Adaptive Region-Based Active Learning

Corinna Cortes [1] **Giulia DeSalvo** [1] **Claudio Gentile** [1] **Mehryar Mohri** [1 2] **Ningshan Zhang** [3]

Abstract

We present a new active learning algorithm that adaptively partitions the input space into a finite number of regions, and subsequently seeks a distinct predictor for each region. We prove theoretical guarantees for both the generalization error and the label complexity of our algorithm, and analyze the number of regions defined by the algorithm under some mild assumptions. We also report the results of an extensive suite of experiments on several real-world datasets demonstrating substantial empirical benefits over existing single-region and non-adaptive region-based active learning baselines.

1. Introduction

In many learning problems, including document classification, image annotation, and speech recognition, large amounts of unlabeled data are at the learner's disposal at practically no cost. In contrast, reliable labeled data is often more costly to acquire, since it requires careful assessments by human labelers. To limit that cost, in *active learning*, the learner seeks to request as few labels as possible to learn an accurate predictor. This is an attractive learning scenario with significant practical benefits, which remains a challenging theoretical and algorithmic setting.

The literature on active learning is very broad. Thus, we give only a brief discussion of previous work here and refer the reader to (Dasgupta, 2011) for an in-depth survey of the main algorithmic and theoretical ideas, as well as its current challenges. For separable problems, Cohn et al. (1994) introduced the CAL algorithm, which only requires a logarithmic number of label requests, $\log(\frac{1}{\epsilon})$, to achieve ϵ-accuracy. Later, other on-line active learning algorithms for general hypothesis classes and distributions were designed with guarantees both for generalization and label complexity in the agnostic setting (Freund et al., 1997; Balcan et al., 2006; Hanneke, 2007; Dasgupta et al., 2008; Beygelzimer et al., 2009; 2010; Huang et al., 2015; Zhang & Chaudhuri, 2014), and in the separable settings (Dasgupta, 2004; Golovin & Krause, 2017; Nowak, 2011; Tosh & Dasgupta, 2017).

The theoretical analysis of the label complexity of active learning for various hypothesis classes and data distributions has been discussed in several publications (Dasgupta, 2006; Castro & Nowak, 2008; Koltchinskii, 2010; Hanneke & Yang, 2015; Hanneke, 2014; Mussmann & Liang, 2018). In particular, for hypothesis sets consisting of linear separators, a series of publications gave margin-based on-line active learning algorithms that admit guarantees under some specific distributional assumptions (Dasgupta et al., 2005; Balcan et al., 2007; Balcan & Long, 2013; Awasthi et al., 2015; Zhang, 2018).

For all these algorithms, the hypothesis set or *version space* $\mathcal{H}$ is fixed beforehand and, over time, as more labeled information is acquired, it is gradually shrunk to rule out hypotheses too far from the best-in-class hypothesis. This paper initiates the study of an alternative family of algorithms where the hypothesis set $\mathcal{H}$ is first expanded over time before shrinking. Specifically, we consider active learning algorithms that *adaptively* partition the input space into a finite number of disjoint regions, each equipped with the hypothesis set $\mathcal{H}$, and that subsequently seek a distinct predictor for each region. Such algorithms can achieve a substantially better performance, as shown by our theoretical analysis and largely demonstrated by our experiments.

The design of such algorithms raises several questions: How should the input space be partitioned to ensure an improvement in overall performance? How can labels be requested most effectively across regions to learn an accurate predictor per region? Can we provide generalization and label complexity guarantees? In this paper, we tackle these questions and devise an algorithm for this problem, called Adaptive Region-Based Active Learning (ARBAL), benefiting from favorable theoretical guarantees. From a theoretical standpoint, there are several challenging problems: ensuring that the region-specific best-in-class hypothesis is not discarded, the selection of the splitting criteria, and the dependency of

[1]Google Research, New York, NY; [2]Courant Institute of Mathematical Sciences, New York, NY; [3]Hudson River Trading, New York, NY. Correspondence to: Ningshan Zhang <nzhang@stern.nyu.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

the final generalization bound on such criteria.

Of course, if a beneficial partition of the input space is available to the learner, as assumed in the related work of Cortes et al. (2019b), then no further work is needed to adaptively seek one. In practice, however, such strong oracle information may not be available and, even when a natural pre-partitioning of the input space is available, without recourse to labeled data, it is not guaranteed to help improve the generalization error. Furthermore, we will not assume that dividing the input space is always beneficial. However, if there exists indeed a partition such that a region-specific predictor performs significantly better than a global one, then, with high probability, ARBAL will find it. Otherwise, no split is made and ARBAL works just like a single-region active learning algorithm. In practice, in almost all cases we tested, ARBAL splits the input space into multiple regions and achieves a significant performance improvement.

Another line of work somewhat related to our paper is the hierarchical sampling approach of Dasgupta & Hsu (2008) in the *pool-based setting* of active learning, further analyzed by Urner et al. (2013) and Kpotufe et al. (2015), where the learner receives as input a batch of unlabeled points to select from. However, it is important to stress that the methods proposed in those publications rely on (hierarchical) clusterability assumptions of the data that help save labels, while, here, we are more concerned with a problem in model selection for active learning, where splitting the input space is likely to improve generalization rather than reducing label complexity.

In summary, we present an active learning algorithm, ARBAL, that adaptively partitions the input space and performs region-based active learning. Our theoretical results (Theorem 3 and Theorem 9) show that, remarkably, when the algorithm splits the input space into K regions, modulo a standard term in $O(1/\sqrt{T})$ decreasing with the number of rounds T, the generalization error of ARBAL is close to $R^* - \gamma(K-1)$, where R^* is the best-in-class error for the unpartitioned original input space and $\gamma > 0$ a parameter of the algorithm. Thus, when at least one split is made by ARBAL ($K > 1$), then, for T sufficiently large, the error of the algorithm is close to a quantity strictly smaller than the original best-in-class error! Moreover, we show that, under mild theoretical assumptions, ARBAL indeed splits the original input space into multiple subregions (Proposition 4 and Corollary 5). Our experiments confirm that this almost always occurs (Section 5). This significant theoretical improvement over even the original best-in-class error is further corroborated by our extensive experimental study with 25 datasets where, in most cases, ARBAL achieves a better performance than the best active learning algorithm working with the original single region.

