The True Sample Complexity of Active Learning

Maria-Florina Balcan
Computer Science Department
Carnegie Mellon University
ninamf@cs.cmu.edu

Steve Hanneke
Machine Learning Department
Carnegie Mellon University
shanneke@cs.cmu.edu

Jennifer Wortman
Computer and Information Science
University of Pennsylvania
wortmanj@seas.upenn.edu

Abstract

We describe and explore a new perspective on the sample complexity of active learning. In many situations where it was generally believed that active learning does not help, we show that active learning does help in the limit, often with exponential improvements in sample complexity. This contrasts with the traditional analysis of active learning problems such as non-homogeneous linear separators or depth-limited decision trees, in which $\Omega(1/\epsilon)$ lower bounds are common. Such lower bounds should be interpreted carefully; indeed, we prove that it is always possible to learn an ϵ-good classifier with a number of samples asymptotically smaller than this. These new insights arise from a subtle variation on the traditional definition of sample complexity, not previously recognized in the active learning literature.

1 Introduction

Machine learning research has often focused on the problem of learning a classifier from labeled examples sampled independent from the particular learning algorithm that is used. However, for many contemporary practical problems such as classifying web pages or detecting spam, there is often an abundance of *unlabeled* data available, from which a relatively small subset is selected to be labeled and used for learning. In such scenarios, the question arises of how to select that subset of examples to be labeled.

One possibility, which has recently been generating substantial interest, is *active learning*. In active learning, the learning algorithm itself is allowed to select the subset of unlabeled examples to be labeled. It does this sequentially (i.e., interactively), using the requested label information from previously selected examples to inform its decision of which example to select next. The hope is that by only requesting the labels of informative examples, the algorithm can learn a good classifier using significantly fewer labels than would be required if the labeled set were sampled at random.

A number of active learning analyses have recently been proposed in a PAC-style setting, both for the realizable and for the agnostic cases, resulting in a sequence of important positive and negative results [6, 7, 8, 2, 10, 4, 9, 13, 12]. In particular, the most concrete noteworthy positive result for when active learning helps is that of learning homogeneous (i.e., through the origin) linear separators, when the data is linearly separable and distributed uniformly over the unit sphere, and this example has been extensively analyzed [8, 2, 10, 4, 9]. However, few other positive results are known, and there are simple (almost trivial) examples, such as learning intervals or non-homogeneous linear separators under the uniform distribution, where previous analyses of sample complexities have indicated that perhaps active learning does not help at all [8].

In this work, we approach the analysis of active learning algorithms from a different angle. Specifically, we point out that traditional analyses have studied the number of label requests required before an algorithm can both produce an ϵ-good classifier *and* prove that the classifier's error is no more than ϵ. These studies have turned up simple examples where this number is no smaller than the number of random labeled examples required for passive learning. This is the case for learning certain nonhomogeneous linear separators and intervals on the real line, and generally seems to be a common problem for many learning scenarios. As such, it has led some to conclude that active learning *does not help* for most learning problems. One of the goals of our present analysis is to dispel this misconception. Specifically, we study the number of labels an algorithm needs to request before it can produce an ϵ-good classifier, even if there is no accessible confidence bound available to verify the quality of the classifier. With this type of analysis, we prove that active learning can essentially always achieve asymptotically superior sample complexity compared to passive learning when the VC dimension is finite. Furthermore, we find that for most natural learning problems, including the negative examples given in the previous literature, active learning can achieve exponential[1] improvements over passive learning with respect to dependence on ϵ. This situation is characterized in Figure 1.1.

1.1 A Simple Example: Intervals

To get some intuition about when these types of sample complexity are different, consider the following example. Suppose that C is the class of all intervals over $[0, 1]$ and D is

[1] We slightly abuse the term "exponential" throughout the paper. In particular, we refer to any $polylog(1/\epsilon)$ as being an exponential improvement over $1/\epsilon$.

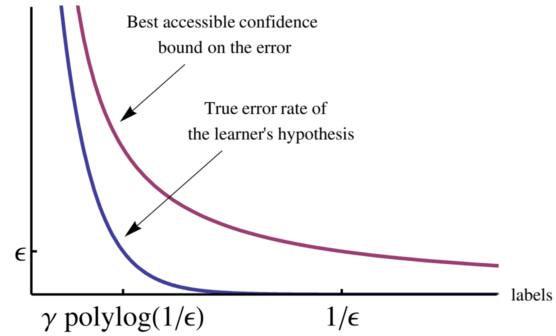

Figure 1.1: Active learning can often achieve exponential improvements, though in many cases the amount of improvement cannot be detected from information available to the learning algorithm. Here γ may be a target-dependent constant.

a uniform distribution over $[0, 1]$. If the target function is the empty interval, then for any sufficiently small ϵ, in order to *verify* with high confidence that this (or any) interval has error $\leq \epsilon$, we need to request labels in at least a constant fraction of the $\Omega(1/\epsilon)$ intervals $[0, \epsilon], [\epsilon, 2\epsilon], \ldots$, requiring $\Omega(1/\epsilon)$ total label requests.

However, no matter what the target function is, we can *find* an ϵ-good classifier with only a logarithmic sample complexity via the following extremely simple 2-phase learning algorithm. We start with a large ($\Omega(1/\epsilon)$) set of unlabeled examples. In the first phase, on each round we choose a point x uniformly at random from the unlabeled sample and query its label. We repeat this until we observe the first $+1$ label, at which point we enter the second phase. In the second phase, we alternate between running one binary search on the examples between 0 and that x and a second on the examples between that x and 1 to approximate the end-points of the interval. At the end, we output a smallest interval consistent with the observed positive labels.

If the target h^* labels every point as -1 (the so-called *all-negative* function), the algorithm described above would output a hypothesis with 0 error even after 0 label requests. On the other hand, if the target is an interval $[a, b] \subseteq [0, 1]$, where $b - a = w > 0$, then after roughly $O(1/w)$ queries (a constant number that depends only on the target), a positive example will be found. Since only $O(\log(1/\epsilon))$ queries are required to run the binary search to reach error rate ϵ, the sample complexity is at worst logarithmic in $1/\epsilon$. Thus, we see a sharp distinction between the sample complexity required to *find* a good classifier (logarithmic) and the sample complexity needed to both find a good classifier *and verify* that it is good.

This example is particularly simple, since there is effectively only *one* "hard" target function (the all-negative target). However, most of the spaces we study are significantly more complex than this, and there are generally many targets for which it is difficult to achieve good verifiable complexity.

Our Results: We show that in many situations where it was previously believed that active learning cannot help, active learning does help in the limit. Our main specific contributions are as follows:

- We distinguish between two different variations on the definition of sample complexity. The traditional definition, which we refer to as *verifiable sample complexity*, focuses on the number of label requests needed to obtain a confidence bound indicating an algorithm has achieved at most ϵ error. The newer definition, which we refer to simply as *sample complexity*, focuses on the number of label requests before an algorithm actually achieves at most ϵ error. We point out that the latter is often significantly smaller than the former, in contrast to passive learning where they are often equivalent up to constants for most nontrivial learning problems.

- We prove that *any* distribution and finite VC dimension concept class has active learning sample complexity asymptotically smaller than the sample complexity of passive learning for nontrivial targets. A simple corollary of this is that finite VC dimension implies $o(1/\epsilon)$ active learning sample complexity.

- We show it is possible to actively learn with an *exponential rate* a variety of concept classes and distributions, many of which are known to require a linear rate in the traditional analysis of active learning: for example, intervals on $[0, 1]$ and non-homogeneous linear separators under the uniform distribution.

- We show that even in this new perspective, there do exist lower bounds; it is possible to exhibit somewhat contrived distributions where exponential rates are not achievable even for some simple concept spaces (see Theorem 12). The learning problems for which these lower bounds hold are much more intricate than the lower bounds from the traditional analysis, and intuitively seem to represent the core of what makes a hard active learning problem.

2 Background and Notation

Let $\mathcal{X}$ be an instance space and $\mathcal{Y} = \{-1, 1\}$ be the set of possible labels. Let C be the hypothesis class, a set of measurable functions mapping from $\mathcal{X}$ to $\mathcal{Y}$, and assume that C has VC dimension d. We consider here the realizable setting in which it is assumed that the instances are labeled by a target function h^* in the class C. The *error rate* of a hypothesis h with respect to a distribution D over $\mathcal{X}$ is defined as $\mathrm{er}(h) = \mathbb{P}_D(h(x) \neq h^*(x))$.

We assume the existence of an infinite sequence $x_1, x_2, \ldots$ of examples sampled i.i.d. according to D. The learning algorithm may access any finite initial segment $x_1, x_2, \ldots, x_m$. Essentially, this means we allow the algorithm access to an arbitrarily large, but finite, sequence of random unlabeled examples. In active learning, the algorithm can select any example x_i, and request the label $h^*(x_i)$ that the target assigns to that example, observing the labels of all previous requests before selecting the next example to query. The goal is to find a hypothesis h with small error with respect to D, while simultaneously minimizing the number of label requests that the learning algorithm makes.

