Asymptotic Active Learning

Maria-Florina Balcan* Eyal Even-Dar† Steve Hanneke‡ Michael Kearns§
Yishay Mansour¶ Jennifer Wortman§

Abstract

We describe and analyze a PAC-asymptotic model for active learning. We show that in many cases where it has traditionally been believed that active learning does not help, active learning does help *asymptotically*. This view contrasts sharply with the traditional $1/\epsilon$ lower bounds for active learning classes such as non-homogeneous linear separators under the uniform distribution or unions of k intervals under an arbitrary distribution, both of which are learnable at an exponential rate in our model.

1 Introduction

Traditionally, machine learning has focused on the problem of learning a task from labeled examples only. However, for many contemporary practical problems such as classifying web pages or detecting spam, there is often additional information available. In particular, for many of these settings unlabeled data is often much cheaper and more plentiful than labeled data. As a consequence, there has recently been substantial interest in using unlabeled data together with labeled data for learning [5, 16]. Clearly, if useful information can be extracted from the unlabeled data that reduces the dependence on labeled examples, this can be a significant benefit [17, 4].

A setting for incorporating unlabeled data in the learning process that has been increasingly popular setting in the past few years is *active learning*. In this model, the learning algorithm has both the capability of drawing random unlabeled examples from the underlying distribution, and that of asking for the labels of *any* of these examples. The hope is that a good classifier can be learned with significantly fewer labels by actively directing the queries to informative examples. A number of active-learning analyses have recently been proposed in a PAC-style setting, both for the realizable and for the agnostic cases, and a sequence of important positive and negative results has been obtained [7, 8, 9, 1, 11, 3, 10, 14]. 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 [9, 1, 11, 3, 10]. Unfortunately, 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 active learning does not help at all in the traditional active learning model, even in the realizable case [9].

*Computer Science Department, Carnegie Mellon University, ninamf@cs.cmu.edu
†Google Research, evendar.eyal@gmail.com
‡Machine Learning Department, Carnegie Mellon University, shanneke@cs.cmu.edu
§Department of Computer and Information Science, University of Pennsylvania, {mkearns,wortmanj}@cis.upenn.edu
¶School of Computer Science, Tel Aviv University, and Google Research, mansour@tau.ac.il

In this work we take a different angle on the active learning problem and analyze the asymptotic complexity of active learning. We show that in many interesting cases where it has been thought that active learning does not help, active learning *does help asymptotically*. This contrasts with the usual view of active learning [1, 14, 15, 9, 8, 13], and it is closer in spirit with work done in statistics [12]. The main point we try to make in this paper is that, with a small modification to the traditional PAC-style model, we can show that significant improvements in label complexity are often achievable. Specifically:

1. In this model it is possible to actively learn with an *exponential rate* pairs of concept classes and distributions that are known to require a linear rate in the traditional PAC-style active learning setting: for example, intervals on $[0, 1]$ and non-homogeneous linear separators under the uniform distribution. The exponential rates involve a constant that is dependent on the target function. It is also possible to learn more complicated geometric concept spaces and under more general distributions than in the traditional PAC-style model.

2. Some of our learning procedures are implemented by constructing a hierarchy of nested concept classes, which is carefully chosen to get nice rates, and combining the classes using an aggregation algorithm. We offer a generic procedure for building good hierarchies for many classes in a distribution-dependent way.

3. We show that even in this new model, there do exist lower bounds; it is possible to exhibit somewhat contrived distributions where exponential rates are not possible even for non-complicated concept spaces (see Theorem 4.5). It is currently an open question whether there exist $\Omega(1/\epsilon)$ lower bounds in this model, or if it is the case that *any* concept class and distribution can be learned with active sample complexity $o(1/\epsilon)$.

It is important to note that in our model we bound the number of queries the algorithm makes before it finds a good function (i.e. one of arbitrarily small error rate), but not the number of queries before it can *prove* or it *knows* it has found a good function. This allows us to obtain significantly better bounds on the number of label queries required to learn. To our knowledge, this is the first work to address this subtle point in the context of active learning.

In previous work on asymptotic convergence rates for Active Learning [10, 6] the analysis is done in the same model as ours, however the results proven were *no stronger* than the results one could prove in the traditional PAC model [1, 14, 15, 9, 8, 13].