The rest of this paper is structured as follows. In Section 2, we introduce the preliminaries relevant to our discussion and give a more formal definition of the learning scenario. In Section 3, we present our new learning algorithm, ARBAL, and justify its splitting criterion via theoretical guarantees. In Section 4, we provide generalization and label complexity bounds for ARBAL in terms of a key parameter for the splitting criterion, and the number of regions partitioned. Moreover, in Section 4.2, we show that, under some natural assumptions about the data distribution, ARBAL benefits from guaranteed improvement over IWAL (Beygelzimer et al., 2009). In Section 5, we report the results of a series of experiments on multiple datasets, demonstrating the substantial benefits of ARBAL over existing non-region-based active learning algorithms, such as IWAL and margin-based uncertainty sampling, and over the non-adaptive region-based active learning baseline ORIWAL (Cortes et al., 2019b).

2. Learning scenario

We now discuss the learning scenario, starting with some preliminary definitions. Let $\mathcal{X} \subseteq \mathbb{R}^D$ denote the input space, $\mathcal{Y} = \{-1, +1\}$ the output space, and $\mathcal{D}$ an unknown distribution over $\mathcal{X} \times \mathcal{Y}$. We denote by $\mathcal{D}_{\mathcal{X}}$ the marginal distribution of $\mathcal{D}$ over $\mathcal{X}$ and, given a prediction space $\mathcal{Z} \subseteq \mathbb{R}$, we denote by $\ell \colon \mathcal{Z} \times \mathcal{Y} \to [0,1]$ a loss function, which we assume to be μ-Lipschitz with respect to its first argument, for some constant $\mu > 0$. Let $\mathcal{H}$ be a family of hypotheses consisting of functions mapping $\mathcal{X}$ to $\mathcal{Z}$. Then, the generalization error or expected loss of a hypothesis $h \in \mathcal{H}$ is denoted by $R(h)$ and defined by $R(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(h(x), y)]$.

We consider the on-line setting of active learning where, at each round $t \in [T] = \{1, \ldots, T\}$, the learner receives as input a point $x_t \in \mathcal{X}$ drawn i.i.d. according to $\mathcal{D}_{\mathcal{X}}$ and must decide to request or not its label y_t. The decision is final and cannot be retroactively changed. At the end of T rounds, the learner returns a hypothesis $\widehat{h}_T \in \mathcal{H}$. In this setting, two conflicting quantities determine the performance of an on-line active learning algorithm: its label complexity, that is, the number of labels it has requested over T rounds, and the generalization error $R(\widehat{h}_T)$ of the hypothesis it returns.

In the standard case where the hypothesis set $\mathcal{H}$ is given beforehand, the learner seeks a single best predictor from $\mathcal{H}$. Here, we consider instead the setup where the algorithm adaptively partitions the input space $\mathcal{X}$ into K regions $\mathcal{X}_1, \ldots, \mathcal{X}_K$, each equipped with a copy of the hypothesis set $\mathcal{H}$ and with K upper-bounded by some parameter $\kappa \geq 1$. Given the partition $\mathcal{X}_1, \ldots, \mathcal{X}_K$, the hypothesis $\widehat{h}_T$ returned by the algorithm after T rounds admits the following form: $\widehat{h}_T(x) = \sum_{k=1}^{K} 1_{x \in \mathcal{X}_k} \widehat{h}_{k,T}(x)$, where $\widehat{h}_{k,T}$ is the hypothesis chosen after T rounds by the algorithm for region $\mathcal{X}_k$.

Let $\mathrm{p}_k = \mathbb{P}(\mathcal{X}_k)$ denote the probability of region $\mathcal{X}_k$ with respect to $\mathcal{D}_{\mathcal{X}}$, $k \in [K]$, and let $R_k(h)$ denote the condi-

tional expected loss of a hypothesis h on region $\mathcal{X}_k$, that is $R_k(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(h(x), y)|x \in \mathcal{X}_k]$. By definition, we have $R(h) = \sum_{k=1}^{K} \mathrm{p}_k R_k(h)$ for any hypothesis h. We assume the learner has access to large amounts of *unlabeled* data, which can be used to accurately estimate p_k. In fact, our results can be easily adapted to the case where the p_ks are estimated via a collection of unlabeled examples requested on-the-fly. While this would not add much to our analysis in terms of technical difficulty, it would make the entire theoretical effort unnecessarily more cluttered.

We denote by $h^* \in \mathcal{H}$ the overall best-in-class hypothesis over $\mathcal{X}$ (single region before any splitting) and by $h_k^* \in \mathcal{H}$ the k-th region's best-in-class hypothesis, that is, $h^* = \operatorname{argmin}_{h\in\mathcal{H}} R(h)$ and $h_k^* = \operatorname{argmin}_{h\in\mathcal{H}} R_k(h)$. We will also use as shorthand the following notation: $R^* = R(h^*)$ and $R_k^* = R_k(h_k^*)$.

Observe that minimizing the generalization error within each region $\mathcal{X}_k$ individually is equivalent to minimizing the overall error over the larger set $\mathcal{H}_{[K]} = \left\{ \sum_{k=1}^{K} 1_{x\in\mathcal{X}_k} h_k(x) \colon h_k \in \mathcal{H} \right\}$. Clearly, the performance of the best predictor in $\mathcal{H}_{[K]}$ is always at least as favorable as that of the best predictor in $\mathcal{H}$, but it can be considerably better, especially when the local performances of h_k^*s with large p_k are substantially superior to that of h^* on the same regions.

3. Algorithm

Our algorithm, called ARBAL (Adaptive Region-Based Active Learning), is an on-line active learning algorithm that *adaptively* partitions the input space into subregions. ARBAL adopts a label requesting policy similar to that of the single-region IWAL algorithm of Beygelzimer et al. (2009), which is based on the largest possible disagreement among the current set of hypotheses on the input: at round t, given the hypothesis set $\mathcal{H}_t$ and input point $x_t \in \mathcal{X}$, IWAL flips a coin $Q_t \in \{0, 1\}$ with bias $p_t = p(x_t)$ defined by

$$p_t = \max_{h,h'\in\mathcal{H}_t, y\in\mathcal{Y}} \ell(h(x_t), y) - \ell(h'(x_t), y).$$

If $Q_t = 1$, then the label of x_t is requested and the algorithm receives y_t, otherwise no label is revealed. Since the loss function ℓ takes values in $[0, 1]$, the requesting probability $p_t \in [0, 1]$ is well defined. IWAL then seeks to shrink the current set $\mathcal{H}_t$ to reduce the querying probability p_t for future inputs, while, at the same time, keeping (with high probability) the overall best-in-class hypothesis h^* in this set. At the end of T rounds, IWAL returns the importance-weighted empirical risk minimizer $\widehat{h}_T = \operatorname{argmin}_{h\in\mathcal{H}_T} \sum_{t=1}^{T} Q_t \ell(h(x_t), y_t)/p_t$.