2.1 Two Definitions of Sample Complexity

The following definitions present a subtle but significant distinction we refer to throughout the paper. Several of the results that follow highlight situations where these two definitions of sample complexity can have dramatically different dependence on ϵ.

Definition 1 *A function $S(\epsilon, \delta, h^*)$ is a* verifiable sample complexity *for a pair (C, D) if there exists an active learning algorithm $A(t, \delta)$ that outputs both a classifier h_t* and *a value $\hat{\epsilon}_t \in \mathbb{R}$ after making at most t label requests, such that for any target function $h^* \in C, \epsilon \in (0, 1/2), \delta \in (0, 1/4)$, for any $t \geq S(\epsilon, \delta, h^*)$,*

$$\mathbb{P}_D(er(h_t) \leq \hat{\epsilon}_t \leq \epsilon) \geq 1 - \delta.$$

Definition 2 *A function $S(\epsilon, \delta, h^*)$ is a* sample complexity *for a pair (C, D) if there exists an active learning algorithm $A(t, \delta)$ that outputs a classifier h_t after making at most t label requests, such that for any target function $h^* \in C, \epsilon \in (0, 1/2), \delta \in (0, 1/4)$, for any $t \geq S(\epsilon, \delta, h^*)$,*

$$\mathbb{P}_D(er(h_t) \leq \epsilon) \geq 1 - \delta.$$

Note that both types of sample complexity can be target-dependent and distribution-dependent. The only distinction is whether or not there is an accessible guarantee on the error of the chosen hypothesis that is also at most ϵ. This confidence bound can only depend on quantities accessible to the learning algorithm, such as the t requested labels. Thus, any verifiable sample complexity function is also a sample complexity function, but we study a variety of cases where the reverse is not true. In situations where there are sample complexity functions significantly smaller than any achievable verifiable sample complexities, we sometimes refer to the smaller quantity as the *true sample complexity* to distinguish it from the verifiable sample complexity.

A common alternative formulation of verifiable sample complexity is to let A take ϵ as an argument and allow it to choose online how many label requests it needs in order to guarantee error at most ϵ [8]. This alternative definition is essentially equivalent (either definition can be reduced to the other without significant loss), as the algorithm must be able to produce a confidence bound of size at most ϵ on the error of its hypothesis in order to decide when to stop requesting labels anyway.[2]

2.2 The Verifiable Sample Complexity

To date, there has been a significant amount of work studying the verifiable sample complexity (though typically under the aforementioned alternative formulation). It is clear from standard results in passive learning that verifiable sample complexities of $O\left((d/\epsilon)\log(1/\epsilon) + (1/\epsilon)\log(1/\delta)\right)$ are easy to obtain for any learning problem, by requesting the labels of random examples. As such, there has been much interest in determining when it is possible to achieve verifiable sample complexity *smaller* than this, and in particular, when the verifiable sample complexity is a polylogarithmic function of $1/\epsilon$ (representing exponential improvements over passive learning).

One of the earliest active learning algorithms in this model is the selective sampling algorithm of Cohn, Atlas, and Ladner [6], henceforth referred to as CAL. This algorithm keeps track of two spaces—the current *version space* C_i, defined as the set of hypotheses in C consistent with all labels revealed so far, and the current *region of uncertainty* $R_i = \{x \in \mathcal{X} : \exists h_1, h_2 \in C_i \text{ s.t. } h_1(x) \neq h_2(x)\}$. In each round i, the algorithm picks a random unlabeled example from R_i and requests its label, eliminating all hypotheses in C_i inconsistent with the received label to make the next version space C_{i+1}. The algorithm then defines R_{i+1} as the region of uncertainty for the new version space C_{i+1} and continues. Its final hypothesis can then be taken arbitrarily from C_t, the final version space, and we use the diameter of C_t for the $\hat{\epsilon}_t$ error bound.

While there are a small number of cases in which this algorithm and others have been shown to achieve exponential improvements in the verifiable sample complexity for all targets (most notably, the case of homogeneous linear separators under the uniform distribution), there exist extremely simple concept classes for which $\Omega(1/\epsilon)$ labels are needed for some targets. For example, consider the class of intervals in $[0, 1]$ under the uniform distribution. In order to distinguish the all-negative target from the set of hypotheses that are positive on a region of weight ϵ and make a high probability guarantee, $\Omega(1/\epsilon)$ labeled examples are needed [8].

Recently, there have been a few quantities proposed to measure the verifiable sample complexity of active learning on any given concept class and distribution. Dasgupta's *splitting index* [8], which is dependent on the concept class, data distribution, target function, and a parameter τ, quantifies how easy it is to make progress toward reducing the diameter of the version space by choosing an example to query. Another quantity to which we will frequently refer is Hanneke's *disagreement coefficient* [12], defined as follows.

Definition 3 *For any $h \in C$ and $r > 0$, let $B(h, r)$ be a ball of radius r around h in C. That is,*

$$B(h, r) = \{h' \in C : \mathbb{P}_D(h(x) \neq h'(x)) \leq r\}\ .$$

For any hypothesis class C, define the region of disagreement *as*

$$\mathrm{DIS}(C) = \{x \in \mathcal{X} : \exists h_1, h_2 \in C : h_1(x) \neq h_2(x)\}\ .$$

Additionally, let $\bar{C}$ denote any countable dense subset of C.[3] *For our purposes, the* disagreement coefficient *of a hypothesis h, denoted θ_h, is defined as*

$$\theta_h = \sup_{r>0} \frac{\mathbb{P}(\mathrm{DIS}(\bar{B}(h, r)))}{r}.$$

[2]There is some question as to what the "right" formal model of active learning is in general. For instance, we could instead let A generate an infinite sequence of h_t hypotheses (or $(h_t, \hat{\epsilon}_t)$ in the verifiable case), where h_t can depend only on the first t label requests made by the algorithm along with some initial segment of unlabeled examples (as in [5]), representing the case where we are not sure a-priori of when we will stop the algorithm. However, for our present purposes, such alternative models are equivalent in sample complexity up to constants.

[3]That is, $\bar{C}$ is countable and $\forall h \in C, \forall \epsilon > 0, \exists h' \in \bar{C} : \mathbb{P}(h(X) \neq h'(X)) \leq \epsilon$. Such a subset exists, for example, in any C with finite VC dimension. We introduce this countable dense subset to avoid certain degenerate behaviors, such as when $\mathrm{DIS}(B(h, 0)) = \mathcal{X}$.

The disagreement coefficient for a concept space C *is defined as* $\theta = \sup_{h \in C} \theta_h$.

The disagreement coefficient is often a useful quantity for analyzing the verifiable sample complexity of active learning algorithms. For example, it has been shown that the algorithm of Cohn, Atlas, and Ladner described above achieves a verifiable sample complexity at most $\theta_{h^*} d \cdot \text{polylog}(1/(\epsilon\delta))$ when run with concept class $\bar{C}$ for target function $h^* \in C$ [12]. We will see that both the disagreement coefficient and splitting index are also useful quantities for analyzing true sample complexities, though their use in that case is less direct.

2.3 The True Sample Complexity

This paper focuses on situations where true sample complexities are significantly smaller than verifiable sample complexities. In particular, we show that many common pairs (C, D) have sample complexity that is polylogarithmic in *both* $1/\epsilon$ and $1/\delta$ and linear only in some finite target-dependent constant γ_{h^*}. This contrasts sharply with the infamous $1/\epsilon$ lower bounds mentioned above, which have been identified for verifiable sample complexity. The implication is that, for any fixed target h^*, such lower bounds vanish as ϵ approaches 0. This also contrasts with passive learning, where $1/\epsilon$ lower bounds are typically unavoidable [1].

Definition 4 *We say that (C, D) is actively learnable at an exponential rate if there exists an active learning algorithm achieving sample complexity*

$$S(\epsilon, \delta, h^*) = \gamma_{h^*} \cdot \text{polylog}\left(1/(\epsilon\delta)\right)$$

for some finite $\gamma_{h^*} = \gamma(h^*, D)$ independent of ϵ and δ.

3 Strict Improvements of Active Over Passive

In this section, we describe conditions under which active learning can achieve a sample complexity asymptotically superior to passive learning. The results are surprisingly general, indicating that whenever the VC dimension is finite, essentially *any* passive learning algorithm is asymptotically *dominated* by an active learning algorithm on *all* targets.

Definition 5 *A function $S(\epsilon, \delta, h^*)$ is a* passive learning *sample complexity for a pair (C, D) if there exists an algorithm $A(((x_1, h^*(x_1)), (x_2, h^*(x_2)), \ldots, (x_t, h^*(x_t))), \delta)$ that outputs a classifier h_t, such that for any target function $h^* \in C, \epsilon \in (0, 1/2), \delta \in (0, 1/4)$, for any $t \geq S(\epsilon, \delta, h^*)$,*

$$\mathbb{P}_D(er(h_t) \leq \epsilon) \geq 1 - \delta.$$

Thus, a passive learning sample complexity corresponds to a restriction of an active learning sample complexity to algorithms that specifically request the first t labels in the sequence and ignore the rest. In particular, it is known that for any finite VC dimension class, there is always an $O(1/\epsilon)$ passive learning sample complexity [14]. Furthermore, this is often tight (though not always), in the sense that for any passive algorithm, there exist targets for which the corresponding passive learning sample complexity is $\Omega(1/\epsilon)$ [1]. The following theorem states that for any passive learning sample complexity, there exists an achievable active learning sample complexity with a strictly slower asymptotic rate of growth. Its proof is included in Appendix D.