2 The Traditional PAC Active Learning Model

In the traditional model of active learning, a learning algorithm is given access to a large set of unlabeled examples drawn from some underlying distribution D, and may repeatedly query for the label $h^*(x)$ of any example x in the set. The goal of the learning algorithm is to, after making a (hopefully small) number of label requests, halt and output a classifier that, with probability $1 - \delta$ has error rate at most ϵ with respect to D. The number of label requests necessary to learn a target in a given concept class is referred to as the *label complexity*. It is clear from standard results from the literature on supervised learning [20] that it is trivial to achieve a label complexity which is linear in $1/\epsilon$ and VC dimension, and logarithmic in $1/\delta$. As such, there has been much interest in determining when it is the case that active learning can result in a label complexity that is only logarithmic in $1/\epsilon$.

Unfortunately, while there are a small number of cases where exponential improvements have been shown (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 in the traditional active learning model. For example, consider the class of intervals in $[0,1]$. In order to distinguish *with certainty* the all-negative hypothesis from the set of hypotheses that are positive on a region of weight ϵ, $\Omega(1/\epsilon)$ labeled examples are needed. It is interesting to note that for this concept class and many others, the number of labels needed to learn a good approximation of the target is heavily dependent on the target itself. Indeed, in the intervals setting, a hypothesis that is positive on a region of weight at least $1/2$ can be learned with accuracy $1-\epsilon$ with only a logarithmic dependency on ϵ by sampling to find a positive point and then running binary search from this point to determine the end points of the interval.

Recently, there have been a few quantities proposed to measure the effectiveness of active learning on particular concept classes and distributions [9, 15, 14]. One is Dasgupta's *splitting index* [9], which is dependent on the concept class, data distribution, target function, and a parameter τ, quantifies how easy it is to reduce the diameter of the version space by choosing an example to query. Here we describe an alternate quantity, Hanneke's *disagreement coefficient* [14].

Definition 1 *For any $h \in C$ and $r > 0$, let $B(h,r)$ be a ball of radius r around h in C. That is,*

$$B(h,r) = \{h' \in C : Pr_{x\sim D}[h(x) \neq h'(x)] \leq r\} .$$

Define the disagreement rate at radius r *as*

$$\Delta_r^{(h)} = Pr_{x\sim D}[\exists h_1, h_2 \in B(h,r) : h_1(x) \neq h_2(x)] .$$

The disagreement coefficient of a hypothesis h*, denoted θ_h is the infimum value of $\Theta > 0$ such that for all $r > 0$, $\Delta_r^{(h)} \leq \Theta r$. The disagreement coefficient for a* concept space C *with respect to a distribution D is defined as $\theta = \sup_{h\in C} \theta_h$.*

The disagreement coefficient has previously been a useful quantity for analyzing the label complexity of active learning algorithms. For example, it has been shown that the realizable version of the A^2 active learning algorithm[1] (which is essentially the active learning algorithm of Cohn, Atlas, and Ladner [7]) achieves error at most ϵ with probability $1-\delta$ using a number of label requests at most $\theta d \cdot \mathrm{polylog}(1/\epsilon, 1/\delta)$, where d is the VC dimension of C [14]. We will see that both the disagreement coefficient and splitting index are also useful quantities for analyzing performance in the PAC-asymptotic model.

3 The PAC-Asymptotic Model

Let X be an instance space and $Y = \{-1, 1\}$ be the set of possible labels. Let C be the hypothesis class, a set of functions mapping from X to Y, and assume that C has finite VC dimension d. We consider here the realizable setting [19] in which it is assumed that there is a distribution D over instances in X, and 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 X is defined as $\mathrm{err}_D(h) = \Pr_{x\sim D}[h(x) \neq h^*(x)]$. The goal is to find a hypothesis $h \in H$ with small error with respect D, while simultaneously minimizing the number of label requests that the learning algorithm makes.

We define the *active sample complexity* of a class C and distribution D as a function of the accuracy parameter ϵ, confidence parameter δ, and target function $h^* \in C$ as follows.

Definition 2 *We say the pair (C, D) has active sample complexity at most $S = S(\epsilon, \delta, h^*)$ if there exists an active learning algorithm such that for all target functions h^* in C, for any ϵ and any δ, the algorithm* *achieves error less than ϵ with respect to D with probability at least $1-\delta$, using a number of label requests at most $S(\epsilon, \delta, h^*)$.*

The crucial distinction between this definition and the definition of label complexity in the traditional PAC-style analysis described in Section 2 is that here we are only concerned with the number of label requests the algorithm needs to make before it finds an ϵ-good classifier, rather than the number of label requests before it *knows* it has found an ϵ-good classifier. This distinction is subtle, but will prove to be important in the analysis of many common concept classes.