Our techniques and ideas for splitting are illustrated with IWAL, since IWAL works with any hypothesis set and bounded loss function, and admits generalization guarantees with no distributional assumption. In contrast, CAL (Cohn et al., 1994) assumes a separable case; DHM (Dasgupta et al., 2008) and A^2 (Balcan et al., 2006) are designed for the 0-1 loss, and many other margin-based algorithms only work for linear classifiers. Furthermore, for the separable case ($R^* = 0$), the recent work of (Cortes et al., 2019b) proposes an enhanced version of IWAL, called EIWAL , whose label complexity is on the order of $\log\left(\frac{|\mathcal{H}|}{\epsilon}\right)$, thereby matching the bound of CAL and DHM. Nevertheless, our techniques can be applied to other algorithms, so long as they benefit from theoretical guarantees, such as Corollary 1 of (Dasgupta et al., 2008) or Theorem 1 of (Cortes et al., 2019b). In that case, our splitting criterion can be modified accordingly and our theoretical analysis straightforwardly adapted to such guarantees.

Our algorithm can be viewed as an adaptive region-based version of IWAL, where the label requesting policy and the shrinking procedure of IWAL are applied at the regional level. As already mentioned, the following questions arise when designing the algorithm: (1) How should we determine the regions? (2) Can we learn to adaptively partition the input space into favorable subregions, using actively requested labels? We now explicitly address both questions and describe our algorithm in detail.

The pseudocode of ARBAL is given in Algorithm 1. The algorithm admits two phases: in the first phase (*split phase*), the algorithm partitions the input space into K disjoint regions while actively requesting labels according to IWAL's policy on the regional level. This phase is constrained by two input parameters: κ limits the maximum number of regions generated ($K \leq \kappa$), and τ caps the maximal number of online rounds for this phase. Section 3.1 describes in detail the main subroutine of this phase, SPLIT (Algorithm 2), including the splitting conditions that guarantee a significant improvement in generalization ability resulting from the split. Whenever the algorithm decides to split, each resulting region is equipped with a copy of the original hypothesis set $\mathcal{H}$. Notice that the algorithm actively selects labels in this phase, even if it does not shrink the hypothesis set(s), and thus it still requests fewer labels than passive learning. Thus far, we have not made any assumption about the definition of the subregions. Our splitting criterion is agnostic to their shape, thus any hierarchical partitioning method could be used in the splitting phase. For simplicity, we will adopt the axis-aligned splitting method commonly used for (binary) decision trees, though more convoluted splitting shapes are clearly possible (see Section 5).

In the second phase (IWAL *phase*), ARBAL runs IWAL separately on each of the regions defined by the first phase, to learn a good predictor per region. After T rounds, ARBAL returns $\widehat{h}_T$, which combines region-specific importance-

Algorithm 1 ARBAL$(\mathcal{H}, \tau, \kappa, (\gamma_t)_{t\in[T]})$

$K \leftarrow 1, \mathcal{X}_1 \leftarrow \mathcal{X}, \mathcal{H}_1 \leftarrow \mathcal{H}$
for $t \in [T]$ **do**
  Observe x_t; set $k_t \leftarrow k$ such that $x_t \in \mathcal{X}_k$
  $p_t \leftarrow \max_{h,h'\in\mathcal{H}_{k_t}, y\in\mathcal{Y}} \ell(h(x_t), y_t) - \ell(h'(x_t), y_t)$
  $Q_t \leftarrow$ BERNOULLI(p_t)
  if $Q_t = 1$ **then**
    $y_t \leftarrow$ LABEL(x_t)
  end if
  if $t \le \tau$ and $K < \kappa$ **then**
    $\mathcal{X}_l, \mathcal{X}_r \leftarrow$ SPLIT$(\mathcal{X}_{k_t}, \gamma_t)$ `# split phase`
    if split **then**
      $K \leftarrow K+1, \mathcal{X}_{k_t} \leftarrow \mathcal{X}_l, \mathcal{X}_K \leftarrow \mathcal{X}_r$
      $\mathcal{H}_K \leftarrow \mathcal{H}, \mathcal{H}_{k_t} \leftarrow \mathcal{H}$
    end if
  else
    $\mathcal{H}_{k_t} \leftarrow$ UPDATE$(\mathcal{H}_{k_t})$ `# IWAL phase`
  end if
end for
return $\widehat{h}_T \leftarrow \sum_{k=1}^{K} 1_{x\in\mathcal{X}_k} \widehat{h}_{k,T}$

Algorithm 2 SPLIT$(\mathcal{X}_k, \gamma)$

for $d \in [D]$ and $c \in \mathbb{R}$ **do**
  $(\mathcal{X}_l, \mathcal{X}_r) \leftarrow$ REGSPLIT$(\mathcal{X}_k, d, c)$
  $\gamma_{d,c} \leftarrow \mathrm{p}_k\left[L_{k,t}(\widehat{h}_{k,t}) - L_{k,t}(\widehat{h}_{lr,t}) - \sqrt{\frac{2\sigma_T}{T_{k,t}}}\right]$
end for
$(d^*, c^*) \leftarrow \mathrm{argmax}_{d\in[D], c\in\mathbb{R}} \gamma_{d,c}$
if $\gamma_{d^*,c^*} \ge \gamma$ **then**
  $\mathcal{X}_l^* \leftarrow \{x \in \mathcal{X}_k \colon x[d^*] \le c^*\}$ `# split`
  $\mathcal{X}_r^* \leftarrow \{x \in \mathcal{X}_k \colon x[d^*] > c^*\}$
  return $\mathcal{X}_l^*, \mathcal{X}_r^*$
else
  return $\emptyset$ `# no split`
end if

weighted empirical risk minimizers $\widehat{h}_{k,T}$. In Section 3.2, we describe the IWAL phase, and discuss its connections to ORIWAL (Cortes et al., 2019b).

One may ask why we break up the learning horizon into two phases, where we first determine the partition and next proceed with region-based active learning? Given all possible partitions of the input space, why not running IWAL with the family of hypotheses containing all possible partitions of $\mathcal{X}$ with leaf predictors $h_k \in \mathcal{H}$, that is, $\mathbb{H} = \left\{\sum_{k=1}^{\kappa} 1_{x\in\mathcal{X}_k} h_k \colon h_k \in \mathcal{H}, \cup_{k=1}^{\kappa}\mathcal{X}_k = \mathcal{X}, \mathcal{X}_k \cap \mathcal{X}_{k'} = \emptyset \text{ for } k \ne k'\right\}$? First, $\mathbb{H}$ is an exceedingly complex hypothesis set, whose complexity can lead to vacuous learning guarantees. Second, its computational cost makes it prohibitive to use with IWAL. Moreover, even if we fix the partition and only vary the predictors in the leaf nodes, as proven in Appendix B, running IWAL with $\mathbb{H}$ may cost up to κ times more labels than running IWAL separately within each partitioned region. For all these reasons, we adopt the two-phase learning framework discussed.