Theorem 6 *Suppose C has finite VC dimension, and let D be any distribution on $\mathcal{X}$. For any passive learning sample complexity $S_p(\epsilon, \delta, h)$ for (C, D), there exists an active learning algorithm achieving a sample complexity $S_a(\epsilon, \delta, h)$ such that, for all targets $h \in C$ for which $S_p(\epsilon, \delta, h) = \omega(1)$,*[4]

$$S_a(\epsilon, \delta, h) = o\left(S_p(\epsilon/4, \delta, h)\right).$$

In particular, this implies the following simple corollary.

Corollary 7 *For any C with finite VC dimension, and any distribution D over $\mathcal{X}$, there is an active learning algorithm that achieves a sample complexity $S(\epsilon, \delta, h)$ such that*

$$S(\epsilon, \delta, h) = o\left(1/\epsilon\right)$$

for all targets $h \in C$.

Proof: Let d be the VC dimension of C. The passive learning algorithm of Haussler, Littlestone & Warmuth [14] is known to achieve a sample complexity no more than $(kd/\epsilon)\log(1/\delta)$, for some universal constant $k < 200$ [14]. Applying Theorem 6 now implies the result. ■

Note the interesting contrast, not only to passive learning, but also to the known results on the *verifiable* sample complexity of active learning. This theorem definitively states that the $\Omega(1/\epsilon)$ lower bounds common in the literature on verifiable samples complexity can *never* arise in the analysis of the true sample complexity of finite VC dimension classes.

4 Composing Hypothesis Classes

Recall the simple example of learning the class of intervals over $[0, 1]$ under the uniform distribution. It is well known that the verifiable sample complexity of the "all-negative" classifier in this class is $\Omega(1/\epsilon)$. However, consider the more limited class $C_1 \subset C$ containing only the intervals h with $w(h) = \mathbb{P}(h(X) = +1) > 0$. Using the simple algorithm described in Section 1.1, this restricted class can be learned with a (verifiable) sample complexity of only $O(1/w(h) + \log(1/\epsilon))$. Furthermore, the remaining set of classifiers $C_2 = C \setminus C'$ (which consists of only the all-negative classifier) has sample complexity 0. Thus, $C = C_1 \cup C_2$, and both (C_1, D) and (C_2, D) are learnable at an exponential rate.

It turns out that it is often convenient to view concept classes in terms of such well-constructed, possibly infinite sequences of subsets. Generally, given a distribution D and a function class C, suppose we can construct a sequence of subclasses, $C_1, C_2, \ldots$, where $C = \cup_{i=1}^{\infty} C_i$, such that it is possible to actively learn any subclass C_i with only

[4] Recall that we say a non-negative function $\phi(\epsilon) = o(1/\epsilon)$ iff $\lim_{\epsilon \to 0} \phi(\epsilon)/(1/\epsilon) = 0$. Similarly, $\phi(\epsilon) = \omega(1)$ iff $\lim_{\epsilon \to 0} 1/\phi(\epsilon) = 0$. Here and below, the $o(\cdot)$, $\omega(\cdot)$, $\Omega(\cdot)$ and $O(\cdot)$ notation should be interpreted as $\epsilon \to 0$ (from the $+$ direction), treating all other parameters (e.g., δ and h^*) as fixed constants. Note that any algorithm achieving a sample complexity $S_p(\epsilon, \delta, h) \neq \omega(1)$ is guaranteed, with probability $\geq 1 - \delta$, to achieve error zero using a finite number of samples, and therefore we cannot hope to achieve a slower asymptotic growth in sample complexity.

$S_i(\epsilon, \delta, h)$ sample complexity. Thus, if we know that the target h^* is in C_i, it is straightforward to guarantee $S_i(\epsilon, \delta, h^*)$ sample complexity. However, it turns out it is also possible to learn with sample complexity $O(S_i(\epsilon/2, \delta/2, h^*))$ even without this information. This can be accomplished by using an aggregation algorithm.

We describe a simple algorithm for aggregation below in which multiple algorithms are run on different subclasses C_i in parallel and we select among their outputs by comparisons. Within each subclass C_i we run an active learning algorithm A_i, such as Dasgupta's splitting algorithm [8] or CAL, with some sample complexity $S_i(\epsilon, \delta, h)$.

Algorithm 1 The Aggregation Procedure. Here it is assumed that $C = \cup_{i=1}^{\infty} C_i$, and that for each i, A_i is an algorithm achieving sample complexity at most $S_i(\epsilon, \delta, h)$ for the pair (C_i, D). The procedure takes t and δ as parameters.

Let k be the largest integer s.t. $k^2 \lceil 72 \ln(4k/\delta) \rceil \leq t/2$
for $i = 1, \ldots, k$ **do**
  Let h_i be the output of running $A_i(\lfloor t/(4i^2) \rfloor, \delta/2)$ on the sequence $\{x_{2n-1}\}_{n=1}^{\infty}$
end for
for $i, j \in \{1, 2, \ldots, k\}$ **do**
  if $\mathbb{P}(h_i(X) \neq h_j(X)) > 0$ **then**
    Let R_{ij} be the first $\lceil 72 \ln(4k/\delta) \rceil$ elements in the sequence $\{x_{2n}\}_{n=1}^{\infty}$ for which $h_i(x) \neq h_j(x)$
    Request the labels of all examples in R_{ij}
    Let m_{ij} be the number of elements in R_{ij} on which h_i makes a mistake
  else
    Let $m_{ij} = 0$
  end if
end for
Return $\hat{h}_t = h_i$ where $i = \underset{i \in \{1,2,\ldots,k\}}{\operatorname{argmin}} \max_{j \in \{1,2,\ldots,k\}} m_{ij}$

Using this algorithm, we can show the following sample complexity bound. The proof appears in Appendix A.

Theorem 8 *For any distribution D, let $C_1, C_2, \ldots$ be a sequence of classes such that for each i, the pair (C_i, D) has sample complexity at most $S_i(\epsilon, \delta, h)$ for all $h \in C_i$. Let $C = \cup_{i=1}^{\infty} C_i$. Then (C, D) has a sample complexity at most*

$$\min_{i: h \in C_i} \max \left\{ 4i^2 \lceil S_i(\epsilon/2, \delta/2, h) \rceil, 2i^2 \left\lceil 72 \ln \frac{4i}{\delta} \right\rceil \right\},$$

for any $h \in C$. In particular, Algorithm 1 achieves this, when used with the A_i algorithms that each achieve the $S_i(\epsilon, \delta, h)$ sample complexity.

A particularly interesting implication of Theorem 8 is that, if we can decompose C into a sequence of classes C_i such that each (C_i, D) is learnable at an exponential rate, then this procedure achieves exponential rates. Since it is more abstract and it allows us to use known active learning algorithms as a black box, we often use this compositional view throughout the remainder of the paper. In particular, since the verifiable sample complexity of active learning is presently much better understood in the existing literature, it will often be useful to use this result in combination with an algorithm with a known bound on its *verifiable* sample complexity. As the following theorem states, at least for the case of exponential rates, this approach of constructing algorithms with good true sample complexity by reduction to algorithms with known verifiable complexity on subspaces loses nothing in generality. The proof is included in Appendix B.

Theorem 9 *For any (C, D) learnable at an exponential rate, there exists a sequence $C_1, C_2, \ldots$ with $C = \cup_{i=1}^{\infty} C_i$, and a sequence of active learning algorithms $A_1, A_2, \ldots$ such that the algorithm A_i achieves* verifiable *sample complexity at most $\gamma_i polylog_i(1/(\epsilon\delta))$ for the pair (C_i, D). Thus, the aggregation algorithm (Algorithm 1) achieves exponential rates when used with these algorithms.*

Note that decomposing a given C into a sequence of C_i subsets that have good verifiable sample complexities is not always a simple task. One might be tempted to think a simple decomposition based on increasing values of verifiable sample complexity with respect to (C, D) would be sufficient. However, this is not always the case, and generally we need to use information more detailed than verifiable complexity with respect to (C, D) to construct a good decomposition. We have included in Appendix C a simple heuristic approach that can be quite effective, and in particular yields good sample complexities for every (C, D) described in Section 5.

5 Exponential Rates

The results in Section 3 tell us that the sample complexity of active learning can be made strictly superior to any passive learning sample complexity when the VC dimension is finite. We now ask how much better that sample complexity can be. In particular, we describe a number of concept classes and distributions that are learnable at an *exponential* rate, many of which are known to require $\Omega(1/\epsilon)$ *verifiable* sample complexity.

5.1 Exponential rates for simple classes

We begin with a few simple observations, to point out situations in which exponential rates are trivially achievable; in fact, in each of the cases mentioned in this subsection, the sample complexity is actually $O(1)$.

