence of rounds. In each round n, the learner receives from the environment an instance (or feature vector) x_n from an instance (or feature) space X. Based on what has been observed so far, the learner can then decide whether or not to request the true label associated with x_n. Label y_n is assumed to lie in the output space $Y = \{-1, 1\}$. The pairs $(x_1, y_1), (x_2, y_2), \ldots$ are drawn i.i.d. from a joint distribution D over $X \times Y$.

In order to evaluate the statistical performance of the learner, we consider any bounded surrogate loss function $\ell : \mathbb{R} \times Y \to [0, B]$ that upper bounds the zero-one loss, and let $\ell_B(t, y) = \ell(t, y)/B$ be a $[0, 1]$-normalized version of this loss.

Let H denote a hypothesis class of functions $X \to \mathbb{R}$. For simplicity of exposition, we assume that H is a nite class, but our analysis can be extended to classes with nite VC dimension by standard covering arguments. The true (expected) risk of hypothesis $h \in H$ is de ned as $\mathrm{err}(h) = \mathbb{E}_{(x,y) \sim D}[\ell(h(x), y)]$ and $h^* = \operatorname{argmin}_{h \in H} \mathrm{err}(h)$ denotes the best-in-class hypothesis.

Moreover, we denote by $\mathrm{err}(h; Z) = \frac{1}{n} \sum_{(x,y) \in Z} \ell(h(x), y)$ the empirical risk of h over a training set $Z = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$.

Given round $n \geq 1$, we denote by T_n the set of labeled data points $\{(x_s, y_s)\}$ from rounds $s \in [n] := \{1, \ldots, n\}$ where the label was requested by the learning algorithm, and denote by S_n its complement, that is, the set of labeled data points $\{(x_s, y_s)\}$ from rounds $s \in [n]$ where the label was not requested. Unlike T_n, the set S_n is not fully known to the learner. On those rounds, the learner replaces the true unobserved label y_s by pseudo-label $\hat{y}_s$, which is a suitable proxy to y_s generated by the learner itself. We denote by $\hat{S}_n$ the set containing data points $\{(x_s, \hat{y}_s)\}$ from rounds $s \in [n]$ where the label is not requested, and the true label gets replaced by the corresponding pseudo-label.

For notational convenience (and unless otherwise specified), we set $\hat{y}_s = y_s$ in rounds s where the true label is known to the learner. Then the following two short-hands are used for empirical risk up to round n using ground-truth labels and the empirical risk up to round n using pseudo-labels:

$$\mathrm{err}_n(h) = \mathrm{err}(h; S_n \cup T_n) = \frac{1}{n} \sum_{s=1}^{n} \ell(h(x_s), y_s), \quad \widehat{\mathrm{err}}_n(h) = \mathrm{err}(h; \hat{S}_n \cup T_n) = \frac{1}{n} \sum_{s=1}^{n} \ell(h(x_s), \hat{y}_s).$$

In the sequel, we will also need the following definition of consistency of hypotheses $h \in H$.

Definition 1. We say that $h \in H$ is *consistent* with a labeled data point $(x, y) \in X \times Y$ if $\mathrm{sgn}(h(x)) = y$ and, by extension, that it is consistent with a set of labeled data points Z if it is consistent with all data points in Z. Two hypotheses $h, h' \in H$ are consistent with one another on a set of (unlabeled) points if $\mathrm{sgn}(h(x)) = \mathrm{sgn}(h'(x))$ for all x in that set of points.

At each round n, the learner is compelled to output a hypothesis $h_n \in H$, and we seek to bound its true risk, $\mathrm{err}(h_n)$, in terms of the true risk, $\mathrm{err}(h^*)$, of the best-in-class hypothesis, as a function of the total number of streamed points n thus far. In addition, we want a bound on the label complexity of the learner, that is, the number of labels requested by the learner during the course of training. This bound should be provably smaller than n, which is the label complexity of a passive learner observing all labels. Further ancillary definitions will be introduced in later sections.

3 The ALPS Algorithm

At a high level, ALPS either requests the label of an instance vector x_n or assigns it a pseudo-label at each round n. The algorithm then selects a model h_n that is consistent on the pseudo-labeled points, $\hat{S}_n$, and that minimizes the empirical estimate of the risk, $\widehat{\mathrm{err}}_n(h)$, which is defined in terms of pseudo-labeled and labeled points, $\hat{S}_n \cup T_n$, processed thus far. See pseudo-code in Algorithm 1.

More concretely, the learned hypothesis at round n is $h_n = \mathrm{LEARN}(\hat{S}_n, \hat{S}_n \cup T_n)$ where:

Definition 2. For sets of labeled data points S and Z, $\mathrm{LEARN}(S, Z)$ returns a hypothesis in H that minimizes the empirical risk $\mathrm{err}(h; Z)$ on the second argument Z, and is consistent with all labeled data points in the first argument S, where consistency is defined in Definition 1. LEARN raises a flag if no such hypothesis is found.

In the LEARN procedure, we only need to consider hypotheses consistent with pseudo-labeled points, $\hat{S}_n$, because we can show that with high probability, the sign of the best-in-class, $\mathrm{argmin}_{h \in H} \mathrm{err}(h)$, matches the sign of the pseudo-labeled points.

In order to decide whether to query the label of x_n, ALPS leverages the disagreements between hypotheses $h_{+1}, h_{-1} \in H$ returned by the LEARN procedure, where the label of the current instance x_n is assumed to be either +1 for learning hypothesis h_{+1} and -1 for learning hypothesis h_{-1}. Ideally, the algorithm would use the empirical difference $\mathrm{err}_n(h_{+1}) - \mathrm{err}_n(h_{-1})$ based on ground-truth labels to assess this disagreement, but because not all labels at n have been disclosed to the algorithm, it instead uses the pseudo-labeled version $\widehat{\mathrm{err}}_n(h_{+1}) - \widehat{\mathrm{err}}_n(h_{-1})$, which is indeed observable. The algorithm compares this difference to a slack term $\tilde{\epsilon}_{n,n}$ derived from our theoretical analysis. See Appendix A for all slack term definitions. If the loss difference is smaller than the

[1] The sign of the pseudo-label $\hat{y}$ is determined by the condition $\widehat{\mathrm{err}}_{n-1}(h_{\hat{y}}) - \widehat{\mathrm{err}}_{n-1}(h_{-\hat{y}}) > \tilde{\epsilon}_{n-1}$ and whether $h_{\hat{y}}$ exist for $\hat{y} \in \{-1, 1\}$. E.g., $\hat{y} = +1$ if $\widehat{\mathrm{err}}_{n-1}(h_{-1}) - \widehat{\mathrm{err}}_{n-1}(h_{+1}) > \tilde{\epsilon}_{n-1}$ or if h_{-1} does not exist.

Algorithm 1: Actively Learning over Pseudo-labels for Surrogate losses (ALPS)

Input: LEARN(S, Z), Hypothesis class $\mathcal{H}$, requester class $\mathcal{R}$, and slack terms $\tilde{\gamma}_n$; γ^0_n.

$\hat{S}_0 = \ ; \ ; T_0 = \ ; \ ; F_1 = \mathcal{H} \quad \mathcal{R}$;

for $n = 1, 2, \ldots, N$ **do**

  Receive feature vector x_n;

  For $\hat{y} = \quad 1$, let $h_{\hat{y}} =$ LEARN$(\hat{S}_{n\ 1} [\{(x_n, \hat{y})\}, \hat{S}_{n\ 1} [\{(x_n, \hat{y})\} [T_{n\ 1})$;

  Define version space F_n and bias p_n:

$$F_n = \{(h, r) \in F_{n\ 1} : \min_{(h^0, r^0) \in F_{n\ 1}} l_{n\ 1}(h^0, r^0) \quad l_{n\ 1}(h, r) \quad \gamma^0_{n\ 1}\}$$

$$p_n = \max_{(h,r),(h^0,r^0) \in F_n} \max_{y \in Y} \ell_B(y, h(x_n)) \mathbb{I}\{r(x_n) \quad 0\} \quad \ell_B(y, h^0(x_n)) \mathbb{I}\{r^0(x_n) \quad 0\}$$

  $Q_n \quad$ Bernoulli(p_n);

  if $\widehat{\mathrm{err}}_{n\ 1}(h_{\hat{y}}) \quad \widehat{\mathrm{err}}_{n\ 1}(h_{\hat{y}}) > \tilde{\gamma}_{n\ 1}$ (or if no such $h_{\hat{y}}$ is found) for some $\hat{y} \in \{ 1\}$ AND $Q_n = 0$ **then**

    $r_n = \operatorname{argmin}_{r \in R_n} E[\mathbb{I}\{r(x) > 0\}]$, where

$$R_n = \{r : (h, r) \in F_n \text{ and } h \text{ consistent with } \hat{S}_{n\ 1} [\{(x_n, \hat{y})\}\}$$

    if $r_n(x_n) > 0$ **then**

      Request y_n and update $\hat{S}_n = \hat{S}_{n\ 1}, T_n = T_{n\ 1} [\{(x_n, y_n)\}$;

    else

      Do not request y_n and use pseudo-label $\hat{y}$ to update $\hat{S}_n = \hat{S}_{n\ 1} [\{(x_n, \hat{y})\}, T_n = T_{n\ 1}$;

    end if

  else

    Request y_n and update $\hat{S}_n = \hat{S}_{n\ 1}, T_n = T_{n\ 1} [\{(x_n, y_n)\}$;

  end if

  $h_n =$ LEARN$(\hat{S}_n, \hat{S}_n [T_n)$;

end for

Output $h_N =$ LEARN$(\hat{S}_N, \hat{S}_N [T_N)$;

slack, then the label is requested because the sign of best-in-class cannot be inferred. Otherwise, a pseudo-label is constructed. In more detail, the sign of the pseudo-label is determined by the condition $\widehat{\mathrm{err}}_{n\ 1}(h_{\hat{y}}) \quad \widehat{\mathrm{err}}_{n\ 1}(h_{\hat{y}}) > \tilde{\gamma}_{n\ 1}$ and whether $h_{\hat{y}}$ exist for $\hat{y} \in \quad 1$. Note that by this construction, the LEARN procedure can always return a consistent hypothesis on all pseudo-labeled points.

To understand the next steps in the ALPS algorithm, we relate the ground-truth empirical difference $\mathrm{err}_n(h) \quad \mathrm{err}_n(h^0)$ to its the pseudo-labeled counterpart $\widehat{\mathrm{err}}_n(h) \quad \widehat{\mathrm{err}}_n(h^0)$. To this effect, we define the difference of difference of losses $A_n(h, h^0)$ between two hypotheses $h, h^0 \in H$ at time n as

$$A_n(h, h^0) := \mathrm{err}_n(h) \quad \mathrm{err}_n(h^0) \quad \widehat{\mathrm{err}}_n(h) \quad \widehat{\mathrm{err}}_n(h^0) :$$

This idea is also adopted by Dasgupta et al. [2008] in their analysis of the DHM algorithm. Yet, because they only deal with zero-one loss, in their case $A_n(h, h^0) = 0$ for consistent h, h^0 even when the true label does not match the pseudo-label. This easily follows from the fact that if $\mathrm{sgn}(h)$ and $\mathrm{sgn}(h^0)$ agree on instance x, then[2] $\mathbb{I}\{\mathrm{sgn}(h(x)) \quad y\} \quad \mathbb{I}\{\mathrm{sgn}(h^0(x)) \quad y\} = \mathbb{I}\{\mathrm{sgn}(h(x)) \quad \hat{y}\} \quad \mathbb{I}\{\mathrm{sgn}(h^0(x)) \quad \hat{y}\}$, irregardless of whether $y = \hat{y}$ or not.