3.1. SPLIT phase

The advantage of region-based learning hinges on the improvement in the best-in-class error after each split, which motivates our splitting subroutine: SPLIT splits a region if and only if the best-in-class error is likely to improve by a strictly positive amount. We will show in Corollary 2 that, with high-probability, the best-in-class error is guaranteed to decrease from each split.

The pseudocode of SPLIT is given in Algorithm 2. At time t, SPLIT searches for the most favorable choice of the splitting parameters (d, c) as follows: for a fixed pair (d, c), the algorithm calls subroutine REGSPLIT$(\mathcal{X}_k, d, c)$ to split $\mathcal{X}_k$ into a left region $\mathcal{X}_l$ ($x_d \le c$) and a right region $\mathcal{X}_r$ ($x_d > c$), and then computes a *confidence gap* $\gamma_{d,c}$ as defined in Algorithm 2, where $L_{k,t}(h)$ denotes the importance-weighted empirical risk of hypothesis h on region $\mathcal{X}_k$,

$$L_{k,t}(h) = \frac{1}{T_{k,t}} \sum_{s\in[t], x_s\in\mathcal{X}_k} \frac{Q_s}{p_s}\, \ell(h(x_s), y_s),$$

where $T_{k,t} = |\{s \in [t] \colon x_s \in \mathcal{X}_k\}|$ is the number of samples that have been observed in region $\mathcal{X}_k$ up to time t, and $\sigma_T = \kappa D \log\left[\frac{8T^3|\mathcal{H}|^3\kappa D}{\delta}\right]$ denotes the slack term. Furthermore, $\widehat{h}_{k,t} = \mathrm{argmin}_{h\in\mathcal{H}} L_{k,t}(h)$ denotes the empirical risk minimizer (ERM) on $\mathcal{X}_{k_t}$. Similarly, $\widehat{h}_{l,t}$ and $\widehat{h}_{r,t}$ denote the ERM of region $\mathcal{X}_l$ and $\mathcal{X}_r$, respectively, and $\widehat{h}_{lr,t} = 1_{x\in\mathcal{X}_l}\widehat{h}_{l,t} + 1_{x\in\mathcal{X}_r}\widehat{h}_{r,t}$ is the combination of the two region-specific ERMs. The confidence gap $\gamma_{d,c}$ serves as a conservative estimate of the improvement in the best-in-class error. SPLIT searches for the maximum confidence gap over all distinct pairs: $(d^*, c^*) = \mathrm{argmax}_{d,c}\, \gamma_{d,c}$. When γ_{d^*,c^*} is larger than the pre-specified threshold parameter γ, it splits with (d^*, c^*) and allocates to the two newly created regions the same initial hypothesis set $\mathcal{H}$ (see Algorithm 1), otherwise it does not split.

To implement the SPLIT subroutine, we maintain an array of region labels of past samples, and D sorted arrays of past samples according to each of the D coordinates. At time t, for each coordinate $d \in [D]$, it takes $O(\log(t))$ to insert x_t into the sorted array, and $O(t)$ to compute the key term $L_{k,t}$ for all $t+1$ splitting thresholds on the sorted array. Here, we use the fact that, although there are infinitely many possible splitting threshold values, we only need to consider $t+1$ thresholds to distinguish the t feature values $\{x_{s,d} \colon s \in [t]\}$. It also takes $O(t)$ to update the region labels of past samples after split, and thus a total of $O(tD)$ to run SPLIT at time t. Furthermore, as already mentioned in Section 2, we assume access to a set U of unlabeled samples to estimate all the p_ks. To do so, we maintain a binary tree corresponding to the splits. A new split converts a leaf node u_i with number of elements $|u_i| \le |U|$ into an internal node at the cost of $O(|u_i|)$. The cost of updating the tree for the κ splits in

order to estimate all $\mathtt{p}_k$s is thus $O(\sum_{i=1}^{\kappa}|u_i|)$, where the sum is over all the internal nodes of the tree.

Alternatively, these probabilities can be estimated incrementally during the on-line execution of the algorithm, and our theoretical analysis can be extended along these lines using a union bound similar to the one invoked in the proof of Lemma 1.

We now introduce some additional notation before discussing the theoretical guarantees of the SPLIT algorithm. Let h_k^*, h_l^*, h_r^* be the best-in-class predictors on region $\mathcal{X}_k$, $\mathcal{X}_l$, and $\mathcal{X}_r$, respectively, and denote by $h_{lr}^* = 1_{x\in\mathcal{X}_l}h_l^* + 1_{x\in\mathcal{X}_r}h_r^*$. Then, the improvement in the best-in-class error after this split is $\mathtt{p}_k[R_k(h_k^*) - R_k(h_{lr}^*)]$. The following concentration lemma relates the improvement in the best-in-class error to its empirical counterparts, which leads to the theoretical guarantee for the SPLIT subroutine (Corollary 2). Its proof uses a martingale concentration bound, as well as covering number techniques to guarantee that the high-probability bound holds uniformly for any possible sequence of splitting. The proof is given in Appendix C.

Lemma 1. *With probability at least $1-\delta/4$, for all binary trees with (at most) κ leaf nodes, the improvement in the minimal empirical error by splitting concentrates around the improvement in the best-in-class error:*

$$\left|\left[R_k(h_k^*)-R_k(h_{lr}^*)\right]-\left[L_{k,t}(\widehat{h}_{k,t})-L_{k,t}(\widehat{h}_{lr,t})\right]\right| \le \sqrt{\frac{2\sigma_T}{T_{k,t}}}.$$

Corollary 2. *With probability at least $1-\delta/4$, for all splits made by* ARBAL*, the improvement in the best-in-class error is at least γ_t, where γ_t is the threshold at the split time.*

Corollary 2 guarantees that, with high probability, whenever ARBAL splits, the best-in-class error is strictly improved by at least $\gamma_t > 0$. This yields the fundamental advantage of region-based learning.

One challenge ARBAL faces is that, whenever it chooses to split, it commits to competing against a more accurate predictor, that is the region-specific best-in-class hypothesis on the refined regions. To ensure success, we need to guarantee not only that the best-in-class over the current region or those over subregions after the split are not pruned out, but also that the best-in-class hypothesis over any *future* region produced after further splitting remains in the hypothesis space that will be given as input to ARBAL's second phase.