Clearly if $|\mathcal{X}| < \infty$ or $|C| < \infty$, we can always achieve exponential rates. In the former case, we may simply request the label of every x in the support of D, and thereby perfectly identify the target. The corresponding $\gamma = |\mathcal{X}|$. In the latter case, for every pair $h_1, h_2 \in C$ such that $\mathbb{P}(h_1(X) \neq h_2(X)) > 0$, we may request the label of any x_i such that $h_1(x_i) \neq h_2(x_i)$, and there will be only one (up to measure zero differences) $h \in C$ that gets all of these examples correct: namely, the target function. So in this case, we learn with an exponential rate with $\gamma = |C|^2$.

Less obvious is the fact that this argument extends to any *countably infinite* hypothesis class C. In particular, in this case we can list the classifiers in C: $h_1, h_2, \ldots$. Then we define the sequence $C_i = \{h_i\}$, and simply use Algorithm 1. By Theorem 8, this gives an algorithm with sample complexity $S(\epsilon, \delta, h_i) = 2i^2 \lceil 72 \ln(4i/\delta) \rceil = O(1)$.

5.2 Geometric Concepts, Uniform Distribution

Many interesting geometric concepts in $\mathbb{R}^n$ are learnable at an exponential rate if the underlying distribution is uniform on some subset of $\mathbb{R}^n$. Here we provide some examples; interestingly, every example in this subsection has some targets for which the *verifiable* sample complexity is $\Omega(1/\epsilon)$. As we see in Section 5.3, all of the results in this section can be extended to many other types of distributions as well.

Unions of k intervals under arbitrary distributions: Let $\mathcal{X}$ be the interval $[0, 1)$ and let $C^{(k)}$ denote the class of unions of at most k intervals. In other words, $C^{(k)}$ contains functions described by a sequence $\langle a_0, a_1, \cdots, a_\ell\rangle$, where $a_0 = 0$, $a_\ell = 1$, $\ell \leq 2k + 1$, and $a_0, \cdots, a_\ell$ is the (nondecreasing) sequence of transition points between negative and positive segments (so x is labeled $+1$ iff $x \in [a_i, a_{i+1})$ for some *odd* i). For any distribution, this class is learnable at an exponential rate, by the following decomposition argument. First, let

$$C_1 = \{h \in C^{(k)} : \mathbb{P}(h(X) = +1) = 0\}.$$

That is, C_1 contains the all-negative function, or any function that is equivalent given the distribution D. For $i = 2, 3, \ldots, k + 1$, inductively define

$$C_i = \{h \in C^{(k)} : \exists h' \in C^{(i-1)} \text{ s.t. } \mathbb{P}(h(X) \neq h'(X)) = 0\} \setminus \cup_{j<i} C_j.$$

In other words, C_i contains all of the functions that can be represented as unions of $i - 1$ intervals but cannot be represented as unions of fewer intervals. Clearly C_1 has verifiable sample complexity 0. For $i > 1$, within each subclass C_i, the disagreement coefficient is bounded by something proportional to $k + 1/w(h)$, where

$$w(h) = \min\{\mathbb{P}([a_j, a_{j+1})) : 0 \leq j < \ell, \mathbb{P}([a_j, a_{j+1})) > 0\}$$

is the weight of the smallest positive or negative interval and $\langle a_0, a_1, \cdots, a_\ell\rangle$ is the sequence of transition points corresponding to this h. Thus, running CAL with $\bar{C}_i$ achieves polylogarithmic (verifiable) sample complexity for any $h \in C_i$. Since $C^{(k)} = \cup_{i=1}^{k+1} C_i$, by Theorem 8, $C^{(k)}$ is learnable at an exponential rate.

Ordinary Binary Classification Trees: Let $\mathcal{X}$ be the cube $[0, 1]^n$, D be the uniform distribution on $\mathcal{X}$, and C be the class of binary decision trees using a finite number of axis-parallel splits (see e.g., Devroye et al. [11], Chapter 20). In this case, (similarly to the previous example) we let C_i be the set of decision trees in C distance zero from a tree with i leaf nodes, not contained in any C_j for $j < i$. For any i, the disagreement coefficient for any $h \in C_i$ (with respect to (C_i, D)) is a finite constant, and we can choose $\bar{C}_i$ to have finite VC dimension, so each (C_i, D) is learnable at an exponential rate (by running CAL with $\bar{C}_i$), and thus by Theorem 8, (C, D) is learnable at an exponential rate.

5.2.1 Linear Separators

Theorem 10 *Let C be the hypothesis class of linear separators in n dimensions, and let D be the uniform distribution over the surface of the unit sphere. The pair (C, D) is learnable at an exponential rate.*

Proof: (Sketch) There are multiple ways to achieve this. We describe here a simple proof that uses a decomposition as follows. Let $\lambda(h)$ be the probability mass of the minority class under hypothesis h. C_1 contains only the separators h with $\lambda(h) = 0$, and $C_2 = C \setminus C_1$. As before, we can use a black box active learning algorithm such as CAL to learn within each class C_i. To prove that we indeed get the desired exponential rate of active learning, we show that the disagreement coefficient of any separator h with respect to (C, D) is at most $\propto \sqrt{n}/\lambda(h)$. Hanneke's results concerning the CAL algorithm [12] then imply that C_2 is learnable at an exponential rate. Since C_1 trivially has sample complexity 1, combined with Theorem 8, this would imply the result.

We describe the key steps involved in computing the disagreement coefficient. First we can show that for any two linear separators $h(x) = \text{sign}(w \cdot x + b)$ and $h'(x) = \text{sign}(w' \cdot x + b')$, we can lower bound the distance between them as

$$\mathbb{P}(h(X) \neq h'(X)) \geq \max\left\{|\lambda - \lambda'|, \frac{2\alpha}{\pi}\min\{\lambda, \lambda'\}\right\},$$

where $\alpha = arccos(w \cdot w')$ is the angle between w and w', λ is the probability mass of the minority class under h, and λ' is the probability mass of the minority class under h'. Assume for now that h and h' are close enough together to have the same minority class; it's not necessary, but simplifies things.

We are now ready to compute the disagreement coefficient. Assume $r < \lambda/\sqrt{n}$. From the previous claim we have

$$B(h, r) \subseteq \left\{h' : \max\left\{|\lambda - \lambda'|, \frac{2\alpha}{\pi}\min\{\lambda, \lambda'\}\right\} \leq r\right\}$$

where $B(h, r)$ is the ball of radius r around h in the hypothesis space. The region of disagreement of the set on the left is contained within

$$\text{DIS}\left(\{h' : w' = w \wedge |\lambda' - \lambda| \leq r\}\right) \cup \text{DIS}\left(\left\{h' : \frac{2\alpha}{\pi}(\lambda - r) \leq r \wedge |\lambda - \lambda'| = r\right\}\right).$$

By some trigonometry, we can show this region is contained within

$$\text{DIS}(\{h' : w' = w \wedge |\lambda' - \lambda| \leq r\}) \cup \left\{x : |w \cdot x + b_1| \leq c\frac{r}{\lambda}\right\} \cup \left\{x : |w \cdot x + b_2| \leq c\frac{r}{\lambda}\right\}$$

for some constants b_1, b_2, c. Using previous results [2, 12], it is possible to show that the measure of this region is at most $2r + c'(\sqrt{n}/\lambda)r = c''(\sqrt{n}/\lambda)r$. This finally implies that for any target function, the disagreement coefficient is at most $c''(\sqrt{n}/\lambda)$, where λ is the probability of the minority class of the target function. ■

5.3 Composition results

We can also extend the results from the previous subsection to other types of distributions and concept classes in a variety of ways. Here we include a few results to this end.

Close distributions: If (C, D) is learnable at an exponential rate, then for any distribution D' such that for all measurable

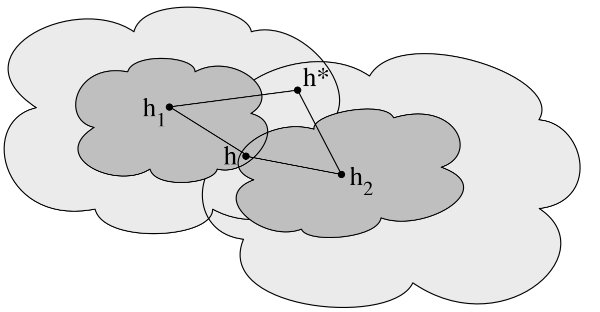

Figure 5.1: Illustration of the proof of Theorem 11. The dark gray regions represent $B_{D_1}(h_1, 2r)$ and $B_{D_2}(h_2, 2r)$. The function h that gets returned is in the intersection of these. The light gray regions represent $B_{D_1}(h_1, \epsilon/3)$ and $B_{D_2}(h_2, \epsilon/3)$. The target function h^* is in the intersection of these. We therefore must have $r \le \epsilon/3$, and by the triangle inequality $\mathrm{er}(h) \le \epsilon$.

$A \subseteq \mathcal{X}$, $\lambda \mathbb{P}_D(A) \le \mathbb{P}_{D'}(A) \le (1/\lambda)\mathbb{P}_D(A)$ for some $\lambda \in (0, 1]$, (C, D') is also learnable at an exponential rate. In particular, we can simply use the algorithm for (C, D), filter the examples from D' so that they appear like examples from D, and then any t large enough to find an $\epsilon\lambda$-good classifier with respect to D is large enough to find an ϵ-good classifier with respect to D'.

