Theoretical Foundations of Active Learning

Steve Hanneke

May 2009
CMU-ML-09-106

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Avrim Blum
Sanjoy Dasgupta
Larry Wasserman
Eric P. Xing

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

This thesis is dedicated to the many teachers who have helped me along the way.

Abstract

I study the informational complexity of active learning in a statistical learning theory framework. Specifically, I derive bounds on the rates of convergence achievable by active learning, under various noise models and under general conditions on the hypothesis class. I also study the theoretical advantages of active learning over passive learning, and develop procedures for transforming passive learning algorithms into active learning algorithms with asymptotically superior label complexity. Finally, I study generalizations of active learning to more general forms of interactive statistical learning.

Acknowledgments

There are so many people I am indebted to for helping to make this thesis, and indeed my entire career, possible. To begin, I am grateful to the faculty of Webster University, where my journey into science truly began. Support from the teachers I was privileged to have there, including Gary Coffman, Britt-Marie Schiller, Ed and Anna B. Sakurai, and John Aleshunas, to name a few, inspired in me a deep curiosity and hunger for understanding. I am also grateful to my teachers and colleagues at the University of Illinois. In particular, Dan Roth deserves my appreciation for nothing less than teaching me how to do effective research; my experience as an undergraduate working with Dan and the other members of his Cognitive Computation Group shaped my fundamental approach to research.

I would like to thank several of the professors at Carnegie Mellon. This institution is an exciting place to be for anyone interested in machine learning; it has been an almost ideal setting for me to develop a mature knowledge of learning theory, and is generally a warm place to call home (metaphorically speaking). I would specifically like to thank my advisors (past and present), Eric Xing, Larry Wasserman, and Steve Fienberg, whose knowledge, insights, and wisdom have been invaluable at various times during the past four years; I am particularly grateful to them for allowing me the freedom to pursue the topics I am passionate about. Several students at Carnegie Mellon have also helped to enrich this experience. In particular, Nina Balcan has been a source for many many interesting, insightful and always exciting discussions.

In addition to those mentioned above, I am also grateful to several colleagues who have been invaluable at times through insightful comments, advice, or discussions, and who have generally made me feel a welcomed part of the larger learning theory community. These include John Langford, Sanjoy Dasgupta, Avrim Blum, Rob Nowak, Leo Kontorovich, Vitaly Feldman, and Elad Hazan, among others.

I would also like to thank Eric Xing, Larry Wasserman, Avrim Blum, and Sanjoy Dasgupta for serving on my thesis committee.

Finally, on a personal note, I would like to thank my parents, grandparents, brother, and all of my family and friends, for helping me understand the value of learning while growing up, and for their continued unwavering support in all that I do.

Contents

Chapter 1

Notation and Background

1.1 Introduction

In active learning, a learning algorithm is given access to a large pool of unlabeled examples, and is allowed to request the label of any particular examples from that pool, interactively. The objective is to learn a function that accurately predicts the labels of new examples, while requesting as few labels as possible. This contrasts with passive learning, where the examples to be labeled are chosen randomly. In comparison, active learning can often significantly decrease the work load of human annotators by more carefully selecting which examples from the unlabeled pool should be labeled. This is of particular interest for learning tasks where unlabeled examples are available in abundance, but label information comes only through significant effort or cost.

In the passive learning literature, there are well-known bounds on the rate of convergence of the loss of an estimator, as a function of the number of labeled examples observed [e.g., Benedek and Itai, 1988, Blumer et al., 1989, Koltchinskii, 2006, Kulkarni, 1989, Long, 1995, Vapnik, 1998]. However, significantly less is presently known about the analogous rate in active learning: namely, the rate of convergence of the loss of an estimator, as a function of the number of label requests made by an active learning algorithm.

In this thesis, I will outline some recent progress I have been able to make toward understand-

ing the achievable rates of convergence by active learning, along with algorithms that achieve them. I will also describe a few of the many open problems remaining on this topic.

The thesis begins with a brief survey of the history of this topic, along with an introduction to the formal definitions and notation that will be used throughout the thesis. It then describes some of my contributions to this area. To begin, Chapter 2 describes some rates of convergence achievable by active learning algorithms under various noise conditions, as quantified by a new complexity parameter called the *disagreement coefficient*. It then continues by exploring an interesting distinction between two different notions of label complexity: namely, *verifiable* and *unverifiable*. This distinction turns out to be extremely important for active learning, and Chapter 3 explains why. Following this, Chapter 4 describes a reductions-based approach to active learning, in which the goal is to transform passive learning algorithms into active learning algorithms having strictly superior label complexity. The results in that chapter are surprisingly general and of deep theoretical significance. The thesis concludes with Chapter 5, which describes some preliminary work on generalizations of active learning to more general types of interactive statistical learning, proving results at a higher level of abstraction, so that they can apply to a variety of interactive learning protocols.

1.2 A Simple Example: Thresholds

We begin with the canonical toy example illustrating the potential benefits of active learning. Suppose we are tasked with finding, somewhere in the interval $[0, 1]$, a threshold value x; we are scored based on how close our guess is to the true value, so that if we guess x equals z for some $z \in [0, 1]$, we are awarded $1 - |x - z|$ points. There is an oracle at hand who knows the value of x, and given any point $x' \in [0, 1]$ can tell us whether $x' \geq x$ or $x' < x$.

The passive learning strategy can be simply described as taking points uniformly at random from the interval $[0, 1]$ and asking the oracle whether each point is $\geq x$ or $< x$ for every one. After a number of these random queries, the passive learning strategy chooses its guess somewhere

between x_1' = the largest x' that it knows is $< x$, and x_2' = the smallest x' it knows is $\geq x$ (say it guesses $\frac{x_1' + x_2'}{2}$). By a simple argument, if the passive strategy asks about n points, then the expected distance between x_1' and x_2' is at least $\frac{1}{n+1}$ (say for $x = 1/2$), so we expect the passive strategy's guess to be off by some amount $\geq \frac{1}{2(n+1)}$.

On the other hand, suppose instead of asking the oracle about every one of these random points, we instead look at each one sequentially, and only ask about a point if it is between the current x_1' and the current x_2'; that is, we only ask about a point if it is *not* greater than a point x' known to be $\geq x$ and *not* less than a point known to be $< x$. This certainly seems to be a reasonable modification to our strategy, since we already know how the oracle would respond for the points we choose not to ask about. In this case, if we ask the oracle about n points, each one reduces the width of the interval $[x_1', x_2']$ at that moment by some factor β_i. These n factors β_i are upper bounded by n independent $Uniform([1/2, 1])$ random variables (representing the fraction of the interval on the larger side of the x'), so that the expected final width of $[x_1', x_2']$ is at most $(\frac{3}{4})^n \leq exp\{-n/4\}$. Therefore, we expect this modified strategy's guess to be off by at most half this amount.[1]

As we will see, this modified strategy is a special case of an active learning algorithm I will refer to as CAL (after its discoverers, Cohn, Atlas, and Ladner [1994]) or Algorithm 0, which I introduce in Section 1.4. The gap between the passive strategy, which can only reduce the distance between the guess and the true threshold at a *linear* rate $\Omega(n^{-1})$, and the active strategy, which can reduce this distance at an *exponential* rate $\frac{1}{2}(\frac{3}{4})^n$, can be substantial. For instance, with $n = 20$, $\frac{1}{2(n+1)} \approx .024$ while $\frac{1}{2}(\frac{3}{4})^n \approx .0016$, better than an order of magnitude improvement. We will see several cases below where these types of exponential improvements are achievable by active learning algorithms for much more realistic learning problems, but in many cases the proofs can be thought of as simple generalizations of this toy example.

---

[1]Of course, the optimal strategy for this task always asks about $\frac{x_1' + x_2'}{2}$, and thus closes the gap at a rate 2^{-n}. However, the less aggressive strategy I described here illustrates a simple case of an algorithm we will use extensively below.

1.3 Notation

Perhaps the simplest active learning task is binary classification, and we will focus primarily on that task. Let $\mathcal{X}$ be an *instance space*, comprising all possible examples we may ever encounter. $\mathbb{C}$ is a set of measurable functions $h : \mathcal{X} \rightarrow \{-1, 1\}$, known as the *concept space* or *hypothesis class*. We also overload this notation so that for $m \in \mathbb{N}$ and a sequence $S = \{x_1, \ldots, x_m\} \in \mathcal{X}^m$, $h(S) = (h(x_1), h(x_2), \ldots, h(x_m))$. We denote by d the VC dimension of $\mathbb{C}$, and by $\mathbb{C}[m] = \max_{S \in \mathcal{X}^m} |\{h(S) : h \in \mathbb{C}\}|$ the shatter coefficient (a.k.a. growth function) value at m [Vapnik, 1998]. Generally, we will refer to any $\mathbb{C}$ with finite VC dimension as a *VC class*. $\mathbb{D}$ is a known set of probability distributions on $\mathcal{X} \times \{-1, 1\}$, in which there is some unknown *target distribution* $\mathcal{D}_{XY}$. I also denote by $\mathcal{D}[\mathcal{X}]$ the marginal of $\mathcal{D}$ over $\mathcal{X}$. There is additionally a sequence of examples $(x_1, y_1), (x_2, y_2), \ldots$ sampled i.i.d. according to $\mathcal{D}_{XY}$. In the active learning setting, the y_i values are hidden from the learning algorithm until requested. Define $\mathcal{Z}_m = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, a finite sequence consisting of the first m examples.