One can show that, if ARBAL prunes out some hypotheses before the split phase has ended, it may lose the future best-in-class predictor, and thus fail dramatically. As a simple example, consider the binary classification problem depicted in Figure 1, where the unlabeled data is uniformly distributed within a square, and the true classification boundary admits a zig-zag shape (the left plot of Figure 1). If the learner uses the class of linear separators as the initial hypothesis set $\mathcal{H}$, then, after receiving a certain number of labeled samples, it finds that the best performing hypothesis is approximately the diagonal separator from bottom left to top right. Suppose the algorithm would now trim $\mathcal{H}$ to only maintain separators performing similarly to the diagonal separator, with decision surfaces indicated by the shaded area in the middle plot of Figure 1. If later on, the learner splits the input space (the square) into two regions (left and right rectangles in the right plot of Figure 1), then the best-in-class separators for the two rectangles are horizontal separators. Clearly, the two horizontal best-in-class separators are not contained in the current $\mathcal{H}$ (which is meant to apply to the entire input space). In summary, trimming $\mathcal{H}$ before making splits introduces the risk of losing the best-in-class separators on the partitioned regions. This is the reason why ARBAL maintains throughout the split phase the original hypothesis space $\mathcal{H}$. The shrinkage of $\mathcal{H}$ only takes place during the IWAL phase, presented next.

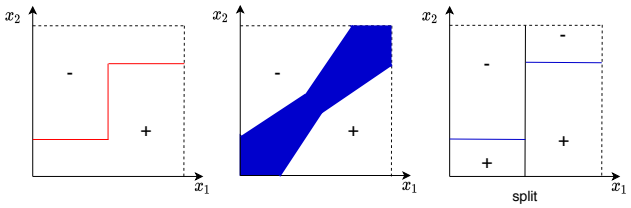

Figure 1: The input space $\mathcal{X}$ is a (2-dimensional) square. Left: the true classification boundary (in red) as a function of x_1 and x_2. Middle: (approximately) the current hypothesis space (shaded blue area) after trimming an initial set of linear separators given several labeled samples. Right: the best-in-class predictors (in blue) when the input space $\mathcal{X}$ splits into two regions $\mathcal{X}_l$ and $\mathcal{X}_r$ from the middle of x_1.

3.2. IWAL phase

In this phase, with the regions $\mathcal{X}_1, \ldots, \mathcal{X}_K$ being fixed, ARBAL runs a separate IWAL subroutine on each one of them, requesting labels and reducing the hypothesis space from $\mathcal{H}$ to region-specific $\mathcal{H}_k, \forall k \in [K]$. As the algorithm requests labels, $\mathcal{H}_k$ shrinks towards the best-in-class hypothesis on region $\mathcal{X}_k$. The hypothesis space is updated according to the IWAL update rule, which is derived from the concentration bound. Specifically, we update the hypothesis space $\mathcal{H}_{k,t}$ sitting on region $\mathcal{X}_k$ at time t by

$$\mathcal{H}_{k,t} = \left\{h \in \mathcal{H}_{k,t-1} : L_{k,t}(h) \le \min_{h\in\mathcal{H}_{k,t-1}} L_{k,t}(h) + \sqrt{\frac{8\sigma_T}{T_{k,t}}}\right\}.$$

To summarize, in this phase, ARBAL freezes the regions $\mathcal{X}_1, \ldots, \mathcal{X}_K$, allowing no further splits, and requests labels and shrinks the set of hypotheses hosted by each such region.

Starting with a fixed partition makes the second phase of ARBAL very similar to the learning scenario recently inves-

tigated by Cortes et al. (2019b), who proposed the ORIWAL algorithm for this learning scenario. In particular, during the second phase, we could also run the ORIWAL algorithm to achieve an additional improvement in generalization error. Since ORIWAL is orthogonal to the main contribution of this paper, we do not further detail this here.

4. Theoretical analysis

In this section, we present generalization error and label complexity guarantees for the ARBAL algorithm. We first need some definitions and concepts from (Beygelzimer et al., 2009). Define the distance $\rho(f,g)$ between two hypotheses $f,g \in \mathcal{H}$ as $\rho(f,g) = \mathbb{E}_{(x,y)\sim\mathcal{D}} |\ell(f(x),y) - \ell(g(x),y)|$.[1] The generalized disagreement coefficient $\theta(\mathcal{D},\mathcal{H})$ of a class of functions $\mathcal{H}$ with respect to distribution $\mathcal{D}$ is defined as the minimum value of θ, such that for all $r > 0$,

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}_X}\left[\sup_{h\in\mathcal{H},\rho(h,h^*)\le r, y\in\mathcal{Y}} \left|\ell(h(x),y) - \ell(h^*(x),y)\right|\right] \le \theta r\,.$$

Since ARBAL calls IWAL as a subroutine, the theoretical results of ARBAL directly depend on those of IWAL, which are summarized in Theorem 6 in Appendix A.

Recall the use of the split threshold γ in Algorithm 2, which is the minimum value of the confidence gap $\gamma_{d,c}$ that allows ARBAL to split a region. We discuss ARBAL under two settings: using a fixed threshold γ, and using a time-varying and data-dependent adaptive threshold γ_t.

4.1. ARBAL with a fixed γ

Suppose we run ARBAL with a fixed threshold γ. The label complexity of the algorithm depends on the region-based disagreement coefficient $\theta_k = \theta(\mathcal{D}_k,\mathcal{H})$, where $\mathcal{D}_k = \mathcal{D}|\mathcal{X}_k$ is defined as the conditional distribution of x on region $\mathcal{X}_k$. Let $\theta_{\max} = \max_{k\in K}\theta_k$ denote the maximum disagreement coefficient across regions, and let $r_0 = \max_{h\in\mathcal{H}}\rho(h,h^*)$. Let $\mathcal{F}_t$ denotes the σ-algebra generated by $(x_1,y_1,Q_1),\ldots,(x_t,y_t,Q_t)$.

Theorem 3. *Assume that a run of* ARBAL *over T rounds has split the input space into K regions. Then, for any $\delta > 0$, with probability at least $1-\delta$, the following inequality holds:*

$$R(\widehat{h}_T) \le R_U + \sqrt{\frac{32K\sigma_T}{T}} + \frac{16K\sigma_T}{T},$$

where $R_U = R^* - \gamma(K-1)$ is an upper bound on the best-in-class error obtained by ARBAL*. Moreover, with probability at least $1-\delta$, the expected number of labels requested, $\tau_T = \sum_{t=1}^{T}\mathbb{E}_{x_t\sim\mathcal{D}_X}\left[p_t|\mathcal{F}_{t-1}\right]$, satisfies*

$$\tau_T \le \min\{2\theta r_0, 1\}\tau + 4\theta_{\max}(T-\tau)\left[R_U + 8\sqrt{\frac{K\sigma_T}{T-\tau}}\right] + \sqrt{32}K\sigma_T.$$

The proof is given in Appendix C. It combines the learning guarantee of IWAL (Theorem 6) with those of splitting (Corollary 2). Theorem 3 shows that, with high probability, the generalization error of the hypothesis returned by ARBAL is close to $R_U = R^* - \gamma(K-1)$, which is more favorable than the single-region best-in class error R^* by $\gamma(K-1)$. We will show later that, under natural assumptions, with high probability, there is at least one split, which implies $\gamma(K-1) > 0$ (Proposition 4). Furthermore, the reduction in the best-in-class error also improves label complexity: when $T \gg \tau$, the label complexity of ARBAL is $O\big(R_U T\big)$ compared to IWAL's $O\big(R^* T\big)$.