A composition theorem for mixtures of distributions: Suppose there exist algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ for learning a class C at an exponential rate under distributions D_1 and D_2 respectively. It turns out we can also learn under any *mixture* of D_1 and D_2 at an exponential rate, by using $\mathcal{A}_1$ and $\mathcal{A}_2$ as black boxes. In particular, the following theorem relates the sample complexity under a mixture to the sample complexities under the mixing components.

Theorem 11 *Let C be an arbitrary hypothesis class. Assume that the pairs (C, D_1) and (C, D_2) have sample complexities $S_1(\epsilon, \delta, h^*)$ and $S_2(\epsilon, \delta, h^*)$ respectively, where D_1 and D_2 have density functions $\mathbb{P}_{D_1}$ and $\mathbb{P}_{D_2}$ respectively. Then for any $\alpha \in [0, 1]$, the pair $(C, \alpha D_1 + (1-\alpha) D_2)$ has sample complexity at most $2\lceil \max\{S_1(\epsilon/3, \delta/2, h^*), S_2(\epsilon/3, \delta/2, h^*)\}\rceil$.*

Proof: If $\alpha = 0$ or 1 then the theorem statement holds trivially. Assume instead that $\alpha \in (0, 1)$. We describe an algorithm in terms of α, D_1, and D_2, which achieves this sample complexity bound.

Suppose algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ achieve the stated sample complexities under D_1 and D_2 respectively. At a high level, the algorithm we define works by "filtering" the distribution over input so that it appears to come from two streams, one distributed according to D_1, and one distributed according to D_2, and feeding these filtered streams to $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively. To do so, we define a random sequence $u_1, u_2, \cdots$ of independent uniform random variables in $[0, 1]$. We then run $\mathcal{A}_1$ on the sequence of examples x_i from the unlabeled data sequence satisfying

$$u_i < \frac{\alpha \mathbb{P}_{D_1}(x_i)}{\alpha \mathbb{P}_{D_1}(x_i) + (1-\alpha)\mathbb{P}_{D_2}(x_i)},$$

and run $\mathcal{A}_2$ on the remaining examples, allowing each to make an equal number of label requests.

Let h_1 and h_2 be the classifiers output by $\mathcal{A}_1$ and $\mathcal{A}_2$. Because of the filtering, the examples that $\mathcal{A}_1$ sees are distributed according to D_1, so after $t/2$ queries, the current error of h_1 with respect to D_1 is, with probability $1 - \delta/2$, at most $\inf\{\epsilon' : S_1(\epsilon', \delta/2, h^*) \le t/2\}$. A similar argument applies to the error of h_2 with respect to D_2.

Finally, let

$$r = \inf\{r : B_{D_1}(h_1, r) \cap B_{D_2}(h_2, r) \ne \emptyset\}.$$

Define the output of the algorithm to be any $h \in B_{D_1}(h_1, 2r) \cap B_{D_2}(h_2, 2r)$. If a total of $t \ge 2\lceil \max\{S_1(\epsilon/3, \delta/2, h^*), S_2(\epsilon/3, \delta/2, h^*)\}\rceil$ queries have been made ($t/2$ by $\mathcal{A}_1$ and $t/2$ by $\mathcal{A}_2$), then by a union bound, with probability at least $1 - \delta$, h^* is in the intersection of the $\epsilon/3$-balls, and so h is in the intersection of the $2\epsilon/3$-balls. By the triangle inequality, h is within ϵ of h^* under both distributions, and thus also under the mixture. (See Figure 5.1 for an illustration of these ideas.) ■

5.4 Lower Bounds

Given the previous discussion, one might suspect that *any* pair (C, D) is learnable at an exponential rate, under some mild condition such as finite VC dimension. However, we show in the following that this is *not* the case, even for some simple geometric concept classes when the distribution is especially nasty.

Theorem 12 *There exists a pair (C, D), with the VC dimension of C equal 1, that is not learnable at an exponential rate (in the sense of Definition 4).*

Proof: (Sketch) Let T be a fixed infinite tree in which each node at depth i has c_i children; c_i is defined shortly. We consider learning the hypothesis class C where each $h \in C$ corresponds to a path down the tree starting at the root; every node along this path is labeled 1 while the remaining nodes are labeled -1. Clearly for each $h \in C$ there is precisely one node on each level of the tree labeled 1 by h (i.e. one node at each depth d). C has VC dimension 1 since knowing the identity of the node labeled 1 on level i is enough to determine the labels of all nodes on levels $0, \ldots, i$ perfectly. This learning problem is depicted in Figure 5.2.

Now we define D, a "bad" distribution for C. Let ℓ_i be the total probability of all nodes on level i according to D. Assume all nodes on level i have the same probability according to D, and call this p_i. By definition, we have $p_i = \ell_i / \prod_{j=0}^{i-1} c_j$.

We show that it is possible to define the parameters above in such a way that for any $\epsilon_0 > 0$, there exists some $\epsilon < \epsilon_0$ such that for some level j, $p_j = \epsilon$ and $c_{j-1} \ge (1/p_j)^{1/2} = (1/\epsilon)^{1/2}$. This implies that $\Omega(1/\epsilon^{1/2})$ labels are needed to learn with error less than ϵ, for the following reason. We know that there is exactly one node on level j that has label 1, and that any successful algorithm must identify this node (or have a lucky guess at which one it is) since it has probability ϵ. By the usual probabilistic method trick (picking the target at random by choosing the positive node at each level $i+1$ uniformly from the children of the positive at level i), we

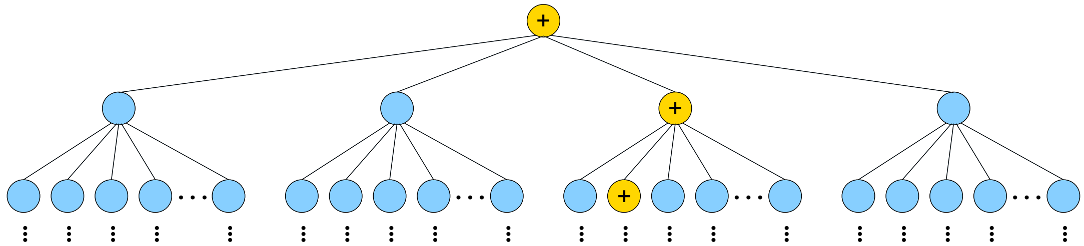

Figure 5.2: A learning problem where exponential rates are not achievable. The instance space is an infinite-depth tree. The target labels nodes along a single infinite path as $+1$, and labels all other nodes -1. When the number of children and probability mass of each node at each subsequent level are set in a certain way, sample complexities of $o\left(1/\sqrt{\epsilon}\right)$ are not achievable.

can argue that in order to label that node positive with at least some constant probability, we need to query at least a constant fraction of the node's siblings, so we need to query on the order of c_{j-1} nodes on level j.

Thus it is enough to show that we can define the values above such that for all i, $c_{i-1} \geq (1/p_i)^{1/2}$, and such that p_i gets arbitrarily small as i gets big.

To start, notice that if we recursively define the values of c_i as $c_i = \prod_{j=0}^{i-1} c_j/\ell_{i+1}$ then

$$c_{i-1}^2 = c_{i-1}\left(\frac{\prod_{j=0}^{i-2} c_j}{\ell_i}\right) = \frac{\prod_{j=0}^{i-1} c_j}{\ell_i} = \frac{1}{p_i}$$

and $c_{i-1} \geq (1/p_i)^{1/2}$ as desired.

To enforce that p_i gets arbitrarily small as i gets big, we simply need to set ℓ_i appropriately. In particular, we need $\lim_{i\to\infty} \ell_i/\prod_{j=0}^{i-1} c_j = 0$. Since the denominator is increasing in i, it suffices to show $\lim_{i\to\infty} \ell_i = 0$. Defining the values of ℓ_i to be any positive probability distribution over i that goes to 0 in the limit completes the proof. ■

For essentially any function $\phi = o\left(1/\epsilon\right)$, the tree example in the proof can be modified to construct a pair (C, D) with the VC dimension of C equal to 1 such that no algorithm achieves $o(\phi(\epsilon))$ sample complexity for all targets: simply choose $c_i = \lfloor \phi(p_{i+1}) \rfloor$, where $\{p_j\}$ is any sequence strictly decreasing to 0 s.t. $p_{i+1}\phi(p_{i+1})\prod_{j<i} c_j \leq \ell_{i+1}$ and $\phi(p_{i+1}) \geq 1$, where as before $\{\ell_j\}$ is any sequence of positive values summing to 1; we can (arbitrarily) assign any left-over probability mass to the root node; $\phi = o(1/\epsilon)$ guarantees that such a $\{p_j\}$ sequence exists for any $\phi = \omega(1)$. Thus, the $o\left(1/\epsilon\right)$ guarantee of Corollary 7 is in some sense the tightest guarantee we can make at that level of generality, without using a more detailed description of the structure of the problem beyond the finite VC dimension assumption.