In the surrogate loss setting analyzed here, whenever the pseudo-labels do not match their corresponding true labels, the term $A_n(h, h^0)$ is typically non-zero even for consistent h, h^0. As a consequence, the magnitude of $A_n(h, h^0)$ must be carefully controlled since, otherwise, it would break both generalization and label complexity bounds as well as the guarantee that the best-in-class is consistent with $\hat{S}_n$. In the sequel, we call A_n the *noise term*, as it captures the noise introduced during training due to the discrepancy between pseudo-labels and true labels.

[2] Throughout, $\mathbb{I}\{\}$ is the indicator function of its argument.

To control the noise term, we introduce the idea of using requester functions. Specifically, we consider a set $\mathcal{R}$ of functions $r : X \to \mathbb{R}$, where $r(x) > 0$ means that the label of x is requested by r, and otherwise it is not requested. At each round n, the algorithm picks a requester function r_n out of the set $\mathcal{R}$, and the algorithm's final condition of whether or not to request the label is decided by the sign of $r_n(x_n)$. The requester function r_n thus works as the final gatekeeper of whether to use the pseudo-label.

Notice that by definition the noise term A_n is non-zero when the true labels do not match the pseudo-labels, $y_s \neq \hat{y}_s$ for some $s \in [n]$. By construction, since the final condition of whether to use a pseudo-label in round s is determined by $\hat{y}_s$, it follows that $\mathbf{If}\{y_s \neq \hat{y}_s\} = \mathbf{If}\{y_s \neq \hat{y}_s\}\mathbf{If}\{r_s(x_s) > 0\}$. In addition, for any h consistent with $\hat{S}_s$, it holds that $\hat{y}_s = \mathrm{sgn}(h(x_s))$, and since the surrogate loss ℓ upper bounds the zero-one loss, we have that for any h consistent with $\hat{S}_s$,

$$\mathbf{If}\{y_s \neq \hat{y}_s\}\mathbf{If}\{r_s(x_s) > 0\} \leq \ell(h(x_s), y_s)\mathbf{If}\{r_s(x_s) > 0\} = \ell_B(h(x_s), y_s)\mathbf{If}\{r_s(x_s) > 0\}. \quad (1)$$

The above can thus be used as an upper bound of the noise.

To estimate this upper bound of A_n, the algorithm constructs an unbiased importance-weighted estimate $I_n(h, r)$, of $\mathbb{E}[\ell_B(h(x), y)\mathbf{If}\{r(x) > 0\}]$ as follows:

$$I_n(h, r) = \frac{1}{n}\sum_{s=1}^{n} \frac{Q_s}{p_s(x_s)}\ell_B(h(x_s), y_s)\mathbf{If}\{r(x_s) > 0\},$$

where Q_s is a Bernoulli random variable with bias $p_s(x_s)$ and where $p_s(\cdot)$ is a (random) function only depending on the past history $\{(x_{s'}, y_{s'}), Q_{s'}\}_{s'=1}^{s-1}$. The algorithm will request the label y_s whenever $Q_s = 1$ to construct this unbiased estimate. In order not to request the label ($Q_s = 1$) too frequently, $p_s(\cdot)$ is based on a shrinking version space F_s that contains only the pairs (h, r) whose empirical estimate $I_n(h, r)$ is close to the smallest one.

Given these estimates of $\mathbb{E}[\ell_B(h(x), y)\mathbf{If}\{r(x) > 0\}]$, the algorithm chooses a requester function r_n with the smallest request rate $\mathbb{E}[\mathbf{If}\{r(x) > 0\}]$ from a set of requester functions in the version space F_n. In this way, the algorithm ensures that the noise term A_n is kept small. This is due to the fact that the version space F_n contains pairs of functions (h, r) that are provably close to those minimizing $\mathbb{E}[\ell_B(h(x), y)\mathbf{If}\{r(x) > 0\}]$, combined with the fact that $\ell_B(h(x_n), y_n)\mathbf{If}\{r_n(x_n) > 0\}$ is an upper bound on the noise at round n via (1).

Putting the above together, ALPS will request the label y_n of x_n whenever one of the following three conditions hold: 1) the loss differences of h_n and h'_n is smaller than the slack term, in which case the revealed label is used as we cannot infer the sign of A_n and cannot construct a pseudo-label 2) $Q_n = 1$, in which case the revealed label is used for the empirical estimates $I_n(h, r)$, 3) $r_n(x_n) > 0$, in which case the revealed label is used to control the noise term A_n introduced by pseudo-labeling. In all cases, the revealed labels are used in LEARN to find the best model at each round.

Notice that, for the sake of argument, we have assumed in the above description to have access to a large unlabeled pool of examples so as to construct a convenient set of requester functions, that is, to estimate $\mathbb{E}[\mathbf{If}\{r(x) > 0\}]$ accurately. Yet, our algorithm and its associated analysis can be generalized straightforwardly to the case when such a pool is available only in a streaming way. Moreover, in our experiments in Section 5, we do not use an unlabeled pool to generate requester functions or to estimate the requesting rate. Instead, we define the set of requester functions to be margin-based functions with varying thresholds, and then estimate the requesting rate based on the data processed thus far. See Section 5 for details.

4 Theoretical Guarantees

We now present generalization and label complexity guarantees of our algorithm. These bounds will be in terms of the best-in-class hypothesis, whose risk $\mathrm{err}(h^*)$ will be denoted by ν. For these guarantees, we consider a class $\mathcal{R}$ of requester functions that satisfies the following:

Assumption 1. There exists $r^* \in \mathcal{R}$ and a constant $C > 0$ such that $\mathbb{E}[\ell(y, h^*(x))\mathbf{If}\{r^*(x) > 0\}] = 0$ and $\mathbb{E}[\mathbf{If}\{r^*(x) > 0\}] \leq C\nu$.

If given access to a large unlabeled pool of data, we can always construct a function r such that $\mathbb{E}[\ell(y, h(x))\mathbf{If}\{r(x) > 0\}] = 0$ for any h, since r can always be augmented in such a way to request

more frequently (that is, making $r(x) > 0$ for most x). Moreover below, we will further relax the first constraint to $E[\ell(y, h(x)) \mid r(x) \geq \rho] \leq \varepsilon_0$ for some $\rho > 0$. However, unless more information about the structure of the problem is known, proving that there exists a function in the set $\mathcal{R}$ such that $E[\mathbb{I}\{r(x) > 0\}] \leq C$ holds with a favorable (i.e., reasonably small) value of C is more difficult. Nevertheless, unfavorable C values, just like unfavorable disagreement coefficients, only impact the label complexity bound and not the generalization guarantee. After the theorems below, we present several natural examples of favorable C values.

Note that Assumumption 1 does not dictate conditions on the true conditional class probability, $\eta(x) = P[y = 1 \mid x]$. Hence, we are not relying on low-noise assumptions often adopted in active learning in realizable scenarios. Specifically, our assumption cannot be viewed as a surrogate loss counterpart to the Tsybakov [Mammen and Tsybakov, 1999, Tsybakov, 2004] low noise condition in the realizable setting [Castro and Nowak, 2008, Hanneke, 2011, Koltchinskii, 2010, Dekel et al., 2012] nor as an incarnation of the Bernstein condition, often invoked in statistical learning settings to achieve fast rates in both passive and active learning [Massart and Nedelec, 2006, Bartlett et al., 2006, Koltchinskii, 2006, Van Erven et al., 2015, Henneke and Yang, 2019].

In general, the richer the $\mathcal{R}$ class is, the easier Assumption 1 is satisfied with a small C. At the same time, since ALPS is learning over $\mathcal{H}$ and $\mathcal{R}$ simultaneously, the complexity of $\mathcal{R}$ affects both generalization bound and label complexity, as per usual in learning guarantees. Below, we first present the generalization bound of the ALPS algorithm. Here and in Theorem 2, notation $\tilde{O}$ is hiding logarithmic factors in n, $1/\delta$, and $|\mathcal{H}|$.

Theorem 1. Under Assumption 1, for any $\delta > 0$, with probability at least $1 - \delta$, for any $n > 0$,

$$\mathrm{err}(h_n) \leq \nu + \tilde{O}\left(\sqrt{\frac{\theta}{n}} + \frac{1}{n}\right),$$

where h_n is the hypothesis computed by ALPS in round n.

This guarantee states that the risk of the hypothesis returned by the algorithm converges to the best-in-class hypothesis at a rate that matches that of standard supervised learning, despite the fact that ALPS uses less labeled points. The analysis used to derive this theorem proves that the noise introduced by the pseudo-labels is, in fact, controlled by the learned requester function so that the above bound can be attained. In passing, we observe that if the loss is convex and the Bayes optimal hypothesis is in the class $\mathcal{H}$, then by using standard consistency techniques [Zhang, 2004, Bartlett et al., 2006] in conjunction with Theorem 1 above, we could also attain a bound on the excess risk of the zero-one loss.

The next theorem shows a bound on the expected number of labels the algorithm requests. This bound will be in terms of a notion of disagreement coefficient derived from the one in [Hanneke, 2007]. Define a metric ρ on the space of hypotheses $\mathcal{H}$ as $\rho(h, h') := E_{(x,y)\sim D}\left|\ell(h(x), y) - \ell(h(x), y)\right|$. Using this metric, we consider the ball around the best-in-class h^*: $B(h^*, r) = \{h : \rho(h, h^*) \leq r\}$. Then, let $\theta := \sup_{r > 0} P_x[\exists h \in B(h^*, r) : \mathrm{sgn}(h(x)) \neq \mathrm{sgn}(h^*(x))] / r$ be the disagreement coefficient of $\mathcal{H}$ (for the given distribution D over $X \times Y$).

Theorem 2. Under Assumption 1, for any $\delta > 0$, with probability $1 - \delta$, the label complexity of the ALPS algorithm at time $n > 0$ is bounded as $\tilde{O}(\theta(n(\nu + \rho) + C))$.

Disregarding constants ν and C, the above bound is $\tilde{O}(\theta n)$. This label complexity guarantee of ALPS in conjunction with its generalization bound matches the known lower bound from Beygelzimer et al. [2009] up to constants and is therefore optimal.

In Assumption 1, we considered the case where $E[\ell(y, h^*(x)) \mid r(x) \geq \rho] = 0$ which is easy to satisfy if we have a rich enough class $\mathcal{R}$ with functions that request more frequently. Nevertheless, we can relax this assumption if desired as follows:

Assumption 2. There exists $r \in \mathcal{R}$ and constants $C > 0$ and $\rho > 0$ such that $E[\ell(y, h^*(x)) \mid r(x) \geq \rho] \leq \nu$ and $E[\mathbb{I}\{r(x) > 0\}] \leq C$.

Under this assumption, the label complexity guarantee in Theorem 2 remains unchanged while an α_K term is added to the generalization bound in Theorem 1 when ALPS is run with slack $+\alpha_K$. By definition, α_K is at most ν and if $\alpha_K \leq \nu$, then this term has basically no effect on generalization.