Given standard results in the supervised passive learning setting [20] it is trivial to achieve an active sample complexity S which is linear in $1/\epsilon$ and VC dimension and logarithmic in $1/\delta$. We will show that many common pairs (C, D) have sample complexity that is polylogarithmic in *both* $1/\epsilon$ and $1/\delta$ and linear only in γ_{h^*}, where γ_{h^*} is a finite target-dependent constant. This contrasts sharply with the infamous $1/\epsilon$ lower bounds of the traditional PAC-style model [15, 9, 8, 13]. The implication is that, for any fixed target h^*, such lower bounds often vanish as ϵ approaches 0. This also contrasts with passive learning, where $1/\epsilon$ lower bounds are typically unavoidable.

Definition 3 *We say that (C, D) is actively learnable at an exponential rate if there exists an algorithm $\mathcal{A}$ such that for any target h^* in C, there exists a $\gamma_{h^*} = \gamma(h^*, D)$ such that for any ϵ and δ, with probability $\geq 1-\delta$, $\mathcal{A}$ finds a hypothesis with error $\leq \epsilon$ after making at most $\gamma_{h^*} \cdot \mathrm{polylog}(1/\epsilon, 1/\delta)$ label queries.*

We can similarly define a notion of active learnability at a *sublinear* rate. We note again that γ_{h^*} is allowed to depend on the *target*, but may not depend on ϵ.

To get some intuition about when this model can be useful, consider the following example. Suppose once again that C is the class of all intervals over $[0, 1]$ and D is any distribution over $[0, 1]$. We can actively learn at an exponential rate using an extremely simple learning algorithm as follows. At each point in time, choose a point x uniformly at random from the unlabeled sample and query its label. If x is negative, output the all-negative function as the current "best guess" of the target and continue running the algorithm. If a point x is eventually found that has a positive label, alternate between running one binary search on the examples between 0 and x and a second on the examples between x and 1 until the end points of the interval are found, outputting any consistent interval at each time step.

If the target is the all-negative function, the algorithm described above will output the all-negative function at every moment in time. If the target is an interval $[a, b]$, where $b - a = w$, then after roughly $1/w$ queries (a constant number that depends only on the target), a positive example will be found. Since only $\log(1/\epsilon)$ queries are required to run the binary search to the accuracy ϵ, the desired active sample complexity is achieved.

It is important to note that in this setting, the learning algorithm is run indefinitely with the requirement that a current guess of the target hypothesis is output after each request for a label. While we bound the number of queries the algorithm makes before it finds and outputs a good function (one of arbitrarily small error rate), no bound is given on the number of queries before it can *prove* or *know* it has found a good function.[1] For example, in the algorithm for learning intervals described above, it is impossible to know that the correct target is the all-negative function, but if this is the case, then the all-negative function will be output immediately and after every subsequent query. This raises the question of whether the $\Omega(1/\epsilon)$ lower bounds for learning many simple classes in the traditional active learning model are often simply an artifact of definitions.

[1]This is not really a problem since many practical algorithms do not use ϵ as a parameter.

3.1 A Hierarchical View of the Model

Another convenient, though possibly less general, way to think about this asymptotic model of active learning is the following. Given the distribution D and the function class C, we partition C (possibly in a distribution-dependent way) into a countably infinite sequence of subclasses, effectively constructing a hierarchy $C_0 \subset C_1 \subset \ldots$. We then show that we can actively learn every subclass C_i with only $O\left(\text{polylog}(1/\epsilon, 1/\delta)\right)$ queries, where the constant hidden in the O depends on C_i. It is straightforward to achieve the guarantee in Definition 3 if we know the complexity of the target (i.e. the smallest index i such that the target is in C_i). However, it is also possible to achieve the guarantee when we do not know the complexity of the target by using an aggregation procedure without incurring much cost in the label complexity. There are multiple ways to do the aggregation; we describe a simple method below in which multiple algorithms are run on different subclasses C_i in parallel and combined using a meta-procedure. Within each subclass we can run a standard active learning algorithm such as the A^2 algorithm [1] or Dasgupta's splitting algorithm [9].