This type of example can be realized by certain nasty distributions, even for a variety of simple hypothesis classes: for example, linear separators in $\mathbb{R}^2$ or axis-aligned rectangles in $\mathbb{R}^2$. We remark that this example can also be modified to show that we cannot expect intersections of classifiers to preserve exponential rates. That is, the proof can be extended to show that there exist classes C_1 and C_2, such that both (C_1, D) and (C_2, D) are learnable at an exponential rate, but (C, D) is not, where $C = \{h_1 \cap h_2 : h_1 \in C_1, h_2 \in C_2\}$.

6 Discussion and Open Questions

The implication of our analysis is that in many interesting cases where it was previously believed that active learning could not help, it turns out that active learning *does help asymptotically*. We have formalized this idea and illustrated it with a number of examples and general theorems throughout the paper. This realization dramatically shifts our understanding of the usefulness of active learning: while previously it was thought that active learning could *not* provably help in any but a few contrived and unrealistic learning problems, in this alternative perspective we now see that active learning essentially *always* helps, and does so significantly in all *but* a few contrived and unrealistic problems.

The use of decompositions of C in our analysis also generates another interpretation of these results. Specifically, Dasgupta [8] posed the question of whether it would be useful to develop active learning techniques for looking at unlabeled data and "placing bets" on certain hypotheses. One might interpret this work as an answer to this question; that is, some of the decompositions used in this paper can be interpreted as reflecting a preference partial-ordering of the hypotheses, similar to ideas explored in the passive learning literature [16, 15, 3]. However, the construction of a good decomposition in active learning seems more subtle and quite different from previous work in the context of supervised or semi-supervised learning.

It is interesting to examine the role of target- and distribution-dependent constants in this analysis. As defined, both the verifiable and true sample complexities may depend heavily on the particular target function and distribution. Thus, in both cases, we have interpreted these quantities as fixed when studying the asymptotic growth of these sample complexities as ϵ approaches 0. It has been known for some time that, with only a few unusual exceptions, any target- and distribution-independent bound on the verifiable sample complexity could typically be no better than the sample complexity of passive learning; in particular, this observation lead Dasgupta to formulate his splitting index bounds as both target- and distribution-dependent [8]. This fact also applies to bounds on the true sample complexity as well. Indeed, the entire distinction between verifiable and true sample complexities collapses if we remove the dependence on these unobservable quantities.

There are many interesting open problems within this framework. Perhaps two of the most interesting are formulating general necessary and sufficient conditions for learnability at an exponential rate, and determining whether Theorem 6 can be extended to the agnostic case.

Acknowledgments: We thank Eyal Even-Dar, Michael Kearns, and Yishay Mansour for numerous useful discussions and for helping us to initiate this line of thought. We are also grateful to Larry Wasserman and Eric Xing for their helpful feedback.
Maria-Florina is supported in part by an IBM Graduate Fellowship and by a Google Research Grant. Steve is funded by the NSF grant IIS-0713379 awarded to Eric Xing.

References

[1] A. Antos and G. Lugosi. Strong minimax lower bounds for learning. *Machine Learning*, 30:31–56, 1998.

[2] M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

[3] M.-F. Balcan and A. Blum. A PAC-style model for learning from labeled and unlabeled data. Book chapter in "Semi-Supervised Learning", O. Chapelle and B. Schlkopf and A. Zien, eds., MIT press, 2006.

[4] M.-F. Balcan, A. Broder, and T. Zhang. Margin based active learning. In *Proceedings of the 20th Annual Conference on Learning Theory*, 2007.

[5] R. Castro and R. Nowak. Minimax bounds for active learning. In *Proceedings of the 20th Annual Conference on Learning Theory*, 2007.

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

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

[8] S. Dasgupta. Coarse sample complexity bounds for active learning. In *Advances in Neural Information Processing Systems*, 2005.

[9] S. Dasgupta, D. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems*, 2007.

[10] S. Dasgupta, A. Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. In *Proceedings of the 18th Annual Conference on Learning Theory*, 2005.

[11] L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1996.

[12] S. Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.

[13] S. Hanneke. Teaching dimension and the complexity of active learning. In *Proceedings of the 20th Conference on Learning Theory*, 2007.

[14] D. Haussler, N. Littlestone, and M. Warmuth. Predicting $\{0, 1\}$-functions on randomly drawn points. *Information and Computation*, 115:248–292, 1994.

[15] J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.

[16] V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons Inc., 1998.

Appendix

A Proof of Theorem 8

First note that the total number of label requests used by the aggregation procedure in Algorithm 1 is at most t. Initially running the algorithms $A_1, \ldots, A_k$ requires $\sum_{i=1}^{k} \lfloor t/(4i^2) \rfloor \leq t/2$ labels, and the second phase of the algorithm requires $k^2 \lceil 72 \ln(4k/\delta) \rceil$ labels, which by definition of k is also less than $t/2$. Thus this procedure is a valid learning algorithm.

Now suppose that $h^* \in C_i$, and assume that

$$t \geq \max \left\{ 4i^2 \left\lceil S_i(\epsilon/2, \delta/2, h^*) \right\rceil, 2i^2 \left\lceil 72 \ln(4i/\delta) \right\rceil \right\} .$$

We must show that for any such value of t, $er(\hat{h}_t) \leq \epsilon$ with probability at least $1 - \delta$.

First notice that since $t \geq 2i^2 \lceil 72 \ln(4i/\delta) \rceil$, $k \geq i$. Furthermore, since $t/(4i^2) \geq \lceil S_i(\epsilon/2, \delta/2, h^*) \rceil$, with probability at least $1 - \delta/2$, running $\mathcal{A}_i(\lfloor t/(4i^2) \rfloor, \delta/2)$ returns a function h_i with $er(h_i) \leq \epsilon/2$.

Let $j^* = \operatorname{argmin}_j er(h_j)$. By Hoeffding's inequality, with probability at least $1 - \delta/4$, for all ℓ,

$$m_{j^*\ell} \leq \frac{7}{12} \left\lceil 72 \ln\left(4k/\delta\right) \right\rceil ,$$

and thus

$$\min_j \max_\ell m_{j\ell} \leq \frac{7}{12} \left\lceil 72 \ln(4k/\delta) \right\rceil .$$

Furthermore, by Hoeffding's inequality and a union bound, with probability at least $1 - \delta/4$, for any ℓ such that

$$m_{\ell j^*} \leq \frac{7}{12} \left\lceil 72 \ln(4k/\delta) \right\rceil$$

we have that

$$er(h_\ell | h_\ell(x) \neq h_{j^*}(x)) \leq \frac{2}{3}$$

and thus $er(h_\ell) \leq 2er(h_{j^*})$. By a union bound over these three events, we find that, as desired, with probability at least $1 - \delta$,

$$er(\hat{h}_t) \leq 2er(h_{j^*}) \leq 2er(h_i) \leq \epsilon .$$

■

B Proof of Theorem 9

Assume that (C, D) is learnable at an exponential rate. That means there exists an algorithm A such that for any target h^* in C, there exist constants $\gamma_{h^*} = \gamma(h^*, D)$ and k_{h^*} such that for any ϵ and δ, with probability at least $1 - \delta$, for any $t \geq \gamma_{h^*} (\log(1/(\epsilon\delta)))^{k_{h^*}}$, after t label requests, $A(t, \delta)$ outputs an ϵ-good classifier.

We define $C_i = \{h \in C : \gamma_h \leq i, k_h \leq i\}$. For every i, we define an algorithm A_i that achieves the required polylog verifiable sample complexity as follows. We first run A to obtain function h_A. We then let A_i always output the closest classifier in C_i to h_A. If $t \geq i(\log(2/(\epsilon\delta)))^i$, then after t label requests, with probability at least $1-\delta$, $A(t, \delta)$ outputs an $\epsilon/2$-good classifier, so by the triangle inequality, with probability at least $1-\delta$, $A_i(t, \delta)$ outputs an ϵ-good classifier. Furthermore, A_i can output $\hat{\epsilon}_t = (2/\delta) \exp\left\{-(t/i)^{1/i}\right\}$, which is no more than ϵ. Combining this with Theorem 8 we get the desired result. ■

C Heuristic Approaches to Decomposition

As mentioned, decomposing purely based on verifiable complexity with respect to (C, D) typically cannot yield a good decomposition even for very simple problems, such as unions of intervals. The reason is that the set of classifiers with high verifiable sample complexity may itself have high verifiable complexity.

Although we do not yet have a general method that can provably always find a good decomposition when one exists (other than the trivial method in the proof of Theorem 9), we often find that a heuristic recursive technique can be quite effective. That is, we can define $C_1 = C$. Then for $i > 1$, we recursively define C_i as the set of all $h \in C_{i-1}$ such that $\theta_h = \infty$ with respect to (C_{i-1}, D). Suppose that for some N, $C_{N+1} = \emptyset$. Then for the decomposition $C_1, C_2, \ldots, C_N$, every $h \in C$ has $\theta_h < \infty$ with respect to at least one of the sets in which it is contained. Thus, the verifiable sample complexity of h with respect to that set is $O(\mathrm{polylog}(1/\epsilon\delta))$, and the aggregation algorithm can be used to achieve polylog sample complexity.