There are several natural instances where the condition stated in Assumption 1 and Assumption 2 are satisfied with favorable C values. Consider for example a scenario where along the margin of the

Figure 1: Left: In yellow is a margin requesting region $\{x \in X : |h(x)| < \}$, where the hypothesis h, depicted by the black line, admits a conditional error of, say, 1/2. Middle: A tapered sigmoid function that equals 1 or 0 outside [the ;] interval. Right: An example of a requesting region (in yellow) that is not margin-based and where Assumption 2 holds with $C = 1$.

best-in-class hypothesis, there are hard-to-classify examples, while further away the best-in-class hypothesis mostly classifies the examples correctly. See Figure 1 for an illustration where there is such a margin around the best-in-class. This example is often used to explain why the popular margin algorithm works since this algorithm queries the label of points close to the classification surface. In our case, letting $g(x) = |h(x)|$, it holds that $E[\ell(y; h(x)) \mid |h(x)| < g] \leq$ for a small and $E[\mathbb{I}\{|h(x)| < g\}] = \frac{E[\ell(y;h(x)) \mid |h(x)| <]}{E[\ell(y;h(x)) \mid |h(x)| <]}$. If the expected loss of h conditioned on the margin region is high, say $1/2$, then it follows that $E[\mathbb{I}\{r(x) > 0\}] = E[\mathbb{I}\{|h(x)| < g\}] \leq 2$, so that Assumption 2 is satisfied with $C = 2$. Specifically, in Figure 1, it holds that $C = 2$ and $= 2/50$.

When h performs well outside the margin, then Assumption 2 holds with a small while if it correctly classifies all points outside the margin, then $= 0$ and Assumption 1 holds. The $= 0$ case is easily satisfied by generalized linear models $P(y = 1 \mid x] = (h(x))$, where is a tapered sigmoid function. Note that Assumption 1 is not placing specific restrictions on the marginal distribution over X, other than $E[\mathbb{I}\{|h(x)| < g\}] = 2$. This does not imply conditions on (x) when x falls in the region $|h(x)| <$ and hence we are not relying on realizability assumptions.

With any given class of hypotheses H, one can always associate the margin-based requester function class $R = \{r : r(x) = \mathbb{I}\{|h(x)| < \}; 0; h \in H\}$. In this case, ALPS can be seen as trying to simultaneously approximate h and the best threshold . The resulting algorithm turns out to be related to margin-based approaches since ALPS will seek to query the label of these more difficult points along the margin, but ALPS finds this optimal pair by using a different approach.

At the same time, we would like to stress that, in our framework, the function class R does not be restricted to margin-based requesters: if there exists a small region $\{x \in X : r(x) > 0\}$, for some function $r : X \to R$ such that most of the loss incurred by the best-in-class h is coming from this region, and R is rich enough to contain such function, then Assumption 1 and Assumption 2 hold. For an illustration of this situation, see Figure 1 on the right.

4.1 Comparisons to IWAL and DHM

The IWAL algorithm of Beygelzimer et al. [2009] constructs importance weighted estimates of the expected loss and uses them to select the prediction function. In contrast, ALPS leverages importance weighted estimators to select requester functions that minimize the noise term. Thus, even though both algorithms construct importance weighted estimators, they serve entirely different purposes. Both IWAL and ALPS attain generalization and label complexity bounds that are effectively of the same order, but our empirical results show that IWAL considerably underperforms compared to other active learning algorithm on all 18 datasets we tested. These empirical results are consistent with other authors' findings (e.g., Figure 2 in Cortes et al. [2019b] and Figure 3 in Cortes et al. [2020]).

Even though ALPS can be seen as a generalization of DHM [Dasgupta et al., 2008] to surrogate loss functions, there are several key differences. The noise introduced by the pseudo-labels does not affect DHM since by definition $A_n(h; h') = 0$ for the zero-one loss. Thus, DHM does not resort to requester functions or other techniques to deal with unreliable pseudo-labels. Moreover, unlike in DHM, the second argument in the LEARN$(\cdot, \cdot)$ subroutine over which ALPS minimizes errors is $\hat{S}_n \cup T_n$, not just T_n. For the zero-one loss, minimizing the error over $\hat{S}_n \cup T_n$ or only T_n is equivalent since hypothesis consistent on $\hat{S}_n$ admit zero loss over the points $\hat{S}_n$. Note that ALPS's pseudo-code

Figure 2: The plots on the left show the performance of ALPS, IWAL, the ex-post best margin algorithm, and passive learning with 4-layer networks for far and with 3-layer networks for ijcnn . For the 4-layer networks for ifar , the plot on the right shows the margin algorithms with thresholds in $_m$ ordered from smallest to largest threshold in where Margin-1 corresponds to the smallest threshold value in $_m$, Margin-2 to the second smallest, etc. As a function of the number of observed points (or rounds), the plots show the mean over the 50 repetitions and twice its standard error of the number of requested labels as well as the accuracy and logistic loss on the test set of the model returned by each algorithm.

reduces to DHM's when is the zero-one loss and is set to include only the never-requesting function, that is, $_r(x) = 0$ for all x, in which case the importance-weighted estimates $_m(h; r)$ are not constructed. We then attain the same guarantees as DHM's without resorting to Assumption 1 or Assumption 2.

5 Empirical Results

In this section, we present our experimental results in the streaming active learning setting that test the ALPS algorithm, the IWAL algorithm, the margin algorithm (or uncertainty sampling), and a passive learning algorithm, which requests the label of all points and nds the hypothesis that attains the smallest empirical loss. We compare to IWAL since it is a principled algorithm in the general agnostic scenario with strong theoretical guarantees. We test the margin algorithm since even though its theoretical guarantees do not hold in general, it admits a strong performance in practice. We run a passive learning algorithm to demonstrate the bene t of ALPS over standard supervised learning.

We tested these algorithms on 18 publicly available datasets where we used the logistic loss as the surrogate loss and used feedforward arti cial neural networks as our model class. Speci cally, we used the Multi-layer Perceptron algorithm in the scikit-learn library and ran two diverse network architectures, one with 3-layers and another with 4-layers. For each type of neural network, we constructed a diverse nite set of hypotheses by pre-training on small random subsets while varying the l_2 regularization parameter and initial weights. For each experiment, we randomly shuf ed the dataset, generated the nite hypothesis set, and ran all the algorithms. We repeated the experiment 50 times and averaged the results. See Appendix D for details on the datasets and model class settings.

Recall that the margin algorithm in the streaming setting rst selects the hypothesis with the smallest empirical loss and then requests the label of a point if $|P_h[y = +1 \mid x] - P_h[y = -1 \mid x]| \le$, where $P_h[y = -1 \mid x]$ is the model h's (estimated) conditional probability of labeling the given point x either +1 or -1. We ran nine instances of the margin algorithm for all thresholds in $_m$ where the set of thresholds $_m$ was chosen as a result of a tuning procedure that consisted of a standard grid search and zooming in. For ALPS, the requester bias is also de ned by margin-based functions as $_r(x) = \mathbb{I}\{|P_h[y = +1 \mid x] - P_h[y = -1 \mid x]| \le \}$ for each hypothesis in $h \in H$ and each $\in _r$. Note that we run ALPS with = 0 for this class R. Unlike $_m$, the set $_r$ was not tuned since ALPS learns the best threshold in . For details on $_m$ and $_r$, please see Appendix D.

To evaluate the algorithm performance, both accuracy and logistic loss are of interest in different applications as explained in the introduction. Thus, Figure 2 shows the accuracy and logistic loss on

the test set of the model returned by each algorithm at each round. Figure 2 also plots the number of points labeled by each algorithm as a function of the number of unlabeled points (or rounds).

Notice, in particular, that for the passive learning algorithm, since the data is streamed in a random order the value of the curves at time step t can be interpreted as running the algorithm up until we gather t many queries on randomly drawn points. So, for instance, at round $t = 200$ of the passive curve (orange line) for the ijcnn dataset, we effectively randomly sample and reveal the label of 200 points out of the training set, choose a model that corresponds to an empirical risk minimizer over these 200 points, and evaluate this model's accuracy on the test set. The passive learning curve is then used as a baseline comparator of the active learning algorithms. That is, the best performing active learning algorithm is the one that attains an accuracy and logistic loss curve that is on par or better than that of passive learning while requesting the fewest number of labels.

Figure 2 on the right shows that the accuracy and logistic loss curves of the margin algorithm varies with the threshold. In almost all datasets, there exists thresholds whose corresponding margin algorithm admits an accuracy curve that is below that of passive learning and as the threshold increases, the accuracy curve improves eventually matching that of passive learning. This indicates that a good set of threshold values was tested, since we seek an algorithm that requests the fewest number of points (i.e. having a small threshold) with an accuracy curve on par to that of passive learning. We then pick the ex-post best margin algorithm with smallest threshold that admits an learning curve area with respect to accuracy that is within one standard error of the passive learning curve area. If no algorithm admits such a curve, we pick the margin algorithm with the largest learning curve area. Choosing the ex-post best margin algorithm in this way and tuning γ_{min} gives the margin algorithm an unfair advantage, since both of these processes use the learning curves based on the revealed labels. Nevertheless, it provides an upper bound on the best performance of these uncertainty samplers.

The left-most plots of Figure 2 show that, for both cifar and ijcnn datasets, the ALPS algorithm performs better than all baseline methods since out of the algorithms that attain the accuracy and logistic loss curve close to that of passive learning, it requests the fewest number of points. In Appendix D, we report these figures for all 18 datasets.

Overall for the 3-layer networks, our results show that the ALPS algorithm outperforms the ex-post best margin algorithm on all but one dataset. Similarly, for the 4-layer networks, ALPS performs better than the margin algorithm on all but two dataset. Thus, despite the advantage given by the ex-post tuning of the margin algorithm, ALPS attains a better performance. Moreover, ALPS outperforms both passive learning and IWAL on all datasets. On average across all datasets, ALPS requests only a mere 28% of the processed points. Since ALPS was run with $\mu = 0$, these experiments suggest that Assumption 1 holding with favorable C values often happens in practice or that Assumption 1 is simply an artifact of our analysis.

Additionally, the ex-post best margin algorithm outperforms IWAL on the majority of the datasets, which is consistent with previous studies [Cortes et al., 2020]. IWAL attains a higher variance across the trials for some datasets, and overall it performs better than passive learning. In general, the same empirical conclusions about the relative performance of the active learning algorithms hold for logistic loss and accuracy curves for all algorithms except for the margin algorithm. On a few datasets, the margin algorithm attains a logistic loss curve that is worse than that of passive learning while admitting an accuracy curve that is on par to that of passive learning (e.g., ijcnn).

6 Conclusion

We designed an active learning algorithm, called ALPS, for general loss functions that learns to leverage pseudo-labels by using requester functions in order to train a model over a joint set of pseudo-label and labeled points. ALPS operates in the general agnostic setting; its model is guaranteed to converge to the best-in-class prediction model at the same rate as passive learning, while achieving favorable label complexity guarantees under a mild assumption on the class of requester functions. Our comprehensive empirical study on 18 datasets shows that ALPS outperforms relevant baselines, including the margin algorithm often used in practice. As a next step, we will investigate how to extend our algorithm and its associated analysis to multi-class classification setting.

Acknowledgments. Most of this research was done while the Tobias Sommer Thune was visiting Google Research in New York. Tobias Sommer Thune further acknowledges partial support by the Independent Research Fund Denmark, grant number 9040-00361B. We thank the NeurIPS anonymous reviewers whose comments helped us improve the presentation of this paper.

References

Kareem Amin, Giulia DeSalvo, and Afshin Rostamizadeh. Understanding the effects of batching in online active learning. In *Proceedings of AISTATS*, 2020.