For any $h \in \mathbb{C}$ and distribution $\mathcal{D}'$ over $\mathcal{X} \times \{-1, 1\}$, let $er_{\mathcal{D}'}(h) = \mathbb{P}_{(X,Y) \sim \mathcal{D}'}\{h(X) \neq Y\}$, and for $S = \{(x_1', y_1'), (x_2', y_2'), \ldots, (x_m', y_m')\} \in (\mathcal{X} \times \{-1, 1\})^m$, define the empirical error $er_S(h) = \frac{1}{2m} \sum_{i=1}^m |h(x_i') - y_i'|$. When $\mathcal{D}' = \mathcal{D}_{XY}$ (the target distribution), we abbreviate the former by $er(h) = er_{\mathcal{D}_{XY}}(h)$, and when $S = \mathcal{Z}_m$, we abbreviate the latter by $er_m(h) = er_{\mathcal{Z}_m}(h)$. The *noise rate*, denoted $\nu(\mathbb{C}, \mathcal{D}_{XY})$, is defined as $\nu(\mathbb{C}, \mathcal{D}) = \inf_{h \in \mathbb{C}} er_{\mathcal{D}}(h)$; we abbreviate this by ν when $\mathbb{C}$ and $\mathcal{D} = \mathcal{D}_{XY}$ are clear from the context (i.e., the concept space and target distribution). We also define $\eta(x; \mathcal{D}) = \mathbb{P}_{\mathcal{D}}(Y = 1|x)$, and define the *Bayes error rate*, denoted $\beta(\mathcal{D})$, as $\beta(\mathcal{D}) = \mathbb{E}_{X \sim \mathcal{D}[\mathcal{X}]}[\min\{\eta(X; \mathcal{D}), 1 - \eta(X; \mathcal{D})\}]$, which represents the best achievable error rate by *any* classifier; we will also refer to the Bayes optimal classifier, denoted h^*, defined as $h_{\mathcal{D}}^*(x) = 2\mathbb{1}[\eta(x; \mathcal{D}) \geq 1/2] - 1$; again, for $\mathcal{D} = \mathcal{D}_{XY}$, we may abbreviate this as $\eta(x) = \eta(x; \mathcal{D}_{XY})$, $\beta = \beta(\mathcal{D}_{XY})$, and $h^* = h_{\mathcal{D}_{XY}}^*$.

For concept space $\mathcal{H}$ and distribution $\mathcal{D}'$ over $\mathcal{X}$, for any measurable $h : \mathcal{X} \rightarrow \{-1, 1\}$ and

any $r > 0$, define

$$B_{\mathcal{H},\mathcal{D}'}(h,r) = \{h' \in \mathcal{H} : \mathbb{P}_{X \sim \mathcal{D}'}(h(X) \neq h'(X)) \leq r\}.$$

When $\mathcal{H} = \mathbb{C}$, $\mathcal{D}' = \mathcal{D}_{XY}[\mathcal{X}]$, or both are true, we may simply write $B_{\mathcal{D}'}(h,r)$, $B_{\mathcal{H}}(h,r)$, or $B(h,r)$ respectively. For concept space $\mathcal{H}$ and distribution $\mathcal{D}'$ over $\mathcal{X} \times \{-1,+1\}$, for any $\epsilon \in [0,1]$, define the ϵ−*minimal set*, $\mathcal{H}(\epsilon;\mathcal{D}') = \{h \in \mathcal{H} : er_{\mathcal{D}'}(h) - \nu(\mathcal{H},\mathcal{D}') \leq \epsilon\}$. When $\mathcal{D}' = \mathcal{D}_{XY}$ (target distribution) and is clear from the context, we abbreviate this by $\mathcal{H}(\epsilon) = \mathcal{H}(\epsilon;\mathcal{D}_{XY})$. For a concept space $\mathcal{H}$ and distribution $\mathcal{D}'$ over $\mathcal{X}$, define the *diameter* of $\mathcal{H}$ as $diam(\mathcal{H};\mathcal{D}') = \sup_{h_1,h_2 \in \mathcal{H}} \mathbb{P}_{X \sim \mathcal{D}}(h_1(X) \neq h_2(X))$; as before, when $\mathcal{D}' = \mathcal{D}_{XY}[\mathcal{X}]$ and is clear from the context, we will abbreviate this as $diam(\mathcal{H}) = diam(\mathcal{H};\mathcal{D}_{XY}[\mathcal{X}])$.

Also define the *region of disagreement* of a concept space $\mathcal{H}$ as

$$DIS(\mathcal{H}) = \{x \in \mathcal{X} : \exists h_1, h_2 \in \mathcal{H} \text{ s.t. } h_1(x) \neq h_2(x)\}.$$

Also, for a concept space $\mathcal{H}$, distribution $\mathcal{D}$ over $\mathcal{X} \times \{-1,+1\}$, $\epsilon \in [0,1]$, and $m \in \mathbb{N}$, define the *expected continuity modulus* as

$$\omega_{\mathcal{H}}(m,\epsilon;\mathcal{D}) = \mathbb{E}_{S \sim \mathcal{D}^m} \sup_{\substack{h_1,h_2 \in \mathcal{H}: \\ \mathbb{P}_{X \sim \mathcal{D}[\mathcal{X}]}\{h_1(X) \neq h_2(X)\} \leq \epsilon}} |(er_{\mathcal{D}}(h_1) - er_S(h_1)) - (er_{\mathcal{D}}(h_2) - er_S(h_2))|.$$

At this point, let us distinguish between some particular settings, distinguished by the definition of $\mathbb{D}$ as one of the following sets of distributions.

- $\mathcal{Agnostic} = \{ \text{ all } \mathcal{D}\}$ (the set of all joint distributions on $\mathcal{X} \times \{-1,+1\}$).

- $\mathcal{BenignNoise}(\mathbb{C}) = \{\mathcal{D} : \nu(\mathbb{C},\mathcal{D}) = \beta(\mathcal{D})\}$.

- $\mathcal{Tsybakov}(\mathbb{C},\kappa,\mu) = \left\{\mathcal{D} : \forall \epsilon > 0, diam(\mathbb{C}(\epsilon;\mathcal{D});\mathcal{D}) \leq \mu\epsilon^{\frac{1}{\kappa}}\right\}$, (for any finite parameters $\kappa \geq 1, \mu > 0$).

- $\mathcal{Entropy}_{[]}(\mathbb{C},\alpha,\rho) = \left\{\mathcal{D} : \forall m \in \mathbb{N} \text{ and } \epsilon \in [0,1], \omega_{\mathbb{C}}(m,\epsilon;\mathcal{D}) \leq \alpha\epsilon^{\frac{1-\rho}{2}}m^{-1/2}\right\}$, (for any finite parameters $\alpha > 0, \rho \in (0,1)$).

- $\mathcal{UniformNoise}(\mathbb{C}) = \{\mathcal{D} : \exists \alpha \in [0,1/2), f \in \mathbb{C} \text{ s.t. } \forall x \in \mathcal{X}, \mathbb{P}_{\mathcal{D}}(Y \neq f(x)|X = x) = \alpha\}$.

5

- $\mathcal{R}ealizable(\mathbb{C}) = \{\mathcal{D} : \exists f \in \mathbb{C} \text{ s.t. } er_{\mathcal{D}}(f) = 0\}$.

- $\mathcal{R}ealizable(\mathbb{C}, \mathcal{D}_X) = \mathcal{R}ealizable(\mathbb{C}) \cap \{\mathcal{D} : \mathcal{D}[\mathcal{X}] = \mathcal{D}_X\}$, (for any given marginal distribution $\mathcal{D}_X$ over $\mathcal{X}$).

$\mathcal{A}gnostic$ is the most general setting we will study, and is referred to as the *agnostic case*, where $\mathbb{D}$ is the set of *all* joint distributions. However, at times we will consider the other sets, which represent various restrictions of $\mathcal{A}gnostic$. In particular, the set $\mathcal{B}enignNoise(\mathbb{C})$ essentially corresponds to situations in which the lack of a perfect classifier in $\mathbb{C}$ is due to stochasticity of the labels, not model misspecification. $\mathcal{T}sybakov(\mathbb{C}, \kappa, \mu)$ is a further restriction, introduced by Mammen and Tsybakov [1999] and Tsybakov [2004], which (informally) represents those distributions having reasonably low noise near the optimal decision boundary (see Chapter 2 for further explanations). $\mathcal{E}ntropy_{[]}(\mathbb{C}, \alpha, \rho)$ represents the *finite entropy with bracketing* condition common to the empirical processes literature [e.g., Koltchinskii, 2006, van der Vaart and Wellner, 1996]. $\mathcal{U}niformNoise(\mathbb{C})$ represents a (rather artificial) subset of $\mathcal{B}enignNoise(\mathbb{C})$ in which every point has the same probability of being labeled opposite to the optimal label. $\mathcal{R}ealizable(\mathbb{C})$ represents the *realizable case*, popularized by the PAC model of passive learning [Valiant, 1984], in which there is a perfect classifier in the concept space; in this setting, we will refer to this perfect classifier as the *target function*, typically denoted h^*. $\mathcal{R}ealizable(\mathbb{C}, \mathcal{D}_X)$ represents a restriction of the realizable case, which we will refer to as the *fixed-distribution realizable case*; this corresponds to learning problems where the marginal distribution over $\mathcal{X}$ is known *a priori*.