We could alternatively perform a similar decomposition using a suitable definition of splitting index [8], or more generally using

$$\limsup_{\epsilon \to 0} \frac{S_{C_{i-1}}(\epsilon, \delta, h)}{\left(\log\left(\frac{1}{\epsilon\delta}\right)\right)^k}$$

for some fixed constant $k > 0$.

While this procedure does not always generate a good decomposition, certainly if $N < \infty$ exists, then this creates a decomposition for which the aggregation algorithm, combined with an appropriate sequence of algorithms $\{A_i\}$, can achieve exponential rates. In particular, this is the case for all of the (C, D) described in Section 5. In fact, even if $N = \infty$, as long as every $h \in C$ does end up in *some* set C_i for finite i, this decomposition would still provide exponential rates.

D Proof of Theorem 6

We now finally prove Theorem 6. This section is mostly self-contained, though we do make use of Theorem 8 from Section 4 in the final step of the proof.

For any $V \subseteq C$ and $h \in C$, define

$$\bar{B}_V(h, r) = \{h' \in \bar{V} : \mathbb{P}_D(h(x) \neq h'(x)) \leq r\},$$

where $\bar{V}$ is, as before, a countable dense subset of V. Define the *boundary* of h with respect to D and V, denoted $\partial_V h$, as

$$\partial_V h = \lim_{r \to 0} \mathrm{DIS}(\bar{B}_V(h, r)).$$

The proof will proceed according to the following outline. We begin in Lemma 13 by describing special conditions under which a CAL-like algorithm has the property that the more unlabeled examples it processes, the smaller the fraction of them it requests the labels of. Since CAL always identifies the target's true label on any example it processes, we end up with a set of labeled examples growing strictly faster than the number of label requests used to obtain it; we can use this as a training set in any passive learning algorithm. However, the special conditions under which this happens are rather limiting, so we require an additional step, in Lemma 14; there, we exploit a subtle relation between overlapping boundary regions and shatterable sets to show that we can decompose any finite VC dimension class into a countable number of subsets satisfying these special conditions. This, combined with the aggregation algorithm, extends Lemma 13 to the general conditions of Theorem 6.

Lemma 13 *Suppose (C, D) is such that C has finite VC dimension d, and $\forall h \in C, \mathbb{P}(\partial_C h) = 0$. Then for any passive learning sample complexity $S_p(\epsilon, \delta, h)$ for (C, D), there exists an active learning algorithm achieving a sample complexity $S_a(\epsilon, \delta, h)$ such that, for any target function $h^* \in C$ where $S_p(\epsilon, \delta, h^*) = \omega(1)$,*

$$S_a(\epsilon, \delta/2, h^*) = o(S_p(\epsilon/2, \delta, h^*)).$$

Proof: We perform the learning in two phases. The first is a passive phase: we simply request the labels of $x_1, x_2, \ldots, x_{\lfloor t/3 \rfloor}$, and let

$$V = \{h \in \bar{C} : \forall i \leq \lfloor t/3 \rfloor, h(x_i) = h^*(x_i)\}.$$

In other words, V is the set of all hypotheses that correctly label the first $\lfloor t/3 \rfloor$ examples. By standard consistency results [11], with probability at least $1 - \delta/8$, there is a universal constant $c > 0$ such that

$$\sup_{h_1, h_2 \in V} \mathbb{P}_D(h_1(x) \neq h_2(x)) \leq c\left(\frac{d \ln t + \ln \frac{1}{\delta}}{t}\right).$$

In particular, on this event, we have

$$\mathbb{P}(\mathrm{DIS}(V)) \leq \mathbb{P}\left(\mathrm{DIS}\left(\bar{B}\left(h^*, c\frac{d \ln t + \ln \frac{1}{\delta}}{t}\right)\right)\right).$$

Let us denote this latter quantity by Δ_t. Note that Δ_t goes to 0 as t grows.

If ever we have $\mathbb{P}(\mathrm{DIS}(V)) = 0$ for some finite t, then clearly we can return any $h \in V$, so this case is easy.

Otherwise, let $n_t = \lfloor t/(36\mathbb{P}(\mathrm{DIS}(V)) \ln(8/\delta)) \rfloor$, and suppose $t \geq 3$. By a Chernoff bound, with probability at least $1 - \delta/8$, in the sequence of examples $x_{\lfloor t/3 \rfloor+1}, x_{\lfloor t/3 \rfloor+2}, \ldots, x_{\lfloor t/3 \rfloor+n_t}$, at most $t/3$ of the examples are in $\mathrm{DIS}(V)$. If this is not the case, we fail and output an arbitrary h; otherwise, we request the labels of every one of these n_t examples that are in $\mathrm{DIS}(V)$. Now construct a sequence $\mathcal{L} = \{(x'_1, y'_1), (x'_2, y'_2), \ldots, (x'_{n_t}, y'_{n_t})\}$ of labeled examples such that $x'_i = x_{\lfloor t/3 \rfloor+i}$, and y'_i is either the label agreed upon by all the elements of V, or it is the $h^*(x_{\lfloor t/3 \rfloor+i})$ label value we explicitly requested. Note that because $\inf_{h \in V} er(h) = 0$ with probability 1, we also have that with probability 1 every $y'_i = h^*(x'_i)$. We may therefore use these n_t examples as iid training examples for the passive learning algorithm.

Specifically, let us split up the sequence $\mathcal{L}$ into $k = 4$ sequences $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k$, where

$$\begin{aligned}\mathcal{L}_i \quad = \quad & \{(x'_{(i-1)\lfloor n_t/k \rfloor+1}, y'_{(i-1)\lfloor n_t/k \rfloor+1}), \\ & (x'_{(i-1)\lfloor n_t/k \rfloor+2}, y'_{(i-1)\lfloor n_t/k \rfloor+2}), \\ & \ldots, (x'_{i\lfloor n_t/k \rfloor}, y'_{i\lfloor n_t/k \rfloor})\}.\end{aligned}$$

Suppose A is the passive learning algorithm that guarantees $S_p(\epsilon, \delta, h)$ passive sample complexities. Then for $i \in \{1, 2, \ldots, k-1\}$, let h_i be the classifier returned by $A(\mathcal{L}_i, \delta)$.

Additionally, let h_k be any classifier in V consistent with the labels in $\mathcal{L}_k$.

Finally, for each $i, j \in \{1, 2, \ldots, k\}$, request the labels of the first $\lfloor t/(3k^2) \rfloor$ examples in the sequence $\{x_{\lfloor t/3 \rfloor + n_t + 1}, x_{\lfloor t/3 \rfloor + n_t + 2}, \ldots\}$ that satisfy $h_i(x) \neq h_j(x)$ and let R_{ij} denote these $\lfloor t/(3k^2) \rfloor$ labeled examples ($R_{ij} = \emptyset$ if $\mathbb{P}_D(h_i(x) \neq h_j(x)) = 0$). Let m_{ij} denote the number of mistakes h_i makes on the set R_{ij}. Finally, let $\hat{h}_t = h_i$ where

$$i = \operatorname*{argmin}_i \max_j m_{ij}.$$

This will be the classifier we return.

It is known (see, e.g., [11]) that if $\lfloor n_t/k \rfloor \geq c'((d/\epsilon)\log(1/\epsilon) + (1/\epsilon)\log(1/\delta))$ for some finite universal constant c', then with probability at least $1 - \delta/8$ over the draw of $\mathcal{L}_k$, $er(h_k) \leq \epsilon$. Define

$$\bar{S}_p(\epsilon, \delta, h^*) = \min\left\{ S_p(\epsilon, \delta, h^*), c' \frac{d\log(1/\epsilon) + \log(1/\delta)}{\epsilon} \right\}.$$

We have chosen k large enough so that, if $\lfloor n_t/k \rfloor \geq \bar{S}_p(\epsilon, \delta, h^*)$, then with probability at least $1 - \delta/8$ over the draw of $\mathcal{L}$, $\min_i er(h_i) \leq \epsilon$. Furthermore, by a Hoeffding bound argument (similar to the proof of Theorem 8), for any $t \geq t_0 = 3k^2 \lceil 72 \ln(16k/\delta) \rceil$, we have that with probability at least $1 - \delta/8$, $er(\hat{h}_t) \leq 2 \min_i er(h_i)$. Define

$$S_a(2\epsilon, \delta/2, h^*) = \\ 1 + \inf\left\{ s \geq t_0 : s \geq 144k \ln \frac{8}{\delta} \bar{S}_p(\epsilon, \delta, h^*) \Delta_s \right\}.$$

Note that if $t \geq S_a(2\epsilon, \delta/2, h^*)$, then (with probability $\geq 1 - \delta/8$)

$$\bar{S}_p(\epsilon, \delta, h^*) \leq \frac{t}{144k \ln \frac{8}{\delta} \Delta_t} \leq \lfloor n_t/k \rfloor .$$