Jordan Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *Proceedings of ICLR*, 2020.

Pranjal Awasthi, Maria Florina Balcan, and Philip M Long. The power of localization for efficiently learning linear separators with noise. In *Proceedings of STOC*, pages 449–458. ACM, 2014.

Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Ruth Urner. Efficient learning of linear separators under bounded noise. In *Proceedings of COLT*, pages 167–190, 2015.

Maria-Florina Balcan and Phil Long. Active and passive learning of linear separators under log-concave distributions. In *Proceedings of COLT*, pages 288–316, 2013.

Maria-Florina Balcan, Andrei Broder, and Tong Zhang. Margin based active learning. In *Proceedings of COLT*, pages 35–50. Springer, 2007.

Peter Bartlett, Michael Jordan, and Jon McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101:138–156, 2006.

Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance weighted active learning. In *Proceedings of ICML*, pages 49–56. ACM, 2009.

Klaus Brinker. Incorporating diversity in active learning with support vector machines. In *Proceedings of ICML*, 2003.

Rui Castro and Robert Nowak. Minimax bounds for active learning. *IEEE Transactions on Information Theory*, 54:2339–2353, 2008.

Nicolò Cesa-Bianchi and Claudio Gentile. Improved risk tail bounds for on-line algorithms. *IEEE Transactions on Information Theory*, 54(1):386–390, 2008.

Chih-Chung Chang and Chih-Jen Lin. Libsvm : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011.

Galen Chuang, Giulia DeSalvo, Lazarus Karydas, Jean-francois Kagy, Afshin Rostamizadeh, and A Theeraphol. Active learning empirical study. In *NeurIPS 2019 LIRE Workshop*, 2019.

Corinna Cortes, Spencer Greenberg, and Mehryar Mohri. Relative deviation learning bounds and generalization with unbounded loss functions. *ArXiv:1310.5796*, 2013.

Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Learning with rejection. In *Proceedings of ALT*, 2016.

Corinna Cortes, Giulia DeSalvo, Claudio Gentile, Mehryar Mohri, and Ningshan Zhang. Active learning with disagreement graphs. In *Proceedings of ICML*, 2019a.

Corinna Cortes, Giulia DeSalvo, Claudio Gentile, Mehryar Mohri, and Ningshan Zhang. Region-based active learning. In *Proceedings of AISTATS*, 2019b.

Corinna Cortes, Giulia DeSalvo, Claudio Gentile, Mehryar Mohri, and Ningshan Zhang. Adaptive region-based active learning. In *Proceedings of ICML*, 2020.

Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *Proceedings of AAAI*, 2005.

Sanjoy Dasgupta, Adam Tauman Kalai, and Claire Monteleoni. Analysis of perceptron-based active learning. In *Proceedings of COLT*, pages 249–263. Springer, 2005.

Sanjoy Dasgupta, Daniel Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. In *Proceedings of NeurIPS*, pages 353–360, 2008.

Ofer Dekel, Claudio Gentile, and Karthik Sridharan. Selective sampling and active learning from single and multiple teachers. *Journal of Machine Learning Research*, 13:2655–2697, 2012.

Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. *arXiv*, 2018.

Yarin Gal, Riashat Islam, and Zoubin. Ghahramani. Deep bayesian active learning with image data. In *Proceedings of ICML*, 2017.

Yonatan Geifman and Ran El-Yaniv. Deep active learning over the long tail. *arXiv*, 2017.

Steve Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of ICML*, pages 353–360. ACM, 2007.

Steve Hanneke. Rates of convergence in active learning. *The Annals of Statistics*, 39(1):333–361, 2011.

Steve Henneke and Liu Yang. Surrogate losses in passive and active learning. *Electronic Journal of Statistics*, 2019.

Ajay Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classi cation. In *Proceedings of CVPR*, 2009.

Sham Kakade and Ambuj Tewari. On the generalization ability of online strongly convex programming algorithms. In *Proceedings of NeurIPS*, pages 801–808, 2009.

Vladimir Koltchinskii. Local rademacher complexities and oracle inequalities in risk minimization. *The Annals of Statistics*, 34(6):2593–2656, 2006.

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

Alex Krizhevsky. Learning multiple layers of features from tiny images. *Tech Report*, 2009.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. http://yann.lecun.com/exdb/mnist/ .

David Lewis and William Gale. A sequential algorithm for training text classi ers. In *Proceedings of SIGIR*, 1994.

Enno Mammen and Alexandre Tsybakov. Smooth discrimination analysis. *The Annals of Statistics*, 27:1808–1829, 1999.

Pascal Massart and Elodie Nedelec. Risk bounds for statistical learning. *The Annals of Statistics*, 34(5):2326–2366, 2006.

Jooyoung Moon, Jiho Kim, Younghak Shin, and Sangheum Hwan. Con dence aware learning for deep neural networks. *arXiv*, 2020.

Stephen Mussmann and Percy Liang. Uncertainty sampling is preconditioned stochastic gradient descent on zero-one loss. In *Proceedings of NeurIPS*, 2018.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Silvio Savarese, Ozan Sener, and Zoubin. Ghahramani. Active learning for neural network a core set approach. In *Proceedings of ICLR*, 2018.

Greg Schohn and David Cohn. Less is more: Active learning with support vector machines. In Proceedings of ICML, 2000.

Christopher Schroder and Andreas. Niekler. A survey of active learning for text classification for deep neural networks. In arXiv, 2020.

Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. Journal of Machine Learning Research, 2001.

Alexandre Tsybakov. Optimal aggregation of classifiers in statistical learning. The Annals of Statistics, 32(1):135–166, 2004.

Tim Van Erven, Peter Grunwald, Nishant Mehta, Mark Reid, and Robert Williamson. Fast rates in statistical and online learning. Journal of Machine Learning Research, 16:1793–1861, 2015.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: Networked science in machine learning. SIGKDD Explorations, 15(2):49–60, 2013. doi: 10.1145/2641190.2641198. URL http://doi.acm.org/10.1145/2641190.2641198 .

Yazhou Yang and Marco Loog. A benchmark and comparison of active learning for logistic regression. In arXiv, 2016.

Chicheng Zhang. Efficient active learning of sparse halfspaces. In Proceedings of COLT, 2018.

Chicheng Zhang, Jie Shen, and Pranjal Awasthi. Efficient active learning of sparse halfspaces with arbitrary bounded noise. In Proceedings of NeurIPS, 2020.

Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. Annals of Statistics, 2004.

A Proofs of the Generalization Bound

In this appendix, we prove the generalization guarantees of the hypothesis returned by the ALPS algorithm. We first derive necessary concentration inequalities, then prove consistency guarantees, and show how to control the noise $\hat{\Delta}_n$. Throughout, we will be making use of some technical lemmas in Appendix C. For simplicity, we define a disagreement-based request time t to be when h_t and h_{t+1} exist and $\widehat{\mathrm{err}}_{n,t+1}(h_t, y) \geq \widehat{\mathrm{err}}_{n,t+1}(h_y) + \tilde{\Delta}_{n,t+1}$ for $y \in \{-1, 1\}$.

A.1 Concentration Inequalities

In this section, we provide a series of concentration inequalities used throughout. We consider finite classes, but our analysis can straightforwardly be generalized to VC classes via standard covering arguments. We will need the following definition:

$$\alpha_n(|\mathcal{H}|) = 2\sqrt{B}\sqrt{\frac{4\log(n(n+1)|\mathcal{H}|/\delta)}{n}}\left(\frac{1}{2} + \sqrt{1 + \frac{1}{2}\log\left(\frac{1}{2}\sqrt{\frac{n}{4\log(n(n+1)|\mathcal{H}|/\delta)}}\right)}\right),$$

where $\mathcal{H}$ is a hypothesis space of cardinality $|\mathcal{H}|$.

Lemma 1. Consider a hypothesis space $\mathcal{H}$ of cardinality $|\mathcal{H}|$ and a loss function $g : \mathcal{X} \times \mathcal{Y} \to [0, B]$. Let Z_n be an i.i.d sample of size n from the underlying distribution and let $\mathbb{E}_{Z_n}[g]$ be the empirical mean of g over the set Z_n. Then, for any $\delta > 0$, with probability at least $1 - \delta$, it holds for all n and all $h \in \mathcal{H}$:

$$\begin{aligned}
&\mathbb{E}[g] - \mathbb{E}_{Z_n}[g] \leq \min\left(\alpha_n(|\mathcal{H}|)\sqrt{\mathbb{E}[g]}, \alpha_n(|\mathcal{H}|)^2 + \alpha_n(|\mathcal{H}|)\sqrt{\mathbb{E}_{Z_n}[g]}\right)\\
&\mathbb{E}_{Z_n}[g] - \mathbb{E}[g] \leq \min\left(\alpha_n(|\mathcal{H}|)\sqrt{\mathbb{E}_{Z_n}[g]}, \alpha_n(|\mathcal{H}|)^2 + \alpha_n(|\mathcal{H}|)\sqrt{\mathbb{E}[g]}\right).
\end{aligned}$$

Proof. The lemma follows from a direct application of Corollary 14 of [Cortes et al., 2013]. Note that the second moments fulfill $\mathbb{E}[g^2] \leq B\,\mathbb{E}[g]$ as the loss g is assumed to be bounded and non-negative. For finite class $\mathcal{H}$, the shattering coefficient of set Z_n, denoted by $S_q(Z_n)$, is bounded by $S_q(Z_n) \leq n|\mathcal{H}|$. Replacing δ by $\delta/(n(n+1))$ while taking a union bound over all n, the following inequalities hold for all n and all $h \in \mathcal{H}$, for any $\delta > 0$, with probability at least $1 - \delta$:

$$\begin{aligned}
\mathbb{E}[g] &\leq \mathbb{E}_{Z_n}[g] + \alpha_n(|\mathcal{H}|)\sqrt{\mathbb{E}[g]},\\
\mathbb{E}_{Z_n}[g] &\leq \mathbb{E}[g] + \alpha_n(|\mathcal{H}|)\sqrt{\mathbb{E}_{Z_n}[g]}.
\end{aligned}$$

The bound follows follows directly by applying Lemma 7 (Appendix C) to inequalities above. □

Next, we derive a guarantee based on the loss differences, which will dictate the form of the final generalization guarantee of the ALPS algorithm. To do so, we define the following functions:

$$\begin{aligned}
g^+_{h;h'}(x, y) &= \ell(h(x), y) - \ell(h'(x), y) \quad \text{If } \ell(h(x), y)) > \ell(h'(x), y)\},\\
g_{h;h'}(x, y) &= \ell(h'(x), y) - \ell(h(x), y) \quad \text{If } \ell(h(x), y)) < \ell(h'(x), y)\}.
\end{aligned}$$

so that the difference of $g^+_{h;h'}$ and $g_{h;h'}$ equals to the loss difference of h and h', both empirically and in expectation:

$$\begin{aligned}
\mathbb{E}_Z[g^+_{h;h'}] - \mathbb{E}_Z[g_{h;h'}] &= \mathrm{err}(h, Z) - \mathrm{err}(h', Z),\\
\mathbb{E}[g^+_{h;h'}] - \mathbb{E}[g_{h;h'}] &= \mathrm{err}(h) - \mathrm{err}(h').
\end{aligned}$$

Both g^+ and g^- can be seen as bounded loss functions of a hypothesis $(h, h') \in \mathcal{H}^2$ and as such we can apply Lemma 1 to the the space $\mathcal{H}^2$ of cardinality $|\mathcal{H}|^2$, where we recall that $\mathcal{H}$ is the hypothesis class used in the algorithm.