Assume that we can learn each C_i with only $S(\epsilon, \delta, i)$ queries by using an active learning algorithm A_i. This implies that for each subclass C_i and algorithm A_i, we can compute an upper bound $B_i(q_i, \delta_i)$ on the error of the function h_i currently output by A_i based on the number of queries q_i that algorithm A_i has made. This bound will hold with probability $1 - \delta_i$ as long as the target function is in C_i. Using these bounds, we can build a meta-algorithm for aggregation as follows.

For each query $t = 1, 2, \ldots$, the meta-algorithm picks the algorithm A_i with $i \in \{1, \ldots, \lfloor \log_2(2t) \rfloor\}$ that has made the fewest queries so far and allows it to query the label of any point. The algorithm A_i receives this label, updates its current hypothesis h_i accordingly, and increments q_i. The meta-algorithm then outputs as its current hypothesis the classifier h_j output by the algorithm A_j with smallest index j that satisfies the the property that for all $k > j$, the distance between h_j and h_k can be upper bounded by the sum of the current error bounds $B_j(q_j, \delta_j)$ and $B_k(q_k, \delta_k)$ for the classes C_j and C_k respectively.

Using this meta-algorithm, we can then show that the sample complexity of C is at most $O(S(\epsilon, \delta, i) \log S(\epsilon, \delta, i))$, and we only incur an additional $\log S(\epsilon, \delta, i)$ factor in the sample complexity for not knowing the complexity of the target. For example if the label complexity for set C_i is $\gamma_i \cdot \log\left(1/\epsilon\right)$, and the target is in C_i, then the error bound our meta-procedure is $3 \cdot 2^{-t/(\log(2t)\gamma_i)}$, where t is the number of label requests. In other words, the number of label requests needed before the algorithm's hypothesis has error no worse than ϵ is at most $O(\gamma_i \cdot \log\left(1/\epsilon\right) \cdot \log\left(\gamma_i \cdot \log\left(1/\epsilon\right)\right))$.

Since it is a bit more abstract and it allows us to use known active learning algorithms as a black box we will mostly use this alternate, hierarchical view of Definition 3 throughout the remainder of the paper. As we will see, however, not every pair (C, D) is actively learnable at an exponential rate, even when C has low complexity (e.g. VC dimension 1) when the distribution is not very nice. In consequence, it makes sense to consider and explore weaker sublinear active learning rates as well. In this paper we will focus on exponential rates though.

4 Exponential Rates

In this section, we describe a number of concept classes and distributions that are learnable at an exponential rate in the PAC-asymptotic model, many of which require $\Omega(1/\epsilon)$ labels in the traditional PAC-style model of active learning. These results illustrate the subtle power the PAC-asymptotic model gains by requiring only that the learning algorithm outputs an ϵ-good hypothesis without requiring that the algorithm *knows* that its current hypothesis is ϵ-good.

4.1 Exponential Rates for Simple Classes

A simple observation is that if the instance space X is finite, then any pair (C, D) is learnable under Definition 3 with $\gamma = |X|$; we simply query for the label of every instance $x \in X$ in the support of D and learn the target exactly. Another simple observation is that if C is countable, then (C, D) is learnable under Definition 3. Listing the elements of C as $h_1, h_2, \ldots$, we can simply let $S_i = \{h_1, h_2, \ldots, h_i\}$. Clearly S_i can be learned with $\leq i$ queries, so we can effectively apply the aggregation algorithm of the previous section. It is also clear that any pair (C, D) learnable with an exponential rate in the traditional PAC-style active learning setting is learnable in our setting as well. We present here a few other simple positive examples under Definition 3.

1. **Unions of k intervals under arbitrary distributions:** Let X be $[0, 1]$ and let C be the class of unions of at most k intervals, i.e. C contains functions of the form $\{a_0 = 0, a_1, ..., a_l = 1\}$, where $l \leq k$ and a_i, $i \in \{0, \ldots, l\}$ are the transition points between positive and negative segments. We can learn this class for any distribution D under Definition 3 by using a hierarchical structure defined as follows. C_0 contains the all negative function, and in general C_i contains all the functions with $\min_i Pr_{x\sim D}[a_i \leq x < a_{i+1})] \geq 2^{-i}$. We can then use the A^2 algorithm [1] or the splitting algorithm in [9] to learn within each C_i together with the aggregation meta-procedure in Section 3.1 to get the overall guarantee.