In practice, we set $\gamma = \Omega(\sqrt{\sigma_T/T})$ to ensure that the generalization bound in Theorem 3 is more favorable than the generalization bound of IWAL (Theorem 6). We give more details on how to set this fixed γ in Appendix C (see comments following the proof of Theorem 3).

There is a critical trade-off when determining the key parameters τ and κ. With a larger τ and κ, ARBAL is likely to split into more regions and thus admits a smaller R_U. On the other hand, a larger τ means a longer split phase, where ARBAL requests labels more often compared to the original IWAL algorithm since ARBAL does not shrink the hypothesis set $\mathcal{H}$ during this phase. Furthermore, a larger κ yields a larger σ_T, which slightly affects the generalization error. Nevertheless, our experimental results show that larger values of τ and κ almost always improve the final excess risk, at the expense of higher computational cost.

4.2. ARBAL with adaptive γ_t

The learning guarantees of Theorem 3 depend on the number of regions K defined by the algorithm. Given any fixed value of γ, however, there is no guarantee on the number of times ARBAL splits within the first τ rounds (the duration of the first phase). In the worst case when $K = 1$, ARBAL offers no improvement over IWAL, yet ARBAL requests more labels than IWAL during the initial τ rounds.

In this section, we show that by adopting a time-varying and data-dependent splitting threshold γ_t, we can enable SPLIT to split more often, and thus achieve an enhanced performance guarantee. To do so, we make additional assumptions on the potential gain of splitting.

Let $\mathcal{X}_k$ be an intermediate region created during the split phase, possibly the original input space $\mathcal{X}$. Assume that for any such $\mathcal{X}_k$, there exists at least one way of splitting $\mathcal{X}_k$

[1] This definition of $\rho(f,g)$ slightly differs from the original definition in (Beygelzimer et al., 2009), and it improves the label complexity bound of IWAL by a constant. See Appendix A for more details.

into $\mathcal{X}_l \cup \mathcal{X}_r$ such that the improvement in the conditional best-in-class error is at least ρ: $R_k(h_k^*) - R_k(h_{lr}^*) \geq \rho$, where $\rho > 0$ is a positive constant. With this assumption, we can derive upper bounds on the time ARBAL splits when run with a time-varying adaptive $\gamma_t = \mathbb{P}(\mathcal{X}_{k_t})\rho/2$.

Proposition 4. *Let* ARBAL *be run with* $\gamma_t = \rho\,\mathbb{P}(\mathcal{X}_{k_t})/2$. *Then, for any* $\delta > 0$, *with probability at least* $1 - \delta/2$, *the first split occurs before round* $\left\lceil 2\sigma_T\left(\frac{4}{\rho}+1\right)^2\right\rceil$.

Thus, when $\tau \geq \left\lceil 2\sigma_T\left(\frac{4}{\rho}+1\right)^2\right\rceil$, with high probability, ARBAL will split $\mathcal{X}$ for the first time, and reduce the best-in-class error by at least $\rho/2$, according to Proposition 4 and Corollary 2. In Appendix C, we prove a more general version (Lemma 11) that upper bounds the time of split for all regions created during the split phase.

If we further assume that the splitting with at least ρ improvement in the conditional best-in-class error results in regions that are not too small, i.e., $\min\{\mathbb{P}(\mathcal{X}_l),\, \mathbb{P}(\mathcal{X}_r)\} \geq c\,\mathbb{P}(\mathcal{X}_k)$, with $0 < c < 0.5$, then we can also prove a lower bound on the number of splits.

Corollary 5. *Let* ARBAL *run with* $\gamma_t = \mathbb{P}(\mathcal{X}_{k_t})\rho/2$. *Then, with probability at least* $1 - \delta/2$, ARBAL *splits more than* $\min\left\{\log_{1/c}\left[\frac{\tau}{2\sigma_T\left(\frac{4}{\rho}+1\right)^2}\right], \kappa - 1\right\}$ *times by the end of the split phase.*

Corollary 5 gives the minimal number of splits under the assumptions made in this section. It states that, as the duration of the split phase τ increases, or as the conditional improvement ρ increases, or as the minimal proportion of subregion size c increases, ARBAL tends to make more splits and therefore achieves a better generalization guarantee. The lower bound in Corollary 5 tends to be loose, as it assumes that ARBAL keeps splitting the smallest region, which is unlikely to be the case in practice. In Appendix C, we combine Proposition 4 and Corollary 5 to give an upper bound on the final best-in-class error after the splits by ARBAL.

Note that the true value of ρ is the property of the underlying distribution, and to accurately estimate ρ is an open question that is beyond the scope of this paper. One practical solution is to explore ρ on various orders of magnitude, for instance $(0.1, 0.01, 0.001)$, such that ARBAL makes a reasonable number of splits. We set $\rho = 0.01$ in our experiments.

5. Experiments

In this section, we report the results of a series of experiments. We tested 24 binary classification datasets from the UCI and openml repositories, and also the MNIST dataset with 3 and 5 as the two classes, which is standard binary classification task extracted from the MNIST dataset (e.g., (Crammer et al., 2009)). Table 1 in Appendix D lists summary statistics for these datasets. For ease of experimental comparison, for datasets with large input dimension D, we followed the preprocessing step in (Cortes et al., 2019b), retaining only the first 10 principal components of the original feature vectors. Due to space limitations, in this section we show the results on several medium-sized datasets. The results for the remaining datasets are provided in Appendix D. For each experiment, we randomly shuffled the dataset, ran the algorithms on the first half of the data (so that the number of active learning rounds T equals $N/2$), and tested the classifier returned on the remaining half to measure misclassification loss. We only showed results on the first $10^{3.5} \approx 3000$ requested labels, which are enough to differentiate the performances among various algorithms. We repeated this process 50 times on each dataset, and report average results with standard error across the 50 repetitions. We use the logistic loss function ℓ defined for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and hypotheses $h\colon \mathcal{X} \to \mathbb{R}$ by $\ell(h(x), y) = \log(1 + e^{-yh(x)})$, which we then rescale to $[0, 1]$. The initial hypothesis set $\mathcal{H}$ consists of 3,000 randomly drawn hyperplanes with bounded norms. As mentioned in Section 4.1, larger values of τ and κ almost always yield better final excess risk. We chose $\kappa = 20$ and allow the first phase to run at most $\tau = 800$ rounds so as to make ARBAL fully split into the desired number of regions on almost all datasets. Since the slack term σ_T derived from high-probability analyses are typically overly conservative, we simply use $0.01/\sqrt{T_k}$ in the SPLIT subroutine.