For below, we let $\tilde{\Delta}_n(h, h') = \frac{1}{n}\sum_{s=1}^n |\ell(h(x_s), y_s) - \ell(h'(x_s), y_s) - (\ell(h(x_s), \tilde{y}_s) - \ell(h'(x_s), \tilde{y}_s))|$, which is equivalent to the definition of $\hat{\Delta}_n(h, h')$ except for absolute values and will be dealt with in a similar way.

Corollary 1. For any > 0, with probability at least $1 \quad$, for any $h; h^0 2 H$ and any n,

$$\widehat{err}_n(h) \quad \widehat{err}_n(h^0) \quad err(h) \quad err(h^0) \quad A_n(h;h^0) + \quad _n(jHj^2)^2 + 2 \quad _n(jHj^2) \quad \widetilde{A}_n(h;h^0) + \widehat{err}_n(h) + \widehat{err}_n(h^0):$$

Proof. First, we recall the de nitions of $A_n(h;h^0)$:

$$A_n(h;h^0) = \widehat{err}_n(h) \quad \widehat{err}_n(h^0) \quad [\widehat{err}_n(h) \quad \widehat{err}_n(h^0)] :$$

Letting Z be the full data set $S_n [T_n$ with true, unknown labels, it follows that $E_Z [g^+_{h;h^0}] \quad E_Z [g_{h;h^0}] = err(h; S_n [T_n) \quad err(h^0; S_n [T_n) = err_n(h) \quad err_n(h^0)$. Using this fact and rewriting A_n, it holds that

$$\widehat{err}_n(h) \quad \widehat{err}_n(h^0) = \underset{Z}{E}[g^+_{h;h^0}] \quad \underset{Z}{E}[g_{h;h^0}] \quad A_n(h;h^0) :$$

Applying Lemma 1 to $g^+_{h;h^0}$ and $g_{h;h^0}$, for $h; h^0 2 H$,

$$\begin{aligned} \widehat{err}_n(h) \quad \widehat{err}_n(h^0) \quad E[g^+_{h;h^0}] \quad E[g_{h;h^0}] \quad & A_n(h;h^0) + \quad _n(jHj^2)^2 \\ & + \quad _n(jHj^2) \sqrt{\underset{Z}{E}[g^+_{h;h^0}]} + \quad _n(jHj^2) \sqrt{\underset{Z}{E}[g_{h;h^0}]} : \end{aligned}$$

Next, we can bound the expectations under the square roots :

$$\begin{aligned} & \underset{Z}{E}[g^+] \quad \frac{1}{n} \sum_{s=1}^{n} j`(h(x_s); y_s) \quad `(h^0(x_s); y_s)j \\ & = \frac{1}{n} \sum_{s=1}^{n} j`(h(x_s); y_s) \quad `(h(x_s); \hat{y}_s) + `(h(x_s); \hat{y}_s) \quad `(h^0(x_s); \hat{y}_s) + `(h^0(x_s); \hat{y}_s) \quad `(h^0(x_s); y_s)j \\ & \quad \frac{1}{n} \sum_{s=1}^{n} j`(h(x_s); y) \quad `(h^0(x_s); y_s) \quad (`(h(x_s); \hat{y}_s) \quad `(h^0(x_s); \hat{y}_s))j + `(h(x_s); \hat{y}_s) + `(h^0(x_s); \hat{y}_s) \\ & = \widetilde{A}_n(h;h^0) + \widehat{err}_n(h) + \widehat{err}_n(h^0) : \end{aligned}$$

Inserting this above and using the fact that $E[g^+_{h;h^0}] \quad E[g_{h;h^0}] = err(h) \quad err(h^0)$ nishes the proof. □

Notice that the above bound has the noise term $_n$ and its close cousin $\widetilde{A}_n$. In the next section, we show how by using requesters functions, this noise term can be bounded, thereby resulting in a favorable generalization guarantee. For simplicity, throughout the rest of the paper, we will abbreviate $_n(jHj^2)$ by $_n$.

A.2 Consistency and Controlling the Noise $\widetilde{A}_n$.

We start by some necessarity de nitions of slack terms used by the algorithm. The slack term used for trimming the version space $\widehat{E}_n$ is given by

$$\begin{aligned} {}^0_n := \frac{2}{n} \Bigg(& \sum_{s=1}^{n} p_s + 6 \quad \ln \frac{(3 + n)n^2}{} \Bigg) \quad \ln \frac{8n^2 jF_1 j^2 \ln(n)}{} \end{aligned}$$

and the slack term used in make the disagreement based requests is given by:

$$\widetilde{}_n := \quad _n + \quad ^2_n + 2 \quad _n \sqrt{\quad _n + \widehat{err}_n(h_+) + \widehat{err}_n(h \quad)} ;$$

where $_n$ is de ned as follows:

$$_n := \frac{8B^2}{n} \sum_{s=1}^{n} \quad ^0_{s \quad 1} + \frac{2B}{n} \ln \frac{jF_1 j^2 n(n+1)}{} + \frac{2\sqrt{B}}{n} \Bigg(\sum_{s=1}^{n} \quad ^0_{s \quad 1} \Bigg) \ln \frac{jF_1 j^2 n(n+1)}{} :$$

In the following lemma, we show a bound on the noise $A_n(h, h^0)$ (and $\tilde{A}_n$) in terms of ϵ_n. As we will see, the requester function r_h are the key component in the proof of this lemma. In fact, the main role of the requester functions is to control the noise term A_n so that it bounded by ϵ_n. Notice that this directly implies that $\tilde{\epsilon}_n$ is a upper bound on the right-hand side of the generalization guarantee of Corollary 1, excluding $\mathrm{err}(h) - \mathrm{err}(h^0)$, that can be empirically computed. At each round n, this empirically known generalization upper bound is leveraged by our algorithm to decide whether to request the label (i.e. disagreement based request) or infer the sign $\mathrm{sign}(x_n)$ with high probability and use a pseudo-label that matches this sign. The lemma below will be used both in the subsequent consistency theorem and later also in the label complexity analysis.

Lemma 2. Assume that for any $s \in [n]$, it holds that $(h, r) \in F_s$, h is consistent on $\hat{S}_{s-1}$ and if a disagreement based request is not made for y_s, then h is consistent on $\hat{S}_{s-1} \cup (x_s, \hat{y}_s)$. Then, for any $\delta > 0$, with probability at least $1 - \delta$,

$$|A_n(h, h^0)| \leq \epsilon_n \text{ and } \tilde{A}_n(h, h^0) \leq \epsilon_n$$

for any $h, h^0 \in H$.

Proof. We start by deriving an intermediate bound on $A_n(h, h^0)$ for any $h, h^0 \in H$. First, we re-write the noise term as:

$$|A_n(h, h^0)| = \frac{1}{n}\sum_{s=1}^{n} \ell(h(x_s), y_s) - \ell(h^0(x_s), y_s) - (\ell(h(x_s), \hat{y}_s) - \ell(h^0(x_s), \hat{y}_s)) \; . \quad (2)$$

Each component in the sum of Equation (2) is non-zero only when $y_s \neq \hat{y}_s$. Note that $y_s \neq \hat{y}_s$ whenever the algorithm does not request ($r_s(x_s) = 0$), and whenever $y_s \neq \mathrm{sgn}(h(x_s))$ for any h consistent on $\hat{S}_s$.

Consider the pair $(h_s^0, r_s) \in F_s$ where h_s^0 is consistent on $\hat{S}_{s-1} \cup (x_s, \hat{y}_s)$. There always exists such a pair by the assumptions that $(h, r) \in F_s$ and h is consistent on $\hat{S}_{s-1} \cup (x_s, \hat{y}_s)$. Using the fact that $\mathrm{sgn}(h_s^0(x_s)) = \hat{y}_s$ holds with probability $1 - \delta$, it then follows with probability $1 - \delta$ that

$$\begin{aligned}(2) = &\frac{1}{n}\sum_{s=1}^{n} (\ell(h(x_s), y_s) - \ell(h^0(x_s), y_s) - (\ell(h(x_s), \hat{y}_s) - \ell(h^0(x_s), \hat{y}_s))) \\ &\qquad \mathbf{I}\{y_s \neq \hat{y}_s\}\mathbf{I}\{r_s(x_s) = 0\} \\ &\frac{1}{n}\sum_{s=1}^{n} 4B\,\mathbf{I}\{y_s \neq \mathrm{sgn}(h_s^0(x_s))\}\mathbf{I}\{r_s(x_s) = 0\} \; ; \end{aligned} \quad (3)$$

where we bounded the loss differences by $4B$.

Next, we bound the expectation of the above random variable on the right-hand-side of (3) by using Lemma 6, which holds uniformly for all $(h, r) \in F_s$. Since $(h_s^0, r_s) \in F_s$, Lemma 6 states that for any $\delta > 0$, with probability at least $1 - \delta$,

$$\mathbb{E}[\ell(y, h_s^0(x))\mathbf{I}\{r_s(x) = 0\}] \leq \min_{(h,r)\in F_s} \mathbb{E}[\ell(y, h(x))\mathbf{I}\{r(x) = 0\}] + 2B\,\epsilon_{s-1}. \quad (4)$$

Using this fact and bounding the expectation of the right-hand-side of (3) by the surrogate loss ℓ, it follows that

$$\begin{aligned}\mathbb{E}[\mathbf{I}\{y \neq \mathrm{sgn}(h_s^0(x))\mathbf{I}\{r_s(x) < 0\}] &\leq \mathbb{E}[\ell(y, h_s^0(x))\mathbf{I}\{r_s(x) < 0\}] \\ &\leq \min_{(h,r)\in F_s} \mathbb{E}[\ell(y, h(x))\mathbf{I}\{r(x) = 0\}] + 2B\,\epsilon_{s-1} \\ &= 2B\,\epsilon_{s-1}\end{aligned}$$

where the last equality holds by Assumption 1.

The statement of the theorem for A_n follows by using this inequality in conjunction with Inequality (3) and applying Lemma 3 of [Kakade and Tewari, 2009] to the martingale difference sequence

$$I_s = \mathbf{I}\{y_s \neq \mathrm{sgn}(h_s^0(x_s))\mathbf{I}\{r_s(x_s) < 0\} - \mathbb{E}[\mathbf{I}\{y_s \neq \mathrm{sgn}(h_s^0(x_s))\mathbf{I}\{r_s(x_s) < 0\}|I_{s-1}]\; ;$$

where $I_s = \{(x_1, y_1, Q_1), \ldots, (x_s, y_s, Q_s)\}$ is the history up to time s. The above analysis can be similarly applied to $\tilde{A}_n$ to conclude the proof. □

Theorem 3. For any $\delta > 0$, with probability at least $1 - \delta$, it holds that for all $n > 0$:

(1) $(h^*, r^*) \in F_n$;

(2) h^* is consistent on $\hat{S}_{n-1}$;

(3) if a disagreement based request is not made for y_n, h^* is consistent on $\hat{S}_{n-1} \cup (x_n, y_n)$. In particular, h^* is consistent on $\hat{S}_n$.

Proof. We proceed by strong induction over n. For the base case, consider $n = 1$. Here a disagreement-based request will always be made, so being consistent with $\hat{S}_0 = \emptyset$ is clear. Also since $F_1 = H \times R$, it follows that $(h^*, r^*) \in F_1$. Assume now that the theorem holds for time steps $1, \ldots, n$.