Several of the more restrictive sets above may initially seem unrealistic. However, they become more plausible when we consider fairly complex concept spaces (e.g., nonparametric spaces). On the other hand, some (specifically, $\mathcal{U}niformNoise(\mathbb{C})$ and $\mathcal{R}ealizable(\mathbb{C}, \mathcal{D}_X)$) are basically toy scenarios, which are only explored as stepping stones toward more realistic assumptions.

We now define the primary quantities of interest throughout this thesis: namely, rates of

convergence, and label complexity.

Definition 1.1. *(Unverifiable rate) An algorithm $\mathcal{A}$ achieves a rate of convergence $\bar{R}(\cdot, \cdot)$ on expected excess error with respect to $\mathbb{C}$ if for any $\mathcal{D}_{XY}$ and $n \in \mathbb{N}$, if $h_n = \mathcal{A}(n)$ is the algorithm's output after at most n label requests, for target distribution $\mathcal{D}_{XY}$, then*

$$\mathbb{E}[er(h_n)] - \nu(\mathbb{C}, \mathcal{D}_{XY}) \leq \bar{R}(n, \mathcal{D}_{XY}).$$

An algorithm $\mathcal{A}$ achieves a rate of convergence $R(\cdot, \cdot, \cdot)$ on confidence-bounded excess error with respect to $\mathbb{C}$ if, for any $\mathcal{D}_{XY}$, $\delta \in (0, 1)$, and $n \in \mathbb{N}$, if $h_n = \mathcal{A}(n)$ is the algorithm's output after at most n label requests, for target distribution $\mathcal{D}_{XY}$, then

$$\mathbb{P}(er(h_n) - \nu(\mathbb{C}, \mathcal{D}_{XY}) \leq R(n, \delta, \mathcal{D}_{XY})) \geq 1 - \delta.$$

Definition 1.2. *(Verifiable rate) An algorithm $\mathcal{A}$ achieves a rate of convergence $R(\cdot, \cdot, \cdot)$ on an accessible bound on excess error with respect to $\mathbb{C}$, under $\mathbb{D}$ if, for any $\mathcal{D}_{XY} \in \mathbb{D}$, $\delta \in (0, 1)$, and $n \in \mathbb{N}$, if $(h_n, \hat{\epsilon}_n) = \mathcal{A}(n)$ is the algorithm's output after at most n label requests, for target distribution $\mathcal{D}_{XY}$, then*

$$\mathbb{P}(er(h_n) - \nu(\mathbb{C}, \mathcal{D}_{XY}) \leq \hat{\epsilon}_n \leq R(n, \delta, \mathcal{D}_{XY})) \geq 1 - \delta.$$

I will refer to Definition 1.2 as a *verifiable rate* under $\mathbb{D}$, for short. If ever I simply refer to the *rate*, I will mean Definition 1.1. To distinguish these two notions of convergence rates, I may sometimes refer to Definition 1.1 as the *unverifiable rate* or the *true rate*. Clearly any algorithm that achieves a verifiable rate R also achieves R as an unverifiable rate. However, we will see interesting cases where the reverse is not true.

At times, it will be necessary to express some results in terms of the number of label requests required to guarantee a certain error rate. This quantity is referred to as the *label complexity*, and is defined quite naturally as follows.

Definition 1.3. *(Unverifiable label complexity) An algorithm $\mathcal{A}$ achieves a* label complexity $\bar{\Lambda}(\cdot, \cdot)$ *for expected error, if for any $\mathcal{D}_{XY}$, $\forall \epsilon \in (0, 1)$, $\forall n \geq \bar{\Lambda}(\epsilon, \mathcal{D}_{XY})$, if $h_n = \mathcal{A}(n)$ is the algorithm's output after at most n label requests, for target distribution $\mathcal{D}_{XY}$, then $\mathbb{E}[er(h_n)] \leq \epsilon$.*

An algorithm $\mathcal{A}$ achieves a label complexity $\Lambda(\cdot, \cdot, \cdot)$ *for confidence-bounded error, if for any $\mathcal{D}_{XY}$, $\forall \epsilon, \delta \in (0, 1)$, $\forall n \geq \Lambda(\epsilon, \delta, \mathcal{D}_{XY})$, if $h_n = \mathcal{A}(n)$ is the algorithm's output after at most n label requests, for target distribution $\mathcal{D}_{XY}$, then $\mathbb{P}(er(h_n) \leq \epsilon) \geq 1 - \delta$.*

Definition 1.4. *(Verifiable label complexity) An algorithm $\mathcal{A}$ achieves a* verifiable label complexity $\Lambda(\cdot, \cdot, \cdot)$ *for $\mathbb{C}$ under $\mathbb{D}$ if it achieves a verifiable rate R with respect to $\mathbb{C}$ under $\mathbb{D}$ such that, for any $\mathcal{D}_{XY} \in \mathbb{D}$, $\forall \delta \in (0, 1)$, $\forall \epsilon \in (0, 1)$, $\forall n \geq \Lambda(\epsilon, \delta, \mathcal{D}_{XY})$, $R(n, \delta, \mathcal{D}_{XY}) \leq \epsilon$.*

Again, to distinguish between these definitions, I may sometimes refer to the former as the *unverifiable label complexity* or the *true label complexity*. Also, throughout the thesis, I will maintain the convention that whenever I refer to a "rate R" or "label complexity Λ," I refer to the confidence-bounded variety, and similarly when I refer to a "rate $\bar{R}$" or "label complexity $\bar{\Lambda}$," in those cases I refer to the version of the definition for *expected* error rates.

A brief note on measurability:

Throughout this thesis, we will let $\mathbb{E}$ and $\mathbb{P}$ (and indeed *any* reference to "probability") refer to the *outer* expectation and probability [van der Vaart and Wellner, 1996], so that quantities such as $\mathbb{P}(DIS(B(h, r)))$ are well defined, even if $DIS(B(h, r))$ is not measurable.

1.4   A Simple Algorithm Based on Disagreement

One of the earliest, and most elegant, theoretically sound active learning algorithms for the realizable case was provided by Cohn, Atlas, and Ladner [1994]. Under the assumption that there exists a perfect classifier in $\mathbb{C}$, they proposed an algorithm which processes unlabeled examples in sequence, and for each one it determines whether there exists a classifier in $\mathbb{C}$ consistent with all previously observed labels that labels this new example $+1$ *and* one that labels this example

-1; if so, the algorithm requests the label, and otherwise it does not request the label; after n label requests, the algorithm returns any classifier consistent with all observed labels. In some sense, this algorithm corresponds to the very least we could expect of an active learning algorithm, as it never requests the label of an example it can derive from known information, but otherwise makes no effort to search for informative examples. We can equivalently think of this algorithm as maintaining two sets: $V \subseteq \mathbb{C}$ is the set of candidate hypotheses still under consideration, and $R = DIS(V)$ is their region of disagreement. We can then think of the algorithm as requesting a random labeled example from the conditional distribution of $\mathcal{D}_{XY}$ given that $X \in R$, and subsequently removing from V any classifier inconsistent with the observed label.

Most of the active learning algorithms we study in subsequent chapters will be, in some way, variants of, or extensions to, this basic procedure. In fact, at this writing, all of the published general-purpose agnostic active learning algorithms achieving nontrivial improvements are derivatives of Algorithm 0. A formal definition of the algorithm is given below.

Algorithm 0
Input: hypothesis class $\mathcal{H}$, label budget n
Output: classifier $h_n \in \mathcal{H}$ and error bound $\hat{\epsilon}_n$

---
0. $V_0 \leftarrow \mathcal{H}, q \leftarrow 0$
1. For $m = 1, 2, \ldots$
2.   If $\exists h_1, h_2 \in V_q$ s.t. $h_1(x_m) \neq h_2(x_m)$,
3.   Request y_m
4. $q \leftarrow q + 1$
5. $V_q \leftarrow \{h \in V_{q-1} : h(x_m) = y_m\}$
6.   If $q = n$, Return an arbitrary classifier $h_n \in V_n$ and value $\hat{\epsilon}_n = diam(V_n)$

One of the most appealing properties of this algorithm, besides its simplicity, is the fact that it makes extremely efficient use of the unlabeled examples; in fact, supposing the algorithm processes m unlabeled examples before returning, we can take the classifier h_n and label all of the examples we skipped over (i.e., those we did *not* request the labels of); this actually produces a set of m perfectly labeled examples, which we can feed into our favorite passive learning algorithm, even though we only requested the labels of a subset of those examples. This fact also provides a simple proof that $er(h_n)$ can be bounded by a quantity that decreases to zero (in

probability) with n: namely, $diam(V_n)$. However, Cohn et al. [1994] did not provide any further characterization of the rates achieved by this algorithm in general. For this, we must wait until Chapter 2, where I provide the first general characterization of the rates achieved by this method in terms of a quantity I call the disagreement coefficient.