So, by a union bound over the possible failure events listed above ($\delta/8$ for $\mathbb{P}(\mathrm{DIS}(V)) > \Delta_t$, $\delta/8$ for more than $t/3$ examples of $\mathcal{L}$ in $\mathrm{DIS}(V)$, $\delta/8$ for $\min_i er(h_i) > \epsilon$, and $\delta/8$ for $er(\hat{h}_t) > 2 \min_i er(h_i)$), if $t \geq S_a(2\epsilon, \delta/2, h^*)$, then with probability at least $1 - \delta/2$, $er(\hat{h}_t) \leq 2\epsilon$. So $S_a(\epsilon, \delta, h^*)$ is a valid sample complexity function, achieved by the described algorithm. Furthermore,

$$S_a(\epsilon, \delta/2, h^*) \leq 1 \\ + \max\left\{ t_0, 144k \ln \frac{8}{\delta} \bar{S}_p(\epsilon/2, \delta, h^*) \Delta_{S_a(\epsilon, \delta/2, h^*) - 2} \right\}.$$

$S_p(\epsilon, \delta, h^*) = \omega(1)$ implies $S_a(\epsilon, \delta/2, h^*) = \omega(1)$, so we know that $\Delta_{S_a(\epsilon, \delta/2, h^*) - 2} = o(1)$. Thus, $S_a(\epsilon, \delta/2, h^*) = o\left(\bar{S}_p(\epsilon/2, \delta, h^*)\right)$, and thus we have $S_a(\epsilon, \delta/2, h^*) = o(S_p(\epsilon/2, \delta, h^*))$. ■

As an interesting aside, it is also true (by essentially the same argument) that under the conditions of Lemma 13, the *verifiable* sample complexity of active learning is strictly smaller than the *verifiable* sample complexity of passive learning in this same sense. In particular, this implies a verifiable sample complexity that is $o(1/\epsilon)$ under these conditions. For instance, with some effort one can show that these conditions are satisfied when the VC dimension of C is 1, or when the support of D is at most countably infinite. However, for more complex learning problems, this condition will typically not be satisfied, and as such we require some additional work in order to use this lemma toward a proof of the general result in Theorem 6. Toward this end, we again turn to the idea of a decomposition of C, this time decomposing it into subsets satisfying the condition in Lemma 13.

Lemma 14 *For any (C, D) where C has finite VC dimension d, there exists a countably infinite sequence $C_1, C_2, \ldots$ such that $C = \cup_{i=1}^{\infty} C_i$ and $\forall i, \forall h \in C_i, \mathbb{P}(\partial_{C_i} h) = 0$.*

Proof: The case of $d = 0$ is clear, so assume $d > 0$. A decomposition procedure is given in Algorithm 2. We will show that, if we let $\mathbb{H} = \mathrm{Decompose}(C)$, then the maximum recursion depth is at most d (counting the initial call as depth 0). Note that if this is true, then the lemma is proved, since it implies that $\mathbb{H}$ can be uniquely indexed by a d-tuple of integers, of which there are at most countably many.

Algorithm 2 Decompose($\mathcal{H}$)

```
Let H_∞ = {h ∈ H : P(∂_H h) = 0}
if H_∞ = H then
  Return {H}
else
  For i ∈ {1, 2, ...}, let H_i =
    {h ∈ H : P(∂_H h) ∈ ((1+2^{-(d+3)})^{-i}, (1+2^{-(d+3)})^{1-i}]}
  Return ⋃_{i∈{1,2,...}} Decompose(H_i) ∪ {H_∞}
end if
```

For the sake of contradiction, suppose that the maximum recursion depth of Decompose(C) is more than d (or is infinite). Thus, based on the first $d+1$ recursive calls in one of those deepest paths in the recursion tree, there is a sequence of sets

$$C = \mathcal{H}^{(0)} \supseteq \mathcal{H}^{(1)} \supseteq \mathcal{H}^{(2)} \supseteq \cdots \mathcal{H}^{(d+1)} \neq \emptyset$$

and a corresponding sequence of finite positive integers $i_1, i_2, \ldots, i_{d+1}$ such that for each $j \in \{1, 2, \ldots, d+1\}$, every $h \in \mathcal{H}^{(j)}$ has

$$\mathbb{P}(\partial_{\mathcal{H}^{(j-1)}} h) \in \left((1 + 2^{-(d+3)})^{-i_j}, (1 + 2^{-(d+3)})^{1-i_j} \right].$$

Take any $h_{d+1} \in \mathcal{H}^{(d+1)}$. There must exist some $r > 0$ such that $\forall j \in \{1, 2, \ldots, d+1\}$,

$$\mathbb{P}(\mathrm{DIS}(\bar{B}_{\mathcal{H}^{(j-1)}}(h_{d+1}, r))) \\ \in \left((1 + 2^{-(d+3)})^{-i_j}, (1 + 2^{-(d+2)})(1 + 2^{-(d+3)})^{-i_j} \right].$$

In particular, any set of $\leq 2^{d+1}$ classifiers $T \subset \bar{B}_{\mathcal{H}^{(j)}}(h_{d+1}, r/2)$ must have $\mathbb{P}(\cap_{h \in T} \partial_{\mathcal{H}^{(j-1)}} h) > 0$.

We now construct a shattered set of points of size $d+1$. Consider constructing a binary tree with 2^{d+1} leaves as follows. The root node contains h_{d+1} (call this level 0). Let $h_d \in \bar{B}_{\mathcal{H}^{(d)}}(h_{d+1}, r/4)$ be some classifier with $\mathbb{P}(h_d(X) \neq h_{d+1}(X)) > 0$. Let the left child of the root be h_{d+1} and the right child be h_d (call this level 1). Define $A_1 = \{x :$

$h_d(x) \neq h_{d+1}(x)\}$, and let $\Delta_1 = 2^{-(d+2)}\mathbb{P}(A_1)$, Now for each $j \in \{d-1, d-2, \ldots, 0\}$ in decreasing order, we define the $d-j+1$ level of the tree as follows. Let T_{j+1} denote the nodes at the $d-j$ level in the tree, and let $A'_{d-j+1} = \bigcap_{h \in T_{j+1}} \partial_{\mathcal{H}^{(j)}} h$. We iterate over the elements of T_{j+1} in left-to-right order, and for each one h, we find $h' \in \bar{B}_{\mathcal{H}^{(j)}}(h, \Delta_{d-j})$ with

$$\mathbb{P}_D(h(x) \neq h'(x) \wedge x \in A'_{d-j+1}) > 0 .$$

We then define the left child of h to be h and the right child to be h', and we update

$$A'_{d-j+1} \leftarrow A'_{d-j+1} \cap \{x : h(x) \neq h'(x)\} .$$

After iterating through all the elements of T_{j+1} in this manner, define A_{d-j+1} to be the final value of A'_{d-j+1} and $\Delta_{d-j+1} = 2^{-(d+2)}\mathbb{P}(A_{d-j+1})$. The key is that, because every h in the tree is within $r/2$ of h_{d+1}, the set A'_{d-j+1} always has nonzero measure, and is contained in $\partial_{\mathcal{H}^{(j)}} h$ for any $h \in T_{j+1}$, so there always exists an h' arbitrarily close to h with $\mathbb{P}_D(h(x) \neq h'(x) \wedge x \in A'_{d-j+1}) > 0$.

Note that for $i \in \{1, 2, \ldots, d+1\}$, every node in the left subtree of any h at level $i-1$ is strictly within distance $2\Delta_i$ of h, and every node in the right subtree of any h at level $i-1$ is strictly within distance $2\Delta_i$ of the right child of h. Since $2\Delta_i 2^{d+1} = \mathbb{P}(A_i)$, there must be some set $A_i^* \subseteq A_i$ with $\mathbb{P}(A_i^*) > 0$ such that for every h at level $i-1$, every node in its left subtree agrees with h on every $x \in A_i^*$ and every node in its right subtree disagrees with h on every $x \in A_i^*$. Therefore, taking any $\{x_1, x_2, \ldots, x_d, x_{d+1}\}$ such that each $x_i \in A_i^*$ creates a shatterable set (shattered by the set of leaf nodes in the tree). This contradicts VC dimension d, so we must have that the maximum recursion depth is at most d. ■

Proof:[Theorem 6] Theorem 6 now follows by a simple combination of Lemmas 13 and 14, along with Theorem 8. That is, the passive learning algorithm achieving passive learning sample complexity $S_p(\epsilon, \delta, h)$ on (C, D) also achieves $S_p(\epsilon, \delta, h)$ on any (C_i, D), where $C_1, C_2, \ldots$ is the decomposition from Lemma 14. So Lemma 13 guarantees the existence of active learning algorithms $A_1, A_2, \ldots$ such that A_i achieves a sample complexity $S_i(\epsilon, \delta/2, h) = o(S_p(\epsilon/2, \delta, h))$ on (C_i, D) for all $h \in C_i$ s.t. $S_p(\epsilon, \delta, h) = \omega(1)$. Finally, Theorem 8 tells us that this implies the existence of an active learning algorithm based on these A_i combined with Algorithm 1, achieving sample complexity $o(S_p(\epsilon/4, \delta, h))$ on (C, D). ■

Note there is nothing special about 4 in Theorem 6. Using a similar argument, it can be made arbitrarily close to 1.