To prove (1), by Lemma 5, it holds that for $(h', r') = \operatorname{argmin}_{(h,r) \in F_n} l_n(h, r)$,

$$l_n(h^*, r^*) - l_n(h', r') \le E[\ell_B(h^*(x), y) \mid r^*(x) = 0] - E[\ell_B(h'(x), y) \mid r'(x) = 0] + \frac{\delta'_n}{2}.$$

The last inequality above follows since

$$E[\ell_B(y, h^*(x)) \mid r^*(x) = 0] = \min_{(h,r) \in H \times R} E[\ell_B(y, h(x)) \mid r(x) = 0] \le E[\ell_B(y, h'(x)) \mid r'(x) = 0].$$

Thus, $(h^*, r^*) \in F_{n+1}$ by definition.

To prove (2) and (3), we need only consider the case where the algorithm is not querying due to disagreement, so without loss of generality, we assume that upon seeing x_{n+1}, we have

$$\widehat{err}_n(h_+) - \widehat{err}_n(h^*) > \tilde{\varepsilon}_n.$$

Our goal is then to show that $\operatorname{sign} h^*(x_{n+1}) = -1$. For the sake of contradiction, assume the opposite. Then $err_n(h^*) \ge err_n(h_+)$. Using the above two conditions, it follows that

$$\begin{aligned} \widehat{err}_n(h^*) - err_n(h^*) &= \widehat{err}_n(h^*) - \widehat{err}_n(h_+) + (\widehat{err}_n(h_+) - err_n(h^*)) \\ &> \widehat{err}_n(h^*) - \widehat{err}_n(h_+) + \tilde{\varepsilon}_n - \tilde{\varepsilon}_n \\ &= \varepsilon_n + \frac{\delta'_n}{2} + 2\sqrt{\varepsilon_n + \widehat{err}_n(h^*) + \widehat{err}_n(h_+)}. \end{aligned}$$

At the same time, from our Corollary 1 and Lemma 2 which holds with probability at least $1 - \delta$ and by the inductive assumption, we have

$$err(h^*) - err(h_+) \le \widehat{err}_n(h^*) - \widehat{err}_n(h_+) + \varepsilon_n + \frac{\delta'_n}{2} + 2\sqrt{\varepsilon_n + \widehat{err}_n(h^*) + \widehat{err}_n(h_+)}.$$

Combined these two bounds would imply $err(h^*) > err(h_+)$, which is a contradiction. □

A.3 Final Generalization Bound

Given the above results, we are ready to prove the final generalization bound of the ALPS algorithm.

Theorem 1. Under Assumption 1, for any $\delta > 0$, with probability at least $1 - \delta$, for any $n > 0$,

$$err(h_n) \le \nu + O\left(\sqrt{\frac{\nu}{n} + \frac{1}{n}}\right),$$

where h_n is the hypothesis computed by ALPS in round n.

Proof. From Corollary 1, with probability at least $1 - \delta$,

$$err(h_n) \le \nu + |A_n(h_n, h^*)| + \frac{\delta'_n}{2} + 2\sqrt{\varepsilon_n + A_n(h_n, h^*) + \widehat{err}_n(h_n) + \widehat{err}_n(h^*)}.$$

Focusing on the terms under the square root, since $\widehat{\mathrm{err}}_n(h_n) \le \widehat{\mathrm{err}}_n(h^*)$ by definition, we need to bound $\widehat{\mathrm{err}}_n(h^*)$. Since the main difference between $\widehat{\mathrm{err}}_n(h^*)$ and $\mathrm{err}_n(h^*)$ depends on how often the pseudo-labels differ from the true labels, it follows that $\widehat{\mathrm{err}}_n(h^*) \le \mathrm{err}_n(h^*) + \frac{B}{n}\sum_{t=1}^{n} \mathbf{1}\{y_t \neq \hat{y}_t\}$. By Theorem 3, h^* is consistent on the pseudo-labels, that is with probability $1-\delta$, $\mathrm{sgn}(h^*(x_t)) = \mathrm{sgn}(\hat{y}_t)$ for any pseudo-labeled $\hat{y}_t$ for $t \ge 2$ [n]. Using this fact and since zero-one loss is bounded by the loss function, it holds that $\frac{B}{n}\sum_{t=1}^{n} \mathbf{1}\{y_t \neq \hat{y}_t\} \le B\,\mathrm{err}_n(h^*)$. Putting the above together, $\widehat{\mathrm{err}}_n(h^*) \le (1+B)\,\mathrm{err}_n(h^*) \le (1+B)(\epsilon^* + \epsilon_n + \sqrt{\epsilon_n})$, where the last inequality follows from using Lemma 1, which holds with probability $1-\delta$. Thus,

$$\mathrm{err}(h_n) \le \epsilon^* + |A_n(h_n, h^*)| + 2\sqrt{\epsilon_n\left(\widetilde{A}_n(h_n, h^*) + O(\epsilon^* + \epsilon_n + \sqrt{\epsilon_n})\right)}.$$

By using Theorem 3 and Lemma 2, we bound A_n and $\widetilde{A}_n$ term as follows with probability at least $1-\delta$,

$$\mathrm{err}(h_n) \le \epsilon^* + \epsilon_n + 2\sqrt{\epsilon_n\left(\epsilon_n + O(\epsilon^* + \epsilon_n + \sqrt{\epsilon_n})\right)}.$$

Recalling that $\epsilon_n = \sqrt{\frac{8B^2}{n}\sum_{s=1}^{n}\mathbf{1}_{\{s<s'\le n\}} + \frac{2B}{n}\ln\frac{|\mathcal{F}_1|^2 n(n+1)}{2\sqrt{2B}}}\sqrt{\sum_{s=1}^{n}\mathbf{1}_{\{s<s'\le n\}}\ln\frac{|\mathcal{F}_1|^2 n(n+1)}{2\sqrt{2B}}}$ and by the same reasoning as in Theorem 5 and by Assumption 1, it holds, with probability at least $1-\delta$, that $\sum_{s}\mathbf{1}_{\{s<s'\le n\}} = O(\log s)$. The statement then follows from $\epsilon_n = O(1/\sqrt{n})$, while absorbing constants factors of from the union bounds into log factors. □

B Proofs of the Label Complexity

In this appendix, we first bound the amount of labels requested due to disagreement-based requests and then bound the queries used for learning the requester functions.

B.1 Bounding the disagreement-based requests

Lemma 3. For any $\delta > 0$, with probability at least $1-\delta$, letting $h^*(x_{n+1}) = \hat{y}$, the probability of requesting y_{n+1} due to disagreement-based request is bounded as

$$P[\widehat{\mathrm{err}}_n(h^*_{\hat{y}}) \le \widehat{\mathrm{err}}_n(h^*_{\hat{y}}) \le \widetilde{\epsilon}_n] \le P\,\mathrm{err}(h^*_{\hat{y}}) = O\left(\epsilon^* + \epsilon_n^2\right).$$

Proof. We use several facts that hold concurrently with probability at least $1-\delta$: The original generalization bound for a single hypothesis, Lemma 2, and Theorem 3.

Without loss of generality, we assume $h^*(x_{n+1}) = +1$ and that the label is requested. The hypothesis in the lemma statement is then $h^*_{\hat{y}} = h^*_+$. We have

$$\widehat{\mathrm{err}}_n(h^*_+) - \widehat{\mathrm{err}}_n(h^*_-) \le \epsilon_n + \epsilon_n^2 + 2\sqrt{\epsilon_n\left(\epsilon_n + \widehat{\mathrm{err}}_n(h^*_+) + \widehat{\mathrm{err}}_n(h^*_-)\right)}.$$

We lower bound the left-hand-side by $\mathrm{err}_n(h^*_+) - \widehat{\mathrm{err}}_n(h^*_-) = \mathrm{err}_n(h^*_+) - \mathrm{err}_n(h^*_-) - A_n(h^*_+, h^*_-) \ge \mathrm{err}_n(h^*_+) - \mathrm{err}_n(h^*_-) - \epsilon_n$ by Lemma 2 and Theorem 3. Then, we upper bound the right-hand-side by using $\widehat{\mathrm{err}}_n(h^*_+) \le (1+B)\,\mathrm{err}_n(h^*_+)$ and $\widehat{\mathrm{err}}_n(h^*_-) \le (1+B)\,\mathrm{err}_n(h^*_-)$ by a similar reasoning as in Theorem 1. Thus,

$$\mathrm{err}_n(h^*_+) - \mathrm{err}_n(h^*_-) \le 2\epsilon_n + \epsilon_n^2 + 2\sqrt{\epsilon_n\left(\epsilon_n + (1+B)(\mathrm{err}_n(h^*_+) + \mathrm{err}_n(h^*_-))\right)}.$$

Using $\sqrt{a+b} \le \sqrt{a} + \sqrt{b}$ for positive a, b and Lemma 7 (therein with $A = \mathrm{err}_n(h^*_+)$), we then have

$$\mathrm{err}_n(h^*_+) - \mathrm{err}_n(h^*_-) \le 2\epsilon_n + 4(2+B)\left(\epsilon_n^2 + 2\sqrt{\epsilon_n\left(\epsilon_n + (1+B)\,\mathrm{err}_n(h^*_-)\right)}\right) + 2\sqrt{\epsilon_n\left((1+B)(\mathrm{err}_n(h^*_-) + 2\sqrt{\epsilon_n\left(\epsilon_n + \epsilon_n^2 + 2\sqrt{\epsilon_n\left(\epsilon_n + (1+B)\,\mathrm{err}_n(h^*_-)\right)}\right)}\right)}.$$

We now use Lemma 1 on the true unknown empirical errors: $\mathrm{err}_n(h_+) - \mathrm{err}(h_+) \le \sqrt{\frac{\mathrm{err}(h_+)}{n}}\sqrt{\mathrm{err}(h_+)} + \frac{2}{n}$ and $\mathrm{err}_n(h_-) - \mathrm{err}(h_-) \le \sqrt{\frac{\mathrm{err}(h_-)}{n}}\sqrt{\mathrm{err}(h_-)} + \frac{2}{n}$. We further repeatedly use the reduction $\sqrt{ab} - \sqrt{cd} \le \sqrt{a-c}\sqrt{b} + \sqrt{c}\sqrt{b-d}$.

$$\mathrm{err}(h_+) - \sqrt{\frac{\mathrm{err}(h_+)}{n}}\sqrt{\mathrm{err}(h_+)} = O\left(\frac{2}{n} + \sqrt{\frac{2}{n}}\sqrt{\mathrm{err}(h_-)} + \sqrt{\frac{\mathrm{err}(h_-)}{n}}\sqrt{\mathrm{err}(h_-) - \frac{\mathrm{err}(h_-)}{n}}\right).$$

Putting the above together along with the definition of Δ_n and using Lemma 7 once more finishes the proof. □

Definition 3. We define two metrics. The first on the space of hypotheses as

$$\rho(h, h') := \mathop{E}_{(x,y)\sim D}\left|\ell(h(x), y) - \ell(h'(x), y)\right|.$$

The second metric is defined on the space $H \times R$:

$$\rho'((h, r), (h', r')) = E[|\ell_B(h(x), y) - \ell_B(h'(x), y)| \mathbf{1}[f_r(x) \neq 0 \text{ or } f_{r'}(x) \neq 0]].$$

Using each metric, we define the ball around the best-in-class as

$$B_\rho(h^*, \varepsilon) = \{h \in H : \rho(h, h^*) \le \varepsilon\} \text{ and } B'_{\rho'}((h^*, r^*), \varepsilon) = \{(h, r) \in H \times R : \rho'((h, r), (h^*, r^*)) \le \varepsilon\}.$$

With these metrics in place, we introduce the following disagreement coefficients. These can be thought of as two different generalizations of the binary disagreement used for the zero-one loss, each tailored to one of the two notions of consistency or disagreement used in the analysis.