2. **Axis-Parallel Splits** Let X be the cube $[0, 1]^d$, D is uniform on X, and let C be the class of decision trees using a finite number of axis-parallel splits [12]. We can define a parameter λ for each concept as the smallest value ℓ such that a grid of cubes with length ℓ has the property that any cube contains either a single vertex or no vertices and intersects only a single hyperplane which is "reasonably balanced." In this case, we could active learn according to Definition 3 with γ exponential in λ, i.e., $(1/\lambda)^d$. In each sub-cube of size λ we will perform the learning separately. Since each cube is reasonably balanced we can achieve the desired exponential rate.

   We can alternatively stratify based on the splitting index or disagreement coefficient (which can be easily bounded based on the volume of the smallest region in the partition) and use the A^2 algorithm to learn within each class.

4.2 Geometric Concepts, Uniform Distribution

Many interesting geometric concepts in d dimensions are learnable under Definition 3 if the underlying distribution is uniform. Here we provide some examples. All of the results in this section also hold if the distribution is λ-close to uniform.

4.2.1 Linear Separators

Theorem 4.1 *Let C be the hypothesis class of linear separators in d dimensions, and let D be the uniform distribution over the surface of the unit sphere. The pair (C, D) is learnable under Definition 3.*

Proof Sketch : There are multiple ways to achieve this. We describe here a simple proof that uses a hierarchical decomposition as follows. Let $\lambda(h)$ by the probability mass of the minority class under hypothesis h. C_0 will contain only the all-negative and all-positive separators, and more generally, C_i will be the class of all separators h such that $\lambda(h) \geq 2^{-i}$ in addition to the all-positive and all-negative separators. We then use the A^2 algorithm [1] 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 is of the order $\sqrt{d}/\lambda(h)$. The results in [14] concerning the A^2 algorithm then imply the desired result. ∎

4.2.2 Unions of Convex Polytopes

Theorem 4.2 *Let C be the hypothesis class of union of k convex polytopes with nonintersecting decision boundaries, in d dimensions, completely contained within $(0,1)^d$. Let D be the uniform distribution over $[0,1]^d$. The pair (C,D) is learnable under Definition 3.*

The proof involves defining a hierarchy such that every level has disagreement coefficient bounded by a constant.

4.3 A Composition Theorem

The following composition result can be obtained by showing that it is possible to learn a concept class on a distribution D by filtering examples from D into two streams of data, running active learning algorithms in parallel on each stream, and outputing any hypothesis in the intersection of the algorithms' version spaces.

Theorem 4.3 *Let C be an arbitrary hypothesis class. Assume that the pairs (C,D_1) and (C,D_2) are learnable under Definition 3. Then for any $\alpha \in [0,1]$ the pair $(C, \alpha D_1 + (1-\alpha)D_2)$ is learnable under Definition 3.*

Note that this result significantly extends the variety of distributions for which we can prove learnability with exponential rates. In particular, we can use this result to prove that linear separators (and generally, convex polytopes) are learnable under Definition 3 with respect to any distribution that is locally λ-close to constant in a finite number of convex regions.

4.4 A Generic Decomposition Procedure

In general, given the concept class C and the distribution D, we can use the use the following procedure to produce a data-dependent hierarchy. We start with the whole space C; we put all the concepts with infinite disagreement coefficient in a set A, and structure the others based on increasing disagreement coefficient as $S_1 \subset S_2 \subset \cdots$. If no classifier in A has infinite disagreement coefficient with respect to A, then we define $C_0 = A$, and we redefine all sets C_i as $C_i \cup A$. Otherwise, we recurse on A, and get back a structure $C'_0 \subset C'_1 \subset C'_2 \subset \cdots$. In this case, we redefine all C_i as $C_i \cup C'_i$, $i \geq 1$ and let C_0 be C'_0.

Combining results in Hanneke[14] with the fact that the union of any two sets with finite disagreement coefficient also has finite disagreement coefficient yields the following theorem.

Theorem 4.4 *If the above procedure stops after a finite number of recursive calls, then the pair (C,D) is learnable under Definition 3.*

Note that we could equivalently replace the disagreement coefficient with a slightly modified version of the splitting index; in particular, given some "optimal" method $\tau(\epsilon)$ for setting the τ parameter in the original definition [9] as a function of ϵ, we refer to splitting index ρ as the limiting splitting index as $\epsilon \to 0$.

Examples and Relationship to the Splitting Index Analysis: This procedure further highlights the difference between this analysis and the analysis in [9].