ARBAL with fixed or adaptive γ. We first compare ARBAL with fixed γ to ARBAL with an adaptive γ_t. Figure 2 plots the misclassification loss versus the number of labels requested on four datasets. The vertical lines indicate the label counts when ARBAL transits from the first to the second phase, and the legends give the average number of resulting regions K the algorithms produce. As one can see, adaptive γ_t tends to split into more regions and to exit the split phase earlier, and hence often results in superior prediction performance over fixed γ. Thus, in the rest of this section, we show the performance of adaptive γ_t. Results on other datasets (see Appendix D) show similar patterns. During the active learning split phase, even though ARBAL does not shrink the hypothesis set(s), both versions are observed to request labels in only 50% - 90% of the rounds, which is far less than passive learning.

ARBAL VS. ORIWAL. Since the key idea of ARBAL is the informed adaptive splitting criterion, we compare ARBAL with the ORIWAL algorithm of Cortes et al. (2019b), a "non-adaptive splitting" algorithm that first randomly generates κ regions, and then runs region-based active learning on these regions. The regions of ORIWAL are obtained from terminal nodes of random binary trees, that is, binary trees with random splitting coordinates and thresholds (hence, they are axis-aligned rectangles, as for ARBAL). Figure 3 compares

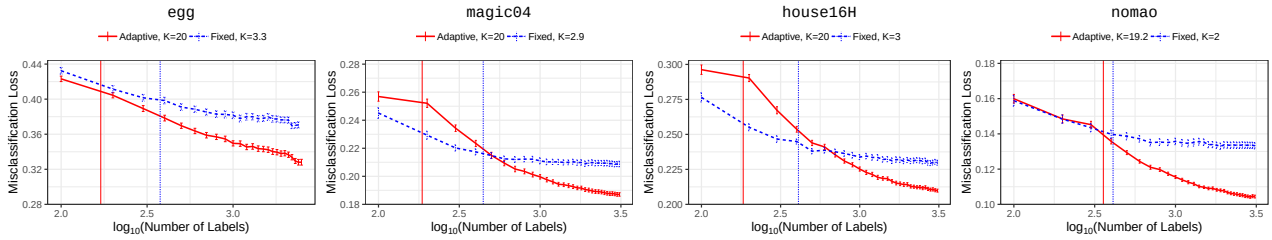

Figure 2: Misclassification loss of ARBAL with fixed and adaptive threshold γ on held out test data vs. number of labels requested ($\log_{10}$ scale), with $\kappa = 20$ and $\tau = 800$. The vertical lines indicate the end of the first (split) phase.

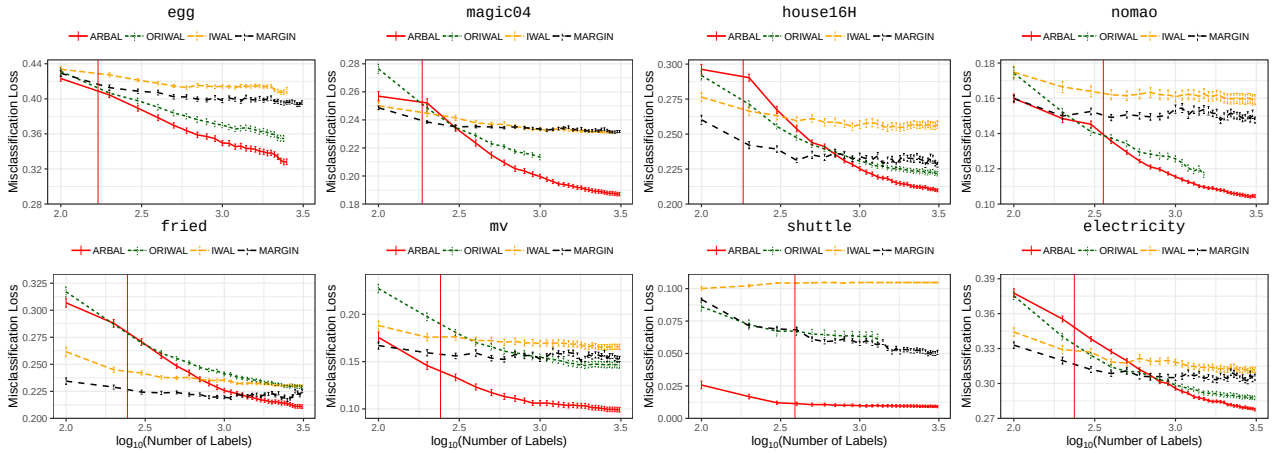

Figure 3: Misclassification loss of ARBAL(with adaptive γ_t), ORIWAL, IWAL, and MARGIN on hold out test data vs. number of labels requested ($\log_{10}$ scale), with $\kappa = 20$ and $\tau = 800$. The ARBAL curves are repetitions from Figure 2.

ARBAL of adaptive γ_t with ORIWAL, and shows that ARBAL quickly takes over (recall that the x axis is on log scale) and performs substantially better than ORIWAL, on eight datasets covered by Figure 3. Results on other datasets show similar patterns, even though ORIWAL sometimes uses more regions than ARBAL, since ARBAL may not always fully split into κ regions. These results empirically verify the advantage of an adaptive splitting criterion.

ARBAL vs. non-splitting baselines. We also compare ARBAL with the single-region IWAL algorithm, and the single-region MARGIN algorithm, which is a standard uncertainty sampling algorithm that requests the label closest to the decision boundary of the current empirical risk minimizer (note that MARGIN runs under a pool-based setting and thus sees more information than on-line algorithms).

Figure 3 shows that MARGIN is a strong baseline that outperforms IWAL on almost all the datasets, sometimes even ORIWAL (e.g. `house16H`), but ARBAL is still more favorable than MARGIN. The difference of errors observed in these plots after consuming much of the sample is essentially due to the difference of the split-region and single-region best-in-class errors, that is, R_U vs. R^*, which further corroborates our theory. The results for most other datasets show similar patterns. In fact, ARBAL can also be used with the MARGIN algorithm as a subroutine, which is likely to lead to even better performance but, as with the MARGIN algorithm, that extension would not benefit from any general theoretical guarantee and might actually underperform in some instances where the MARGIN technique can fail.