Definition 4. Let the disagreement coefficient with respect to H be

$$\theta := \sup_{\varepsilon > 0} \frac{P_X[\exists h \in B_\rho(h^*, \varepsilon) : \mathrm{sgn}(h(x)) \neq \mathrm{sgn}(h^*(x))]}{\varepsilon}.$$

Let the disagreement coefficient with respect to $H \times R$ be

$$\theta_0 = \inf\left\{\theta'' \ge 8 : \forall \varepsilon > 0, \mathop{E}_x \sup_{(h,r)\in B'((h^*,r^*),\varepsilon)} \sup_y |\ell(h(x), y)\mathbf{1}[f_r(x) \neq 0] - \ell(h^*(x), y)\mathbf{1}[f_{r^*}(x) \neq 0]| \le \theta''\varepsilon\right\}.$$

Lemma 4. With probability 1 it holds for every n, that letting $h^*(x_{n+1}) = \hat{y}$, we have

$$P[\mathrm{err}(h^{\hat{y}}) - \mathrm{err}(h^*) \ge \varepsilon] \le \theta(\rho(h^{\hat{y}}, h^*) + \varepsilon)$$

for any $\varepsilon > 0$.

Proof. Consider the high probability event that h^* is consistent with S_n, which holds due to Theorem 3, and consider the case where a request is made. From the definition of ρ,

$$\rho(h^{\hat{y}}, h^*) \ge \mathrm{err}(h^{\hat{y}}) - \mathrm{err}(h^*).$$

By this inequality, we get

$$\begin{aligned} P[\mathrm{err}(h^{\hat{y}}) - \mathrm{err}(h^*) \ge \varepsilon] &\le P[\rho(h^{\hat{y}}, h^*) \ge \varepsilon] \\ &\le P[\exists h \in B_\rho(h^*) : \mathrm{sgn}(h(x)) \neq \mathrm{sgn}(h^*(x))] \\ &\le \theta(\rho(h^{\hat{y}}, h^*) + \varepsilon), \end{aligned}$$

where the second inequality uses that $h^{\hat{y}}$ and h^* disagree on x_{n+1} by construction and the final inequality uses the disagreement coefficient of Definition 4. □

Theorem 4. For any $\delta > 0$, with probability at least $1 - \delta$, for all $n > 0$, the expected number of queried labels due to disagreement-based requests is bounded by

$$E\left[\sum_{s=1}^{n} \mathbf{1}[\mathrm{err}(h^{\hat{y}_s}) - \mathrm{err}(h^*) \ge \tilde{\Delta}_{s-1}]\right] = O\left(\theta n \log^2 n\right),$$

where the expectation is with respect to the draws of (x_s, y_s).

Proof. By combining Lemmas 3 and Lemma 4, with probability at least $1 - \delta$ for all n,

$$P[\mathrm{err}(h^{\hat{y}_{n-1}}) - \mathrm{err}(h^*) > \tilde{\Delta}_{n-1}] = O\left(\theta(\tilde{\Delta}_{n-1} + \frac{2}{n})\right).$$

Summing these terms yields the above rates as $\frac{2}{n} = O(\ln(n)/n)$. □

B.2 Bounding the component related to learning requester functions

To learn the requester functions, we use importance weighted estimates and so we seek a bound on $\sum_{s=1}^{n} Q_s$, where we recall each Q_s is coin flip with probability p_s.

Theorem 5. For all $\delta > 0$, for all $n \geq 0$, with probably at least $1 - \delta$, it holds that

$$\sum_{s=1}^{n} Q_s \leq 8\theta^0 E[\ell(y, h^*(x)) \mid r^*(x) = 0]n + O(B\sqrt{E[\ell(y, h^*(x)) \mid r^*(x) = 0]n \log(n|F_1|/\delta)}) + O(B \log^3(n|F_1|/\delta)).$$

Proof. First, we prove a bound on $E[p_s]$: By definition of θ^0, it holds that $\theta^0((h, r), (h^*, r^*)) E[\ell_B(y, h(x)) \mid r(x) = 0] + E[\ell_B(y, h^*(x)) \mid r^*(x) = 0]$. For any pair $(h, r) \in F_s$, $E[\ell_B(y, h(x)) \mid r(x) = 0] \leq E[\ell_B(y, h^*(x)) \mid r^*(x) = 0] + 2\Delta_{s-1}^0$ by Lemma 6. Hence, $F_s \subseteq B(h^*, r^*, \rho)$ where $\rho = 2E[\ell_B(y, h^*(x)) \mid r^*(x) = 0] + 2\Delta_{s-1}^0$. Then by definition of disagreement coefficient:

$$\begin{aligned}
E[p_s] &\leq E[\max_{(h,r),(h',r') \in F_s} \max_y \ell_B(y, h(x_s)) \mid r(x_s) = 0 - \ell_B(y, h'(x_s)) \mid r'(x_s) = 0] \\
&\leq 2E[\max_{(h,r) \in F_s} \max_y \ell_B(y, h(x_s)) \mid r(x_s) = 0 - \ell_B(y, h^*(x_s)) \mid r^*(x_s) = 0] \\
&\leq 2E[\max_{(h,r) \in B(h^*, r^*, \rho)} \max_y \ell_B(y, h(x_s)) \mid r(x_s) = 0 - \ell_B(y, h^*(x_s)) \mid r^*(x_s) = 0] \\
&\leq 2\theta^0 \rho.
\end{aligned}$$

Converting from ℓ_B to the surrogate loss ℓ and using the definition of ρ, we find that $E[p_s] \leq 4\theta^0(E[\ell(y, h^*(x)) \mid r^*(x) = 0] + B\Delta_{s-1}^0)$.

Using the above and by straightforward modifications of Lemma 6 in [Cortes et al., 2019b] and Theorem 1 in [Cortes et al., 2019b] concludes the proof. □

B.3 Final Label Complexity

Putting the above together, we attain the final label complexity guarantee.

Theorem 2. Under Assumption 1, for any $\delta > 0$, with probability $1 - \delta$, the label complexity of the ALPS algorithm at time $n > 0$ is bounded as $O(n(\theta\epsilon + C\epsilon))$.

Proof. The label complexity component is the addition of Theorem 5, Theorem 4 and the term $E[\mathbb{I}r_n(x) > 0]$. Since $(h^*, r^*) \in F_n$ for all n with high probability by Theorem 3, then it must be the case that $r^* \in R_n$. This in turn implies that $E[\mathbb{I}r_n(x) > 0] \leq E[\mathbb{I}r^*(x) > 0]$ since $r_n = \operatorname{argmin}_{r \in R_n} E[\mathbb{I}r(x) > 0]$. □

C Some technical lemmas

We first present two technical lemma for the importance weighted estimates $I_n(h, r)$ of $E[\ell_B(y, h(x)) \mid r(x) = 0]$ in term of the slack term Δ_n^0.

Lemma 5. For any $\delta > 0$, with probability at least $1 - \delta/2$, for all $n \geq 3$, for all $(h, r), (h', r') \in F_n$,

$$|I_n(h, r) - I_n(h', r') - E[\ell_B(y, h(x)) \mid r(x) = 0] + E[\ell_B(y, h'(x)) \mid r'(x) = 0]| \leq \Delta_n^0.$$

Proof. For $s \in [n]$, consider the random variables:

$$Z_s = \frac{Q_s}{p_s}(\ell_B(y_s, h(x_s)) \mid r(x_s) = 0 - \ell_B(y_s, h'(x_s)) \mid r'(x_s) = 0) - E[\ell_B(y, h(x)) \mid r(x) = 0] + E[\ell_B(y, h'(x)) \mid r'(x) = 0],$$

for any $(h, r), (h', r') \in F_n$.

Let $I_n = \{(x_1, y_1, Q_1), \ldots, (x_n, y_n, Q_n)\}$ be the history up to time n. By applying Lemma 4 in [Cortes et al., 2019b], for any $0 < < 1$, and $n \geq 3$, with probability at least 1 ,

$$\sum_{s=1}^{n} Z_s \leq \max\left(2\sqrt{\sum_{s=1}^{n} \mathop{\mathbb{E}}_{x_s}[p_s | I_{s-1}]}, 6\log\frac{8\log(n)}{}\right) \log\frac{8\log(n)}{} .$$

Taking a union bound over $r \geq 3$ and all $(h, r), (h', r') \in \mathcal{F}_1$, it holds that

$$\begin{aligned} &|I_n(h, r) - I_n(h', r') - \mathbb{E}[\ell_B(y, h(x)) \mid r(x) \geq 0] + \mathbb{E}[\ell_B(y, h'(x)) \mid r'(x) \geq 0]| \\ &\leq \frac{1}{n}\max\left(2\sqrt{\sum_{s=1}^{n} \mathop{\mathbb{E}}_{x_s}[p_s | I_{s-1}]}, 6\log\frac{8n^2|\mathcal{F}_1|^2\log(n)}{}\right) \log\frac{8n^2|\mathcal{F}_1|^2\log(n)}{} . \end{aligned}$$

Via Proposition 2 in [Cesa-Bianchi and Gentile, 2008], with probability at least , for all $n \geq 3$,

$$\begin{aligned} &\sum_{s=1}^{n} \mathop{\mathbb{E}}_{x_s}[p_s | I_{t-1}] \leq \sum_{s=1}^{n} p_s + 36\log\frac{(3 + \sum_{s=1}^{n} p_s)n^2}{} \\ &+ 2\sqrt{\sum_{s=1}^{n} p_s \log\frac{(3 + \sum_{s=1}^{n} p_s)n^2)}{}} \sqrt{\sum_{s=1}^{n} p_s + 6\log\frac{(3+n)n^2}{}}^2 . \end{aligned}$$

Combining the above, we attain the desired bound. □

Lemma 6. For any > 0, with probability 1 , for any $s > 0$ and any $(h, r), (h', r') \in \mathcal{F}_s$,

$$\mathbb{E}[\ell(y, h(x)) \mid r(x) \geq 0] \leq \mathbb{E}[\ell(y, h'(x)) \mid r'(x) \geq 0] + 2B \sqrt{}_{s-1} .$$

In particular, it holds that

$$\mathbb{E}[\ell(y, h(x)) \mid r(x) \geq 0] \leq \min_{(h,r)\in\mathcal{F}_s} \mathbb{E}[\ell(y, h(x)) \mid r(x) \geq 0] + 2B \sqrt{}_{s-1} .$$

Proof. Take any $(h, r), (h', r') \in \mathcal{F}_n$, then by using Lemma 5 and since $\mathcal{F}_n \subseteq \mathcal{F}_{n-1}$,

$$\begin{aligned} &\mathbb{E}[\ell_B(y, h(x)) \mid r(x) \geq 0] - \mathbb{E}[\ell_B(y, h'(x)) \mid r'(x) \geq 0] \\ &\leq I_{s-1}(h, r) - I_{s-1}(h', r') + \sqrt{}_{s-1} \\ &\leq \min_{(h,r)\in\mathcal{F}_{s-1}} I_{s-1}(h, r) + \sqrt{}_{s-1} - \min_{(h,r)\in\mathcal{F}_{s-1}} I_{s-1}(h, r) + \sqrt{}_{s-1} \\ &\leq 2\sqrt{}_{s-1} ; \end{aligned}$$

where we used the fact that $\min_{(h,r)\in\mathcal{F}_{s-1}} I_{s-1}(h, r) \leq \min_{(h,r)\in\mathcal{F}_{s-1}} I_{s-1}(h, r) + \sqrt{}_{s-1}$. By multiplying by B, the bound of the lemma directly follows. □

Next consider a technical lemma used throughout the analysis, which will allow us to exchange terms in the fast rate bounds by adding $\ln$ (corresponding to C^2).