1.5   A Lower Bound

When beginning an investigation into the achievable rates, it is natural to first ask what we can possibly hope to achieve, and what results are definitely not possible. That is, what are some fundamental limits on what this type of learning is capable of. This type of question was investigated by Kulkarni et al. [1993] in a more general setting. Informally, the reasoning is that each label request can communicate at most one bit of information. So the best we can hope for is something logarithmic in the "size" of the hypothesis class. Of course, for infinite hypothesis classes this makes no sense, but with the help of a notion of *cover size*, Kulkarni et al. [1993] were able to prove the analogous result.

Specifically, let $N(\epsilon)$ be the size of the smallest set V of classifiers in $\mathbb{C}$ such that $\forall h \in \mathbb{C}, \exists h' \in V : \mathbb{P}_{X \sim \mathcal{D}}[h(X) \neq h'(X)] \leq \epsilon$, for some distribution $\mathcal{D}$ over X. Then any achievable label complexity Λ has the property that $\forall \epsilon > 0$, $\sup\limits_{\mathcal{D}_{XY} \in \mathcal{R}ealizable(\mathbb{C},\mathcal{D})} \Lambda(\epsilon, \delta, \mathcal{D}_{XY}) \geq \log_2[(1{-}\delta)N(2\epsilon)]$.

Since we can often get a reasonable estimate of $N(\epsilon)$ by its distribution-free upper bound $2\left(\frac{2e}{\epsilon}\ln\frac{2e}{\epsilon}\right)^d$ [Haussler, 1992], we can often expect our rates to be at best $exp\{-cn/d\}$ for some constant c. In particular, rather than working with $N(\epsilon)$ in the results below, I will typically formulate upper bounds in terms of d; in most of these cases, some variant of $\log N(\epsilon)$ could easily be substituted to achieve a tighter bound (by using the cover as a hypothesis class instead of the full space), closer in spirit to this lower bound.

10

1.6 Splitting Index

Over the past decade, several special-purpose active learning algorithms were proposed, but notably lacking was a general theory of convergence rates for active learning. This changed in 2005 when Dasgupta published his theory of splitting indices [Dasgupta, 2005].

As before, this section is restricted to the *realizable case*. Let $Q \subseteq \{\{h_1, h_2\} : h_1, h_2 \in \mathbb{C}\}$ be a finite set of unordered pairs of classifiers from $\mathbb{C}$. For $x \in \mathcal{X}$ and $y \in \{-1, +1\}$, define $Q_x^y = \{\{h_1, h_2\} \in Q : h_1(x) = h_2(x) = y\}$. A point $x \in \mathcal{X}$ is said to ρ-*split* Q if

$$\max_{y \in \{-1, +1\}} |Q_x^y| \leq (1 - \rho)|Q|.$$

We say $\mathcal{H} \subseteq \mathbb{C}$ is (ρ, Δ, τ)-*splittable* if for all finite $Q \subseteq \{\{h_1, h_2\} \subseteq \mathbb{C} : \mathbb{P}(h_1(X) \neq h_2(X)) > \Delta\}$,

$$\mathbb{P}(X \ \rho\text{-splits} \ Q) \geq \tau.$$

A large value of ρ for a reasonably large τ indicates that there are highly informative examples that are not too rare. Dasgupta effectively proves the following results.

> **Theorem 1.5.** *For any VC class $\mathbb{C}$, for some universal constant $c > 0$, there is an algorithm with verifiable label complexity Λ for $\mathcal{R}ealizable(\mathbb{C})$ such that, for any $\epsilon \in (0, 1)$, $\delta \in (0, 1)$, and $\mathcal{D}_{XY} \in \mathcal{R}ealizable(\mathbb{C})$, if $B(h^*, 4\Delta)$ is (ρ, Δ, τ)-splittable for all $\Delta \geq \epsilon/2$, then*
> $$\Lambda(\epsilon, \delta, \mathcal{D}_{XY}) \leq c\frac{d}{\rho} \log \frac{d}{\epsilon \delta \tau} \log \frac{1}{\epsilon}.$$

The value ρ has been referred to as the *splitting index*. It can be useful for quantifying the verifiable rates for a variety of problems in the realizable case. For example, Dasgupta [2005] uses it to analyze the problem where $\mathbb{C}$ is the class of homogeneous linear separators in d dimensions, and $\mathcal{D}_{XY}[\mathcal{X}] = \mathcal{D}$ is the uniform distribution on the unit d-dimensional sphere. He shows that this problem is $(1/2, \epsilon, \epsilon)$-splittable for any $\epsilon > 0$ for any target in $\mathbb{C}$. This implies a verifiable rate for $\mathcal{R}ealizable(\mathbb{C}, \mathcal{D})$ of

$$R(n, \delta, \mathcal{D}_{XY}) \propto \frac{d}{\delta} \cdot exp\left\{-c'\sqrt{\frac{n}{d}}\right\}$$

for a constant $c' > 0$. This rate was previously known for other algorithms [e.g., Dasgupta et al., 2005], but had not previously been derived as a special case of such a general analysis.

1.7 Agnostic Active Learning

Though each of the preceding analyses provides valuable insights into the nature of active learning, they also suffer the drawback of reliance on the realizability assumption. In particular, that there is no label noise, and that the Bayes optimal classifier is in $\mathbb{C}$, are severe and often unrealistic assumptions. We would ideally like an analysis of the agnostic case as well. However, the aforementioned algorithms (e.g., CAL, and the algorithm achieving the splitting index bounds) no longer function properly in the presence of nonzero noise rates. So we need to start from the basics and build new techniques that are robust to noise conditions.

To begin, we may again ask what we might hope to achieve. That is, are there fundamental information-theoretic limits on what we can do with this type of learning? This question was investigated by Kääriäinen [2006]. In particular, he was able to prove that for basically any nontrivial marginal $\mathcal{D}$ over $\mathcal{X}$, noise rate ν, number n, and active learning algorithm, there is some distribution $\mathcal{D}_{XY}$ with marginal $\mathcal{D}$ and noise rate ν such that the algorithm's achieved rate $R(n, \delta, \mathcal{D}_{XY})$ at n satisfies (for some constant $c > 0$)

$$R(n, \delta, \mathcal{D}_{XY}) \geq c\sqrt{\frac{\nu^2 \log(1/\delta)}{n}}.$$

Furthermore, this result was improved by Beygelzimer, Dasgupta, and Langford [2009] to

$$R(n, 3/4, D_{XY}) \geq c\sqrt{\frac{\nu^2 d}{n}}.$$

Considering that rates $\propto \sqrt{\frac{\nu d \log(1/\delta)}{n}}$ are achievable in passive learning, this indicates that, even for concept spaces that had exponential rates in the realizable case, any bound on the verifiable rates that shows significant improvement (more than a multiplicative factor of $\sqrt{\nu}$) in the dependence on n for nonzero noise rates must depend on $\mathcal{D}_{XY}$ in more than simply the noise rate.

Chapter 2

Rates of Convergence in Active Learning

In this chapter, we study the rates of convergence in generalization error achievable by active learning under various types of label noise. Additionally, we study the more general problem of active learning with a nested hierarchy of hypothesis classes, and propose an algorithm whose error rate provably converges to the best achievable error among classifiers in the hierarchy at a rate adaptive to both the complexity of the optimal classifier and the noise conditions. In particular, we state sufficient conditions for these rates to be dramatically faster than those achievable by passive learning.

2.1   Introduction

There have recently been a series of exciting advances on the topic of active learning with arbitrary classification noise (the so-called *agnostic* PAC model), resulting in several new algorithms capable of achieving improved convergence rates compared to passive learning under certain conditions. The first, proposed by Balcan, Beygelzimer, and Langford [2006] was the A^2 (agnostic active) algorithm, which is provably never significantly worse than passive learning by empirical risk minimization. This algorithm was later analyzed in more detail in [Hanneke, 2007b], where it was found that a complexity measure called the *disagreement*

coefficient characterizes the worst-case convergence rates achieved by A^2 for any given hypothesis class, data distribution, and best achievable error rate in the class. The next major advance was by Dasgupta, Hsu, and Monteleoni [2007], who proposed a new algorithm, and proved that it improves the dependence of the convergence rates on the disagreement coefficient compared to A^2. Both algorithms are defined below in Section 2.2. While all of these advances are encouraging, they are limited in two ways. First, the convergence rates that have been proven for these algorithms typically only improve the dependence on the magnitude of the noise (more precisely, the noise rate of the hypothesis class), compared to passive learning. Thus, in an asymptotic sense, for nonzero noise rates these results represent at best a constant factor improvement over passive learning. Second, these results are limited to learning with a fixed hypothesis class of limited expressiveness, so that convergence to the Bayes error rate is not always a possibility.

On the first of these limitations, some recent work by Castro and Nowak [2006] on learning threshold classifiers discovered that if certain parameters of the noise distribution are *known* (namely, parameters related to Tsybakov's margin conditions), then we can achieve strict improvements in the asymptotic convergence rate via a specific active learning algorithm designed to take advantage of that knowledge for thresholds. That work left open the question of whether such improvements could be achieved by an algorithm that does not explicitly depend on the noise conditions (i.e., in the *agnostic* setting), and whether this type of improvement is achievable for more general families of hypothesis classes. In a personal communication, John Langford and Rui Castro claimed such improvements are achieved by A^2 for the special case of threshold classifiers. However, there remained an open question of whether such rate improvements could be generalized to hold for arbitrary hypothesis classes. In Section 2.3, we provide this generalization. We analyze the rates achieved by A^2 under Tsybakov's noise conditions [Mammen and Tsybakov, 1999, Tsybakov, 2004]; in particular, we find that these rates are strictly superior to the known rates for passive learning, when the disagreement coefficient is small. We also study a novel modification of the algorithm of Dasgupta, Hsu, and Monteleoni

[2007], proving that it improves upon the rates of A^2 in its dependence on the disagreement coefficient.