Finally, as mentioned in Section 3.1, ARBAL is agnostic to the shape of subregions. For instance, we can split a region via an arbitrary separating hyperplane or via hierarchical clustering, that is, determine two new centers and assign points to the closest center. In Appendix D, we compare axis-aligned binary tree splitting method with hierarchical clustering splitting, using adaptive γ_t. Our results suggest that, for most datasets, splitting via binary trees is more favorable than via hierarchical clustering.

6. Conclusion

We presented a novel algorithm for adaptive region-based active learning, and proved that it benefits from favorable generalization and label complexity guarantees. We also studied the extent to which splitting the input space is likely to lead to improved prediction performance. We complemented our theoretical findings by reporting the results of several experiments with our algorithm on standard benchmarks. Our extensive experiments demonstrate substantial

performance improvements over existing active learning algorithms such as IWAL and margin-based uncertainty sampling, as well as other region-based baselines that do not rely on adaptively splitting of the input space.

Our techniques were showcased through IWAL-like algorithms (Beygelzimer et al., 2009; Cortes et al., 2019b), but they can be straightforwardly combined with other base active learning algorithms, such as the DHM algorithm of Dasgupta et al. (2008), achieving similar generalization and label complexity guarantees. They can also be combined with the margin algorithm and result in even more substantial improvements in practice, as with the combination of ORIWAL with the margin algorithm in (Chuang et al., 2019). However, such algorithms currently do not benefit from proven theoretical guarantees, due to the lack of general guarantees for the margin algorithm.

Altogether, our theory, algorithms, and empirical results provide a new promising solution to active learning, with very important practical benefits. These results also suggest further investigation of the general idea of adaptively refining and enriching the hypothesis set for active learning. From the theoretical standpoint, this work illustrates an significant specific instance of model selection in active learning that benefits from generalization and label complexity guarantees.

Our ARBAL algorithm uses a greedy splitting criterion that maximizes the benefit of the current split, which does not consider the potential benefit of subsequent splits. One direction for future research is to come up with a less greedy, but perhaps computationally more intensive splitting algorithm, possibly via lookahead search techniques, that yields a even better partitioning with smaller best-in-class error. Both IWAL and ARBAL can be extended to multi-class classification tasks with a bounded loss, such as a margin loss, and enjoy similar learning guarantees. When the number of classes is very large, however, the current algorithm may not be sufficiently efficient and is a topic of active research as in standard multi-class classification and structured prediction theory (**?**). Our algorithm and techniques can also be extended to other learning tasks such as regression and ranking using similar methods.

Acknowledgements

This work was partly supported by NSF CCF-1535987, NSF IIS-1618662, and a Google Research Award. Much of NZ's research was done during her Ph.D. studies at New York University. We thank anonymous reviewers for their helpful comments and suggestions.

References

Awasthi, P., Balcan, M.-F., Haghtalab, N., and Urner, R. Efficient learning of linear separators under bounded noise. In *Proceedings of COLT*, pp. 167–190, 2015.

Balcan, M.-F. and Long, P. Active and passive learning of linear separators under log-concave distributions. In *Proceedings of COLT*, pp. 288–316, 2013.

Balcan, M.-F., Beygelzimer, A., and Langford, J. Agnostic active learning. In *Proceedings of ICML*, 2006.

Balcan, M.-F., Broder, A., and Zhang, T. Margin based active learning. In *International Conference on Computational Learning Theory*, pp. 35–50. Springer, 2007.

Beygelzimer, A., Dasgupta, S., and Langford, J. Importance weighted active learning. In *Proceedings of ICML*, pp. 49–56. ACM, 2009.

Beygelzimer, A., Hsu, D. J., Langford, J., and Zhang, T. Agnostic active learning without constraints. In *Proceedings of NIPS*, pp. 199–207, 2010.

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

Chuang, G., DeSalvo, G., Karydas, L., Kagy, J., Rostamizadeh, A., and Theeraphol, A. Active learning empirical study. In *NeurIPS2019 LIRE Workshop*, 2019.

Cohn, D., Atlas, L., and Ladner, R. Improving generalization with active learning. *Machine learning*, 15(2): 201–221, 1994.

Cortes, C., DeSalvo, G., Gentile, C., Mohri, M., and Zhang, N. Active learning with disagreement graphs. In *Proceedings of ICML*, 2019a.

Cortes, C., DeSalvo, G., Gentile, C., Mohri, M., and Zhang, N. Region-based active learning. In *Proceedings of AISTATS 2019*, 2019b.

Crammer, K., Kulesza, A., and Dredze, M. Adaptive regularization of weight vectors. In *Nips*, 2009.

Dasgupta, S. Analysis of a greedy active learning strategy. In *Advances in neural information processing systems*, pp. 337–344, 2004.

Dasgupta, S. Coarse sample complexity bounds for active learning. In *Proceedings of NIPS*, pp. 235–242, 2006.

Dasgupta, S. Two faces of active learning. *Theor. Comput. Sci.*, 412(19):1767–1781, 2011.

Dasgupta, S. and Hsu, D. Hierarchical sampling for active learning. In *Proceedings of ICML*, pp. 208–215. ACM, 2008.

Dasgupta, S., Kalai, A. T., and Monteleoni, C. Analysis of perceptron-based active learning. In *International Conference on Computational Learning Theory*, pp. 249–263. Springer, 2005.

Dasgupta, S., Hsu, D. J., and Monteleoni, C. A general agnostic active learning algorithm. In *Proceedings of NIPS*, pp. 353–360, 2008.

Freund, Y., Seung, H. S., Shamir, E., and Tishby, N. Selective sampling using the query by committee algorithm. *Machine learning*, 28(2-3):133–168, 1997.

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

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

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

Hanneke, S. and Yang, L. Minimax analysis of active learning. *The Journal of Machine Learning Research*, 16(1): 3487–3602, 2015.

Huang, T.-K., Agarwal, A., Hsu, D., Langford, J., and E. Schapire, R. Efficient and parsimonious agnostic active learning. In *Proceedings of NIPS*, 2015.

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

Kpotufe, S., Urner, R., and Ben-David, S. Hierarchical label queries with data-dependent partitions. In *Proceedings of COLT*, pp. 1176–1189, 2015.

Mussmann, S. and Liang, P. On the relationship between data efficiency and error for uncertainty sampling. In *PMLR 80: Proceedings of the 35th International Conference on Machine Learning*, pp. 3674–3682, 2018.

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

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

Urner, R., Wulff, S., and Ben-David, S. PLAL: Cluster-based active learning. In *Proceedings of COLT*, pp. 376–397, 2013.

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

Zhang, C. and Chaudhuri, K. Beyond disagreement-based agnostic active learning. In *Proceedings of NIPS*, pp. 442–450, 2014.