Lemma 7. For non-negative real numbers $A, B, C \geq 0$ we have

$$A \leq B + C\sqrt{A} \implies A \leq B + C^2 + C\sqrt{B} .$$

Proof. Since it holds that $A \leq B + C\sqrt{A}$, it follows that $\sqrt{A}$ is less than the largest root of the equation $x^2 - Bx - C = 0$, that is

$$\sqrt{A} \leq \frac{C + \sqrt{C^2 + 4B}}{2}.$$

The second inequality in the lemma follows from squaring this and using $\sqrt{y + y'} \leq \sqrt{y} + \sqrt{y'}$ for positive y, y'. □

Dataset	# features	# unlabeled examples	# test examples	Notes
ijcnn	22	49990	91701	
satimage	36	4435	2000	`red soil' vs. rest
cod-rna	8	59535	271617	
mnist	780	60000	10000	'odd' vs. 'even'
cifar	3072	12000	2000	`horse' vs. `ship'
acoustic	50	78823	19705	`1' vs. rest
german	24	500	500	
pima	8	500	228	
gestures	33	5000	4873	`rest position' vs. rest
phishing	68	5000	6055	
shuttle	9	43500	14500	`rad ow' vs. rest
skin	3	100,000	145057	
australian	14	500	190	
breastcancer	10	500	183	
guide	4	3089	4000	
a9a	123	32561	16281	
mushroom	112	5119	3005	
splice	60	2000	1175	

Table 1: Dataset statistics including number of features, number of unlabeled examples, number of test examples and notes on the classi cation task.

D Experimental Supplements

We plot the performance of the ALPS algorithm, the IWAL algorithm, the ex-post best margin algorithm, and the passive learning algorithm in Figure 3,4, and 5 for 3-layer neural networks and in Figure 9,10, and 11 for 4-layer neural networks. Overall, ALPS outperforms all baselines on almost all datasets for both types of networks. In a few cases, we nd that the ex-post best margin outperforms ALPS with respect to the accuracy curve, but not with respect to logistic loss (e.g., german). We also nd that IWAL does not attain the accuracy or logistic loss of the passive learning curve on a small subset of the datasets.

For each dataset, rst each feature was normalized to have zero mean and unit variance, and then all examples in the dataset were divided by the maximum l_2-norm of the examples. That is, letting D be the dataset, each data point in the dataset $x \in D$, was divided by $\max_{x_i \in D} \|x_i\|_2$. Table 1 contains the main statistics of each dataset. The datasets were taken from the LIBSVM dataset library, the OpenML dataset library, and MNIST and CIFAR databases Vanschoren et al. [2013], Chang and Lin [2011], Krizhevsky [2009], LeCun and Cortes [2010]. If available, the train/test split in these libraries was used. Most of the datasets admit binary labels, but there is a subset of multi-class datasets, which were turned into binary classi cation tasks. Speci cally, for gestures, satimage, acoustic, and shuttle, the binary classi cation task was separating one-versus-rest for a given xed class (see table for which class). For mnist, the binary task was separating odd versus even digits and for cifar, the binary task was separating `horse' and `ship' images.

To generate the hypothesis functions for each type of neural network models, we trained the respective model on random subsets of the training data while varying hyperparamters. The points used in the random training subsets were then removed from the unlabeled pool. The Multi-layer Perceptron algorithm in scikit-learn library was used with solver=lbfgs and maximum number of iterations set to 1,000 Pedregosa et al. [2011]. The number of hidden units in the 3-layer neural network was $(50, 10, 5)$, while in the 4-layer network it was $(10, 10, 10, 10)$. For each dataset, we generated two hypothesis set, one for 3-layer networks and one for 4-layer networks, and tested the algorithms on both hypothesis sets independently to see the effects the model has on the active learning algorithms. In order to create hypothesis set, we generated 1,000 hypothesis functions by training the the Multi-layer Perceptron algorithm on random subsets of between 50 and 500 points while randomly choosing an l_2-norm regularization parameter (alpha) from the set $\{2^i\}_{i=-4}^{4}$ and randomly initializing with an integer in $\{1, \ldots, 100\}$ the random number generation (random_state) for weights and bias initialization. For the smaller datasets (australian, german, breastcancer and pima), the random training subsets used for training were instead taken between 50 and 100 points.

For the ALPS algorithm, the set of thresholds used was $\Gamma_r = \{0.1, 0.25, 0.5, 0.75, 0.9\}$. For the margin algorithm, we defined the threshold set Γ_m via the following tuning procedure. We started by running margin algorithms with $\Gamma_m = \Gamma_r$, which resulted in algorithms that requested too few points admitting poor accuracy. Thus, the threshold values were enlarged until good accuracy learning curves were reached for nine threshold values. The tuned threshold set turned out to be $\Gamma_m = \{\sum_{i=1}^{j} 9(1/10)^i\}_{j=1}^{9}$. We found that the neural networks were overly confident in their predictions resulting in high threshold values. Notice that the above tuning clearly gives an unfair advantage to the margin algorithms over the other algorithms, but it gives us an upper bound on the best possible performance in this setting.

The best exp-post margin algorithm is chosen as as follows. For all margin algorithms with thresholds in Γ_m and for the passive learning algorithm, we calculated the area under the learning curve of accuracy versus the number of observed points by using the Trapezoid Rule. The best exp-post margin algorithm is the one with smallest threshold that admits an learning curve area that is within one standard error of the passive learning curve area. If no algorithm exists, the best exp-post margin algorithm is the one with the largest learning curve area.

We report the result of the margin algorithm for all these threshold values in Figure 6,7, and 8 for 3-layer networks and in Figure 12,13, and 14 for 4-layer networks. For the legend, the margin algorithms are enumerated in order from the smallest to largest threshold in set Γ_m. That is, Margin-1 corresponds to the smallest threshold, Margin-2 to the second smallest, etc. The generalization ability of the margin algorithm largely varies with the threshold values. In almost all datasets, there exist thresholds whose corresponding margin algorithm admits an accuracy curve that is below that of passive learning and also thresholds that are on par with passive learning, thereby indicating that we tested a good set of threshold values.

For experiments with larger neural networks, ALPS can be run in the online batch setting of Amin et al. [2020] where the streaming algorithm's internal state is frozen until a batch of points is selected. This would considerably dampen the larger networks' longer training and inference times.

Note that we do not compare to the region-based algorithms of Cortes et al. [2019b, 2020] since these algorithms select hypothesis functions outside the class H. Nonetheless, both algorithms in Cortes et al. [2019b, 2020] are meta-algorithms that use IWAL as a subroutine and one can directly derive learning guarantees for region-based algorithms that instead use ALPS as a subroutine.

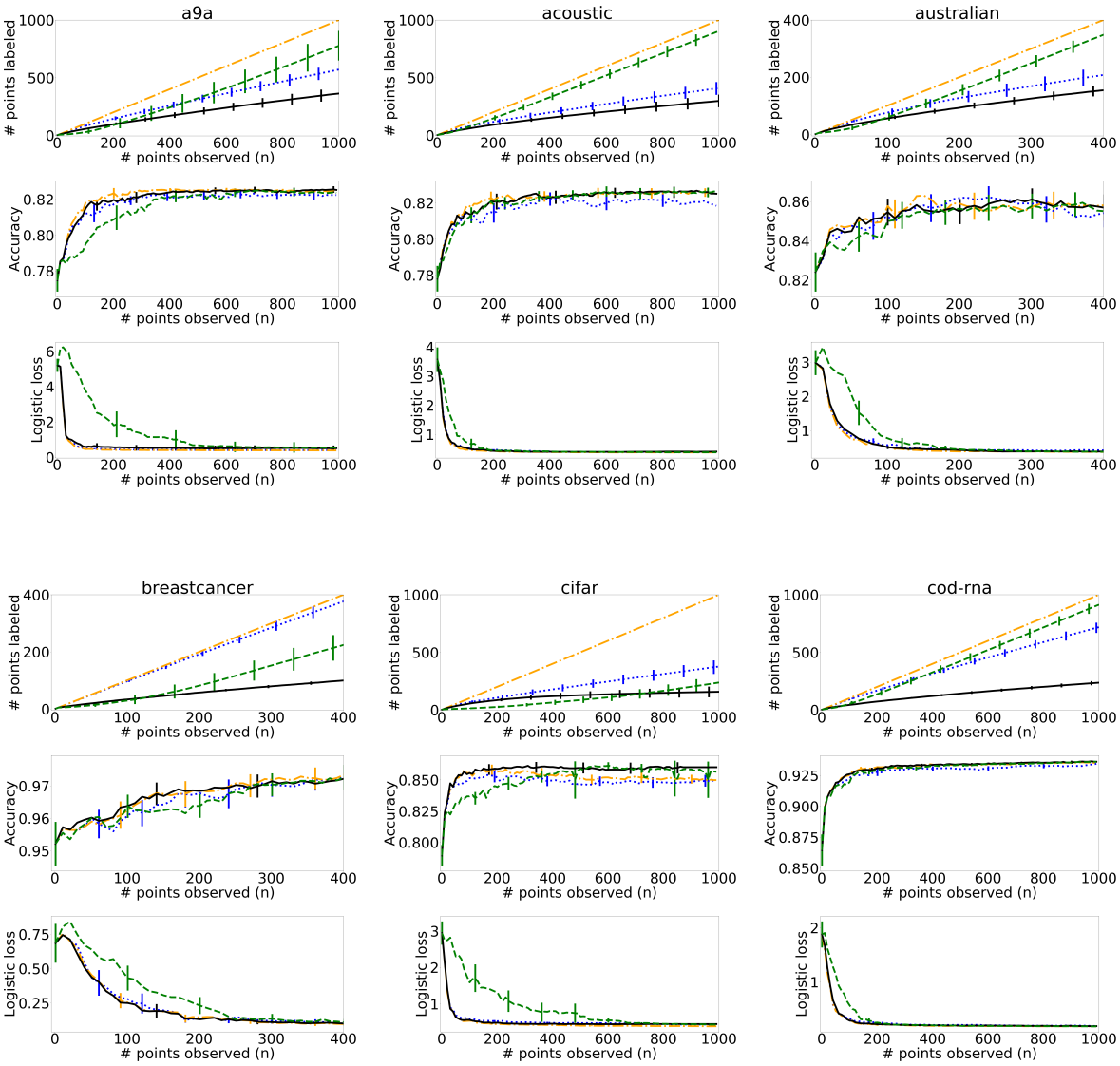

Figure 3: Comparison of the ALPS algorithm, the IWAL algorithm, the ex-post best margin algorithm and the passive learning algorithm for 3-layer networks (1/3). The figures shows the mean over the 50 trials and twice its standard error of the number of labeled points, accuracy and logistic loss on the test set of the model returned by each algorithm as a function of the number of observed points (or rounds) n.