Additionally, in Section 2.4, we address the second limitation by proposing a general model selection procedure for active learning with an arbitrary structure of nested hypothesis classes. If the classes each have finite complexity, the error rate for this algorithm converges to the best achievable error by any classifier in the structure, at a rate that adapts to the noise conditions and complexity of the optimal classifier. In general, if the structure is constructed to include arbitrarily good approximations to any classifier, the error converges to the Bayes error rate in the limit. In particular, if the Bayes optimal classifier is in some class within the structure, the algorithm performs nearly as well as running an agnostic active learning algorithm on that single hypothesis class, thus preserving the convergence rate improvements achievable for that class.

2.1.1 Tsybakov's Noise Conditions

In this chapter, we will primarily be interested in the sets $\mathcal{T}sybakov(\mathbb{C}, \kappa, \mu)$, for parameter values $\mu > 0$ and $\kappa \geq 1$. These noise conditions have recently received substantial attention in the passive learning literature, as they describe situations in which the asymptotic minimax convergence rate of passive learning is faster than the worst case $n^{-1/2}$ rate [e.g., Koltchinskii, 2006, Mammen and Tsybakov, 1999, Massart and Élodie Nédélec, 2006, Tsybakov, 2004].

This condition is satisfied when, for example,

$$\exists \mu' > 0, \kappa \geq 1 \text{ s.t. } \exists h \in \mathbb{C} : \forall h' \in \mathbb{C}, er(h') - \nu \geq \mu' \mathbb{P}\{h(X) \neq h'(X)\}^\kappa.$$

As we will see, the case where $\kappa = 1$ is particularly interesting; for instance, this is the case when $h^* \in \mathbb{C}$ and $\mathbb{P}\{|\eta(X) - 1/2| > c\} = 1$ for some constant $c \in (0, 1/2)$. Informally, in many cases these conditions can often be interpreted in terms of the relation between magnitude of noise and distance to the decision boundary; that is, since in practice the amount of noise in an example's label is often inversely related to the distance from the decision boundary, a κ value of 1 may often result from having low density near the decision boundary (i.e., large

15

margin); when this is not the case, the value of κ is essentially determined by how quickly $\eta(x)$ changes as x approaches the decision boundary. See [Castro and Nowak, 2006, Koltchinskii, 2006, Mammen and Tsybakov, 1999, Massart and Élodie Nédélec, 2006, Tsybakov, 2004] for further interpretations of this margin condition.

It is known that when these conditions are satisfied for some $\kappa \geq 1$ and $\mu > 0$, the passive learning method of empirical risk minimization achieves a convergence rate guarantee, holding with probability $\geq 1 - \delta$, of

$$er(\arg\min_{h \in \mathbb{C}} er_n(h)) - \nu \leq c \left(\frac{d \log(n/\delta)}{n} \right)^{\frac{\kappa}{2\kappa-1}},$$

where c is a (κ and μ -dependent) constant [Koltchinskii, 2006, Mammen and Tsybakov, 1999, Massart and Élodie Nédélec, 2006]. Furthermore, for some hypothesis classes, this is known to be a tight bound (up to the log factor) on the minimax convergence rate, so that there is *no* passive learning algorithm for these classes for which we can guarantee a faster convergence rate, given that the guarantee depends on $\mathcal{D}_{XY}$ only through μ and κ [Tsybakov, 2004].

2.1.2 Disagreement Coefficient

Central to the idea of Algorithm 0, and the various generalizations there-of we will study, is the idea of the *region of disagreement* of the version space. Thus, a quantification of the performance of these algorithms should hinge upon a description of how quickly the region of disagreement collapses as the algorithm processes examples. This rate of collapse is precisely captured by a notion introduced in [Hanneke, 2007b], called the *disagreement coefficient*. It is a measure of the complexity of an active learning problem, which has proven quite useful for analyzing the convergence rates of certain types of active learning algorithms: for example, the algorithms of Balcan, Beygelzimer, and Langford [2006], Beygelzimer, Dasgupta, and Langford [2009], Cohn, Atlas, and Ladner [1994], Dasgupta, Hsu, and Monteleoni [2007]. Informally, it quantifies how much disagreement there is among a set of classifiers relative to how close to

some h they are. The following is a version of its definition, which we will use extensively below.

Definition 2.1. *The disagreement coefficient of h with respect to $\mathbb{C}$ under $\mathcal{D}_{XY}[\mathcal{X}]$ is*

$$\theta_h = \sup_{r > r_0} \frac{\mathbb{P}(DIS(B(h,r)))}{r},$$

where r_0 can either be defined as 0, giving a coarse analysis, or for a more subtle analysis we can take it to be a function of n, the number of labels (see Section 2.7.1 for such a definition valid for the main theorems of this chapter: 2.11-2.15).

We further define the disagreement coefficient for the hypothesis class $\mathbb{C}$ with respect to the target distribution $\mathcal{D}_{XY}$ as $\theta = \limsup_{k \to \infty} \theta_{h^{(k)}}$, where $\{h^{(k)}\}$ is any sequence of $h^{(k)} \in \mathbb{C}$ with $er(h^{(k)})$ monotonically decreasing to ν.

In particular, we can always bound the disagreement coefficient by $\sup_{h \in \mathbb{C}} \theta_h \geq \theta$.

Because of its simple intuitive interpretation, measuring the amount of disagreement in a local neighborhood of some classifier h, the disagreement coefficient has the wonderful property of being relatively simple to calculate for a wide range of learning problems, especially when those problems have some type of geometric representation. To illustrate this, we will go through a few simple examples, taken from [Hanneke, 2007b].

Consider the hypothesis class of thresholds h_z on the interval $[0,1]$ (for $z \in [0,1]$), where $h_z(x) = +1$ iff $x \geq z$. Furthermore, suppose $\mathcal{D}_{XY}[\mathcal{X}]$ is uniform on $[0,1]$. In this case, it is clear that the disagreement coefficient is at most 2, since the region of disagreement of $B(h_z, r)$ is roughly $\{x \in [0,1] : |x - z| \leq r\}$. That is, since the disagreement region grows at rate 1 in two disjoint directions as r increases, the disagreement coefficient $\theta_{h_z} = 2$ for any $z \in (0,1)$.

As a second example, consider the disagreement coefficient for *intervals* on $[0,1]$. As before, let $\mathcal{X} = [0,1]$ and $\mathcal{D}_{XY}[\mathcal{X}]$ be uniform, but this time $\mathbb{C}$ is the set of intervals $I_{[a,b]}$ such that for $x \in [0,1]$, $I_{[a,b]}(x) = +1$ iff $x \in [a,b]$ (for $a,b \in [0,1]$, $a \leq b$). In contrast to thresholds, the disagreement coefficients θ_h for the space of intervals vary widely depending on the particular h. In particular, take any $h = I_{[a,b]}$ where $0 < a \leq b < 1$. In this case, $\theta_h \leq \max\left\{\frac{1}{\max\{r_0, b-a\}}, 4\right\}$.

17

To see this, note that when $r_0 < r < b - a$, every interval in $B(I_{[a,b]}, r)$ has its lower and upper boundaries within r of a and b, respectively; thus, $\mathbb{P}(DIS(B(I_{[a,b]}, r))) \leq 4r$. However, when $r \geq \max\{r_0, b - a\}$, *every* interval of width $\leq r - (b - a)$ is in $B(I_{[a,b]}, r)$, so $\mathbb{P}(DIS(B(I_{[a,b]}, r))) = 1$.

As a slightly more involved example, consider the following theorem.

Theorem 2.2. *[Hanneke, 2007b] If $\mathcal{X}$ is the surface of the origin-centered unit sphere in $\mathbb{R}^d$ for $d > 2$, $\mathbb{C}$ is the space of linear separators whose decision surface passes through the origin, and $\mathcal{D}_{XY}[\mathcal{X}]$ is the uniform distribution on $\mathcal{X}$, then $\forall h \in \mathbb{C}$ the disagreement coefficient θ_h satisfies*

$$\frac{1}{4} \min \left\{ \pi \sqrt{d}, \frac{1}{r_0} \right\} \leq \theta_h \leq \min \left\{ \pi \sqrt{d}, \frac{1}{r_0} \right\}.$$