Using the procedure, we can obtain reasonable hierarchies that allow us to learn under Definition 3 for all of the geometric concepts described so far. As an example, the class of intervals has only one classifier with infinite disagreement coefficient (or zero splitting index) with respect to the full space C, and this is the all-negative function. Every other classifier has some width w to its interval, so the disagreement coefficient is $\approx 1/w$ (splitting index is $\approx w$) . Thus we define C_0 to be the set containing only the all-negative function. For all $i > 0$, define C_i to contain all intervals with width at least 2^{-i} together with the all-negative function. The all-negative classifier has finite disagreement coefficient (nonzero splitting index) within C_0, so we're done.

The reason we sometimes need a recursive procedure is that sometimes when we put all the classifiers with infinite disagreement coefficient *with respect to the full space* C into one set A together, some of them still have infinite disagreement coefficient *with respect to the set* A. This happens, for example, for the space C of unions of two intervals. Almost any classifier that can be represented as at most a single interval has infinite disagreement coefficient with respect to C. When we take all those classifiers together in one set A, the set A becomes essentially isomorphic to the class of intervals. As we know, the all-negative function has infinite disagreement coefficient with respect to the set of single intervals, so we need another recursive call. That will return a structure that has in C'_0 the all-negative and anything that is zero-distance to it, and in C'_i everything in C'_0 together with all the functions that are zero-distance to an interval with the width at least 2^{-i} (let's say). Therefore, in the final structure, C_i will contain everything in C'_i along with every union of two intervals, where (let's say) both intervals have width at least 2^{-i} and are separated by a gap of width at least 2^{-i}.

4.5 Lower Bounds

We show in the following that not every pair (C, D) is learnable under Definition 3. This is true even if C is a class of geometric concepts if the distribution is especially bad.

Theorem 4.5 *There exists a pair (C, D) that is not learnable under Definition 3. Moreover, there exists such a pair for which C has VC dimension* 1.

Proof Sketch : Let T be a fixed infinite tree in which each node at depth i has c_i children; c_i will be 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.

Let D be 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 will show that it is possible to define the parameters above in such a way that for any ϵ_0, there exists some $\epsilon < \epsilon_0$ such that for some level j, $p_j = \epsilon$ and $c_{j-1} \geq (1/p_j)^{1/2} = (1/\epsilon)^{1/2}$. This will imply 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 since it has probability ϵ. We can argue that in order to find that node, we need to check a constant fraction of the children of the node's parent, so we need to query $O(c_{j-1})$ 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 probability distribution over i that goes to 0 in the limit completes the proof. ∎

Note: This bound can be tightened to show that there exist pairs of classes and distributions which can only be learned with $\Omega((1/\epsilon)^\alpha)$ labels for α arbitrarily close to 1. It is not yet known whether a $\Omega(1/\epsilon)$ lower bound holds in the asymptotic model; we are currently thinking about this important question.

Note: 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 $\mathbb{R}^2$.

5 Conclusions

One can interpret this work as answering the "placing bets on hypotheses" question that appears in Dasgupta's paper [9]; although the learning algorithms in our model might not always know when they have found a good hypothesis, they will always output a best guess. However, the construction of the hierarchy is more subtle and quite different than previous work in the context of supervised or semi-supervised learning [18, 2].

Most important is the implication of our analysis: in many interesting cases where it was previously believed that active learning could not help, active learning *does help asymptotically*. We have formalized this idea and illustrated it with a number of examples 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, under this asymptotic model of learning we now see that active learning does help significantly in all *but* a few contrived and unrealistic problems.

There are some interesting open problems within this framework. Perhaps the two most interesting are formulating necessary and sufficient conditions for learnability with an exponential rate, and determining whether active learning can *always* (for all (C, D)) have a sample complexity at most $o(1/\epsilon)$, or whether there are cases where the old $\Omega(1/\epsilon)$ still applies.

References

[1] M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *ICML*, 2006.

[2] 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.

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

[4] A. Blum. Machine learning theory. *Essay*, 2007.

[5] A. Blum and T. M. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, 1998.

[6] R. Castro and R. Nowak. Minimax bounds for active learning. 2007. COLT.

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

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

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

[10] S. Dasgupta, D. J. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. In *Neural Information Processing Systems (NIPS)*, 2007.

[11] S. Dasgupta, A. Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. In *COLT*, 2005.

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

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

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

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

[16] T. Joachims. Transductive inference for text classification using support vector machines. In *Proc. ICML*, pages 200–209, 1999.

[17] T. Mitchell. The discipline of machine learning. *CMU-ML-06 108*, 2006.

[18] 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.

[19] L.G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

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