Proof. First we represent the concepts in $\mathbb{C}$ as weight vectors $w \in \mathbb{R}^d$ in the usual way. For $w_1, w_2 \in \mathbb{C}$, by examining the projection of $\mathcal{D}_{XY}[\mathcal{X}]$ onto the subspace spanned by $\{w_1, w_2\}$, we see that $\mathbb{P}(x : sign(w_1 \cdot x) \neq sign(w_2 \cdot x)) = \frac{arccos(w_1 \cdot w_2)}{\pi}$. Thus, for any $w \in \mathbb{C}$ and $r \leq 1/2$, $B(w, r) = \{w' : w \cdot w' \geq cos(\pi r)\}$. Since the decision boundary corresponding to w' is orthogonal to the vector w', some simple trigonometry gives us that

$$DIS(B(w, r)) = \{x \in \mathcal{X} : |x \cdot w| \leq sin(\pi r)\}.$$

Letting $A(d, R) = \frac{2\pi^{d/2} R^{d-1}}{\Gamma\left(\frac{d}{2}\right)}$ denote the surface area of the radius-R sphere in $\mathbb{R}^d$, we can express the disagreement rate at radius r as

$$\mathbb{P}(DIS(B(w, r)))$$

$$= \frac{1}{A(d, 1)} \int_{-sin(\pi r)}^{sin(\pi r)} A\left(d - 1, \sqrt{1 - x^2}\right) dx = \frac{\Gamma\left(\frac{d}{2}\right)}{\sqrt{\pi} \Gamma\left(\frac{d-1}{2}\right)} \int_{-sin(\pi r)}^{sin(\pi r)} \left(1 - x^2\right)^{\frac{d-2}{2}} dx \qquad (*)$$

$$\leq \frac{\Gamma\left(\frac{d}{2}\right)}{\sqrt{\pi} \Gamma\left(\frac{d-1}{2}\right)} 2 sin(\pi r) \leq \sqrt{d - 2} sin(\pi r) \leq \sqrt{d} \pi r.$$

For the lower bound, note that $\mathbb{P}(DIS(B(w, 1/2))) = 1$ so $\theta_w \geq \min\left\{2, \frac{1}{r_0}\right\}$, and thus we need

18

only consider $r_0 < \frac{1}{8}$. Supposing $r_0 < r < \frac{1}{8}$, note that $(*)$ is at least

$$\sqrt{\frac{d}{12}} \int_{-sin(\pi r)}^{sin(\pi r)} \left(1 - x^2\right)^{\frac{d}{2}} dx \geq \sqrt{\frac{\pi}{12}} \int_{-sin(\pi r)}^{sin(\pi r)} \sqrt{\frac{d}{\pi}} e^{-d \cdot x^2} dx$$

$$\geq \frac{1}{2} \min \left\{ \frac{1}{2}, \sqrt{d} sin(\pi r) \right\} \geq \frac{1}{4} \min \left\{ 1, \pi \sqrt{d} r \right\}.$$

$\square$

The disagreement coefficient has many interesting properties that can help to bound its value for a given hypothesis class and distribution. We list a few elementary properties below. Their proofs, which are quite short and follow directly from the definition, are left as easy exercises.

Lemma 2.3. *[Close Marginals][Hanneke, 2007b] Suppose $\exists \lambda \in (0, 1]$ s.t. for any measurable set $A \subseteq \mathcal{X}$, $\lambda \mathbb{P}_{\mathcal{D}_X}(A) \leq \mathbb{P}_{\mathcal{D}'_X}(A) \leq \frac{1}{\lambda} \mathbb{P}_{\mathcal{D}_X}(A)$. Let $h : \mathcal{X} \to \{-1, 1\}$ be a measurable classifier, and suppose θ_h and θ'_h are the disagreement coefficients for h with respect to $\mathbb{C}$ under $\mathcal{D}_X$ and $\mathcal{D}'_X$ respectively. Then*

$$\lambda^2 \theta_h \leq \theta'_h \leq \frac{1}{\lambda^2} \theta_h.$$

Lemma 2.4. *[Finite Mixtures] Suppose $\exists \alpha \in [0, 1]$ s.t. for any measurable set $A \subseteq \mathcal{X}$, $\mathbb{P}_{\mathcal{D}_X}(A) = \alpha \mathbb{P}_{\mathcal{D}_1}(A) + (1 - \alpha) \mathbb{P}_{\mathcal{D}_2}(A)$. For a measurable $h : \mathcal{X} \to \{-1, 1\}$, let $\theta_h^{(1)}$ be the disagreement coefficient with respect to $\mathbb{C}$ under $\mathcal{D}_1$, $\theta_h^{(2)}$ be the disagreement coefficient with respect to $\mathbb{C}$ under $\mathcal{D}_2$, and θ_h be the disagreement coefficient with respect to $\mathbb{C}$ under $\mathcal{D}_X$. Then*

$$\theta_h \leq \theta_h^{(1)} + \theta_h^{(2)}.$$

Lemma 2.5. *[Finite Unions] Suppose $h \in \mathbb{C}_1 \cap \mathbb{C}_2$ is a classifier s.t. the disagreement coefficient with respect to $\mathbb{C}_1$ under $\mathcal{D}_X$ is $\theta_h^{(1)}$ and with respect to $\mathbb{C}_2$ under $\mathcal{D}_X$ is $\theta_h^{(2)}$. Then if θ_h is the disagreement coefficient with respect to $\mathbb{C} = \mathbb{C}_1 \cup \mathbb{C}_2$ under $\mathcal{D}_X$, we have that*

$$\max \left\{ \theta_h^{(1)}, \theta_h^{(2)} \right\} \leq \theta_h \leq \theta_h^{(1)} + \theta_h^{(2)}.$$

The disagreement coefficient has deep connections to several other quantities, such as doubling dimension [Li and Long, 2007] and VC dimension [Vapnik, 1982]. See [Hanneke, 2007b],

[Dasgupta, Hsu, and Monteleoni, 2007], [Balcan, Hanneke, and Wortman, 2008], and [Beygelzimer, Dasgupta, and Langford, 2009] for further discussions of various uses of the disagreement coefficient and related notions and extensions in active learning. In particular, Beygelzimer, Dasgupta, and Langford [2009] present an interesting analysis using a natural extension of the disagreement coefficient to study active learning with a larger family of loss functions beyond $0-1$ loss. As a related aside, although the focus of this thesis is active learning, interestingly the disagreement coefficient also has applications in the analysis of *passive* learning; see Section 2.9 for an interesting example of this.

2.2 General Algorithms

The algorithms described below for the problem of active learning with label noise each represent noise-robust variants of Algorithm 0. They work to reduce the set of candidate hypotheses, while only requesting the labels of examples in the region of disagreement of these candidates. The trick is to only remove a classifier from the candidate set once we have high statistical confidence that it is worse than some other candidate classifier so that we never remove the best classifier. However, the two algorithms differ somewhat in the details of how that confidence is calculated.

2.2.1 Algorithm 1

The first algorithm, originally proposed by Balcan, Beygelzimer, and Langford [2006], is typically referred to as A^2 for *Agnostic Active*. This was historically the first general-purpose agnostic active learning algorithm shown to achieve improved error guarantees for certain learning problems in certain ranges of n and ν. A version of the algorithm is described below.

> **Algorithm 1**
> Input: hypothesis class $\mathbb{C}$, label budget n, confidence δ
> Output: classifier $\hat{h}$
>
> ---
>
> 0. $V \leftarrow \mathbb{C}, R \leftarrow DIS(\mathbb{C}), Q \leftarrow \emptyset, m \leftarrow 0$
> 1. For $t = 1, 2, \ldots, n$
> 2.   If $\mathbb{P}(DIS(V)) \leq \frac{1}{2}\mathbb{P}(R)$
> 3. $R \leftarrow DIS(V); Q \leftarrow \emptyset$
> 4.   If $\mathbb{P}(R) \leq 2^{-n}$, Return any $h \in V$
> 5. $m \leftarrow \min\{m' > m : X_{m'} \in R\}$
> 6.   Request Y_m and let $Q \leftarrow Q \cup \{(X_m, Y_m)\}$
> 7. $V \leftarrow \{h \in V : LB(h, Q, \delta/n) \leq \min_{h' \in V} UB(h', Q, \delta/n)\}$
> 8. $h_t \leftarrow \arg\min_{h \in V} UB(h, Q, \delta/n)$
> 9. $\beta_t \leftarrow (UB(h_t, Q, \delta/n) - \min_{h \in V} LB(h, Q, \delta/n))\mathbb{P}(R)$
> 10. Return $\hat{h}_n = h_{\hat{t}}$, where $\hat{t} = \operatorname*{argmin}_{t \in \{1,2,\ldots,n\}} \beta_t$

Algorithm 1 is defined in terms of two functions: UB and LB. These represent upper and lower confidence bounds on the error rate of a classifier from $\mathbb{C}$ with respect to an arbitrary sampling distribution, as a function of a labeled sequence sampled according to that distribution. As long as these bounds satisfy

$$\mathbb{P}_{Z \sim \mathcal{D}^m}\{\forall h \in \mathbb{C}, LB(h, Z, \delta) \leq er_{\mathcal{D}}(h) \leq UB(h, Z, \delta)\} \geq 1 - \delta$$

for any distribution $\mathcal{D}$ over $\mathcal{X} \times \{-1, 1\}$ and any $\delta \in (0, 1/2)$, and UB and LB converge to each other as m grows, this algorithm is known to be correct, in that $er(\hat{h}) - \nu$ converges to 0 in probability [Balcan, Beygelzimer, and Langford, 2006]. For instance, Balcan, Beygelzimer, and Langford suggest defining these functions based on classic results on uniform convergence rates in passive learning [Vapnik, 1982], such as

$$UB(h, Q, \delta) = \min\{er_Q(h) + G(|Q|, \delta), 1\}, \quad LB(h, Q, \delta) = \max\{er_Q(h) - G(|Q|, \delta), 0\},$$

$$(2.1)$$

where $G(m, \delta) = \frac{1}{m} + \sqrt{\frac{\ln\frac{4}{\delta} + d\ln\frac{2em}{d}}{m}}$, and by convention $G(0, \delta) = \infty$. This choice is justified by the following lemma, due to Vapnik [1998].

21

Lemma 2.6. *For any distribution $\mathcal{D}$ over $\mathcal{X} \times \{-1, 1\}$, and any $\delta > 0$ and $m \in \mathbb{N}$, with probability $\geq 1 - \delta$ over the draw of $Z \sim \mathcal{D}^m$, every $h \in \mathbb{C}$ satisfies*

$$|er_Z(h) - er_{\mathcal{D}}(h)| \leq G(m, \delta). \tag{2.2}$$

To avoid computational issues, instead of explicitly representing the sets V and R, we may implicitly represent it as a set of constraints imposed by the condition in Step 7 of previous iterations. We may also replace $\mathbb{P}(DIS(V))$ and $\mathbb{P}(R)$ by estimates, since these quantities can be estimated to arbitrary precision with arbitrarily high confidence using only *unlabeled* examples.

2.2.2 Algorithm 2

The second algorithm we study was originally proposed by Dasgupta, Hsu, and Monteleoni [2007]. It uses a type of constrained passive learning subroutine, LEARN, defined as follows.

$$\text{LEARN}_{\mathbb{C}}(\mathcal{L}, Q) = \underset{h \in \mathbb{C}: er_{\mathcal{L}}(h) = 0}{\operatorname{argmin}} er_Q(h).$$

By convention, if no $h \in \mathbb{C}$ has $er_{\mathcal{L}}(h) = 0$, $\text{LEARN}_{\mathbb{C}}(\mathcal{L}, Q) = \varnothing$.

Algorithm 2
Input: hypothesis class $\mathbb{C}$, label budget n, confidence δ
Output: classifier $\hat{h}$, set of labeled examples $\mathcal{L}$, set of labeled examples Q

---

0. $\mathcal{L} \leftarrow \emptyset$, $Q \leftarrow \emptyset$
1. For $m = 1, 2, \ldots$
2.   If $|Q| = n$ or $|\mathcal{L}| = 2^n$, Return $\hat{h} = \text{LEARN}_{\mathbb{C}}(\mathcal{L}, Q)$ along with $\mathcal{L}$ and Q
3.   For each $y \in \{-1, +1\}$, let $h^{(y)} = \text{LEARN}_{\mathbb{C}}(\mathcal{L} \cup \{(X_m, y)\}, Q)$
4.   If some y has $h^{(-y)} = \varnothing$ or
        $er_{\mathcal{L} \cup Q}(h^{(-y)}) - er_{\mathcal{L} \cup Q}(h^{(y)}) > \Delta_{m-1}(\mathcal{L}, Q, h^{(y)}, h^{(-y)}, \delta)$
5.   Then $\mathcal{L} \leftarrow \mathcal{L} \cup \{(X_m, y)\}$
6.   Else Request the label Y_m and let $Q \leftarrow Q \cup \{(X_m, Y_m)\}$

Algorithm 2 is defined in terms of a function $\Delta_m(\mathcal{L}, Q, h^{(y)}, h^{(-y)}, \delta)$, representing a threshold for a type of hypothesis test. This threshold must be set carefully, since the set $\mathcal{L} \cup Q$ is not actually an i.i.d. sample from $\mathcal{D}_{XY}$. Dasgupta, Hsu, and Monteleoni [2007] suggest defining this function as

$$\Delta_m(\mathcal{L}, Q, h^{(y)}, h^{(-y)}, \delta) = \beta_m^2 + \beta_m \left(\sqrt{er_{\mathcal{L} \cup Q}(h^{(y)})} + \sqrt{er_{\mathcal{L} \cup Q}(h^{(-y)})} \right), \tag{2.3}$$

where $\beta_m = \sqrt{\frac{4\ln(8m(m+1)\mathbb{C}[2m]^2/\delta)}{m}}$ and $\mathbb{C}[2m]$ is the shatter coefficient [e.g., Devroye et al., 1996]; this suggestion is based on a confidence bound they derive, and they prove the correctness of the algorithm with this definition. For now we will focus on the first return value (the classifier), leaving the others for Section 2.4, where they will be useful for chaining multiple executions together.

2.3 Convergence Rates

In both of the above cases, one can prove fallback guarantees stating that neither algorithm is significantly worse than the minimax rates for passive learning [Balcan, Beygelzimer, and Langford, 2006, Dasgupta, Hsu, and Monteleoni, 2007]. However, it is even more interesting to discuss situations in which one can prove error rate guarantees for these algorithms significantly *better* than those achievable by passive learning. In this section, we begin by reviewing known results on these potential improvements, stated in terms of the disagreement coefficient; we then proceed to discuss new results for Algorithm 1 and a novel variant of Algorithm 2, and describe the convergence rates achieved by these methods in terms of the disagreement coefficient and Tsybakov's noise conditions.

2.3.1 The Disagreement Coefficient and Active Learning: Basic Results

Before going into the results for general distributions $\mathcal{D}_{XY}$ on $\mathcal{X} \times \{-1, +1\}$, it will be instructive to first look at the special case when the noise rate is zero. Understanding how the disagreement coefficient enters into the analysis of this simpler case may aid in digestion of the theorems and proofs for the general case presented later, where it plays an essentially analogous role. Most of the major ingredients of the proofs for the general case can be found in this special case, albeit in a much simpler form. Although this result has not previously been published, the proof is essentially similar to (one case of) the analysis of Algorithm 1 in [Hanneke, 2007b].

> **Theorem 2.7.** *Suppose $\mathcal{D}_{XY} \in \mathcal{R}ealizable(\mathbb{C})$ for a VC class $\mathbb{C}$, and let $f \in \mathbb{C}$ be such that $er(f) = 0$, and $\theta_f < \infty$. For any $n \in \mathbb{N}$, with probability $\geq 1 - \delta$ over the draw of the unlabeled examples, the classifier h_n returned by Algorithm 0 after n label requests satisfies*
>
> $$er(h_n) \leq 2 \cdot exp\left\{ -\frac{n}{6\theta_f(4d\ln(44\theta_f) + \ln(2n/\delta))} \right\}.$$

Proof. The case $diam(\mathbb{C}) = 0$ is trivial, so assume $diam(\mathbb{C}) > 0$ (and thus $d \geq 1$ and $\theta_f > 0$). Let V_t denote the set of classifiers in $\mathbb{C}$ consistent with the first t label requests. If $\mathbb{P}(DIS(V_t)) = 0$ for some $t \leq n$, then the result holds trivially. Otherwise, with probability 1, the algorithm uses all n label requests; in this case, consider some $t < n$. Let x_{m_t} denote the example corresponding to the tth label request. Let $\lambda_n = 4\theta_f(4d\ln(16e\theta_f) + \ln(2n/\delta))$, $t' = t + \lambda_n$, and let $x_{m_{t'}}$ denote the example corresponding to label request number t' (assuming $t \leq n - \lambda_n$). In particular, this implies $|\{x_{m_t+1}, x_{m_t+2}, \ldots, x_{m_{t'}}\} \cap DIS(V_t)| \geq \lambda_n$, which means there is an i.i.d. sample of size λ_n from $\mathcal{D}_{XY}[\mathcal{X}]$ given $X \in DIS(V_t)$ contained in $\{x_{m_t+1}, x_{m_t+2}, \ldots, x_{m_{t'}}\}$: namely, the first λ_n points in this subsequence that are in $DIS(V_t)$.

Now recall that, by classic results from the passive learning literature [e.g., Blumer et al., 1989, Vapnik, 1982], this implies that on an event $E_{\delta,t}$ holding with probability $1 - \delta/n$,

$$\sup_{h \in V_{t'}} er(h|DIS(V_t)) \leq \frac{4d\ln\frac{2e\lambda_n}{d} + \ln\frac{2n}{\delta}}{\lambda_n} \leq 1/(2\theta_f).$$

Since $V_{t'} \subseteq V_t$, this means

$$\mathbb{P}(DIS(V_{t'})) \leq \mathbb{P}(DIS(B(f, \mathbb{P}(DIS(V_t))/(2\theta_f)))) \leq \mathbb{P}(DIS(V_t))/2.$$

By a union bound, the events $E_{\delta,t}$ hold for all $t \in \{i\lambda_n : i \in \{0, 1, \ldots, \lfloor n/\lambda_n \rfloor - 1\}\}$ with probability $\geq 1 - \delta$. On these events, if $n \geq \lambda_n \lceil \log_2(1/\epsilon) \rceil$, then (by induction)

$$\sup_{h \in V_n} er(h) \leq \mathbb{P}(DIS(V_n)) \leq \epsilon.$$

Solving for ϵ in terms of n gives the result. $\qquad\square$

2.3.2 Known Results on Convergence Rates for Agnostic Active Learning

We will now describe the known results for agnostic active learning algorithms, starting with Algorithm 1. The key to the potential convergence rate improvements of Algorithm 1 is that, as the region of disagreement R decreases in measure, the magnitude of the error difference $er(h|R) - er(h'|R)$ of any classifiers $h, h' \in V$ under the *conditional* sampling distribution (given R) can become significantly larger (by a factor of $\mathbb{P}(R)^{-1}$) than $er(h) - er(h')$, making it significantly easier to determine which of the two is worse using a sample of labeled examples. In particular, [Hanneke, 2007b] developed a technique for analyzing this type of algorithm, resulting in the following convergence rate guarantee for Algorithm 1. The proof follows similar reasoning to what we will see in the next subsection, but is omitted here to reduce redundancy; see [Hanneke, 2007b] for the full details.

Theorem 2.8. *[Hanneke, 2007b] Let $\hat{h}_n$ be the classifier returned by Algorithm 1 when allowed n label requests, using the bounds* (2.1) *and confidence parameter $\delta \in (0, 1/2)$. Then there exists a finite universal constant c such that, with probability $\geq 1 - \delta$, $\forall n \in \mathbb{N}$,*

$$er(\hat{h}_n) - \nu \leq c\sqrt{\frac{\nu^2\theta^2 d \log\frac{1}{\delta}}{n}} \log \frac{n}{\nu^2\theta^2 d \log\frac{1}{\delta}} + \frac{1}{\delta}exp\left\{-\sqrt{\frac{n}{c\theta^2 d}}\right\}.$$

Similarly, the key to improvements from Algorithm 2 is that as m increases, we only need to request the labels of those examples in the region of disagreement of the set of classifiers with near-optimal empirical error rates. Thus, if $\mathbb{P}(DIS(\mathbb{C}(\epsilon)))$ shrinks as ϵ decreases, we expect the frequency of label requests to shrink as m increases. Since we are careful not to discard the best classifier, and the excess error rate of a classifier can be bounded in terms of the Δ_m function, we end up with a bound on the excess error which is converging in m, the number of *unlabeled* examples processed, even though we request a number of labels growing slower than m. When this situation occurs, we expect Algorithm 2 will provide an improved convergence rate compared to passive learning. Using the disagreement coefficient, Dasgupta, Hsu, and Monteleoni [2007] prove the following convergence rate guarantee.

Theorem 2.9. *[Dasgupta, Hsu, and Monteleoni, 2007] Let $\hat{h}_n$ be the classifier returned by Algorithm 2 when allowed n label requests, using the threshold (2.3), and confidence parameter $\delta \in (0, 1/2)$. Then there exists a finite universal constant c such that, with probability $\geq 1 - \delta$, $\forall n \in \mathbb{N}$,*

$$er(\hat{h}_n) - \nu \leq c\sqrt{\frac{\nu^2\theta d \log\frac{1}{\delta}\log\frac{n}{\theta\nu\delta}}{n}} + \sqrt{d\log\frac{1}{\delta}} \cdot exp\left\{-\sqrt{\frac{n}{c\theta d \log^2\frac{1}{\delta}}}\right\}.$$

Note that, among other changes, this bound improves the dependence on the disagreement coefficient, θ, compared to the bound for Algorithm 1. In both cases, for certain ranges of θ, ν, and n, these bounds can represent significant improvements in the excess error guarantees, compared to the corresponding guarantees possible for passive learning. However, in both cases, when $\nu > 0$ these bounds have an *asymptotic* dependence on n of $\tilde{\Theta}(n^{-1/2})$, which is no better than the convergence rates achievable by passive learning (e.g., by empirical risk minimization). Thus, there remains the question of whether either algorithm can achieve asymptotic convergence rates strictly superior to passive learning for distributions with nonzero noise rates. This is the topic we turn to next.

2.3.3 Adaptation to Tsybakov's Noise Conditions

It is known that for most nontrivial $\mathbb{C}$, for any n and $\nu > 0$, for every active learning algorithm there is some distribution with noise rate ν for which we can guarantee excess error no better than $\propto \nu n^{-1/2}$ [Kääriäinen, 2006]; that is, the $n^{-1/2}$ asymptotic dependence on n in the above bounds matches the corresponding minimax rate, and thus cannot be improved as long as the bounds depend on $\mathcal{D}_{XY}$ only via ν (and θ). Therefore, if we hope to discover situations in which these algorithms have strictly superior asymptotic dependence on n, we will need to allow the bounds to depend on a more detailed description of the noise distribution than simply the noise rate ν.

As previously mentioned, one way to describe a noise distribution using a more detailed

parameterization is to use Tsybakov's noise conditions ($\mathfrak{T}sybakov(\mathbb{C}, \kappa, \mu)$). In the context of passive learning, this allows one to describe situations in which the rate of convergence is between n^{-1} and $n^{-1/2}$, even when $\nu > 0$. This raises the natural question of how these active learning algorithms perform when the noise distribution satisfies this condition with finite μ and κ parameter values. In many ways, it seems active learning is particularly well-suited to exploit these more favorable noise conditions, since they imply that as we eliminate suboptimal classifiers, the diameter of the version space decreases; thus, for small θ values, the region of disagreement should also be decreasing, allowing us to focus the samples in a smaller region and accelerate the convergence.

Focusing on the special case of one-dimensional threshold classifiers under a uniform marginal distribution, Castro and Nowak [2006] studied conditions related to $\mathfrak{T}sybakov(\mathbb{C}, \kappa, \mu)$. In particular, they studied a threshold-learning algorithm that, unlike the algorithms described here, takes κ as *input*, and found its convergence rate to be $\propto \left(\frac{\log n}{n}\right)^{\frac{\kappa}{2\kappa-2}}$ when $\kappa > 1$, and $exp\{-cn\}$ for some (μ-dependent) constant c, when $\kappa = 1$. Note that this improves over the $n^{-\frac{\kappa}{2\kappa-1}}$ rates achievable in passive learning [Tsybakov, 2004]. Furthermore, they prove that a value $\propto n^{-\frac{\kappa}{2\kappa-2}}$ (or $exp\{-c'n\}$, for some c', when $\kappa = 1$) is also a *lower bound* on the minimax rate. Later, in a personal communication, Langford and Castro claimed that this near-optimal rate is also achieved by Algorithm 1 (up to log factors) for the same learning problem (one-dimensional threshold classifiers under a uniform marginal distribution), leading to speculation that perhaps these improvements are achievable in the general case as well (under conditions on the disagreement coefficient).

Other than the one-dimensional threshold learning problem, it was not previously known whether Algorithm 1 or Algorithm 2 generally achieves convergence rates that exhibit these types of improvements.

2.3.4 Adaptive Rates in Active Learning

The above observations open the question of whether these algorithms, or variants thereof, improve this asymptotic dependence on n. It turns out this is indeed possible. Specifically, we have the following result for Algorithm 1.

Theorem 2.10. *Let $\hat{h}_n$ be the classifier returned by Algorithm 1 when allowed n label requests, using the bounds (2.1) and confidence parameter $\delta \in (0, 1/2)$. Suppose further that $\mathcal{D}_{XY} \in \mathfrak{T}sybakov(\mathbb{C}, \kappa, \mu)$ for finite parameter values $\kappa \geq 1$ and $\mu > 0$ and VC class $\mathbb{C}$. Then there exists a finite (κ- and μ-dependent) constant c such that, for any $n \in \mathbb{N}$, with probability $\geq 1 - \delta$,*

$$er(\hat{h}_n) - \nu \leq \begin{cases} exp\left\{-\frac{n}{cd\theta^2 \log(n/\delta)}\right\}, & \text{when } \kappa = 1 \\ c\left(\frac{d\theta^2 \log^2(n/\delta)}{n}\right)^{\frac{\kappa}{2\kappa-2}}, & \text{when } \kappa > 1 \end{cases}.$$

Proof. The case of $diam(\mathbb{C}) = 0$ clearly holds, so we will focus on the nontrivial case of $diam(\mathbb{C}) > 0$ (and therefore, $\theta > 0$ and $d \geq 1$). We will proceed by bounding the *label complexity*, or size of the label budget n that is sufficient to guarantee, with high probability, that the excess error of the returned classifier will be at most ϵ (for arbitrary $\epsilon > 0$); with this in hand, we can simply bound the inverse of the function to get the result in terms of a bound on excess error.

First note that, by Lemma 2.6 and a union bound, on an event of probability $1 - \delta$, (2.2) holds with $\eta = \delta/n$ for every set Q, relative to the conditional distribution given its respective R set, for any value of n. For the remainder of this proof, we assume that this $1 - \delta$ probability event occurs. In particular, this means that for every $h \in \mathbb{C}$ and every Q set in the algorithm, $LB(h, Q, \delta/n) \leq er(h|R) \leq UB(h, Q, \delta/n)$, for the set R that Q is sampled under. Thus, we always have the invariant that at all times,

$$\forall \gamma > 0, \{h \in V : er(h) - \nu \leq \gamma\} \neq \emptyset, \tag{2.4}$$

and therefore also that $\forall t, er(h_t) - \nu = (er(h_t|R) - \inf_{h \in V} er(h|R))\mathbb{P}(R) \leq \beta_t$. We will spend

the remainder of the proof bounding the size of n sufficient to guarantee some $\beta_t \leq \epsilon$.

Recalling the definition of the $h^{(k)}$ sequence (from Definition 2.1), note that after step 7,

$$\left\{ h \in V : \limsup_k \mathbb{P}(h(X) \neq h^{(k)}(X)) > \frac{\mathbb{P}(R)}{2\theta} \right\}$$

$$= \left\{ h \in V : \left(\frac{\limsup_k \mathbb{P}(h(X) \neq h^{(k)}(X))}{\mu} \right)^{\kappa} > \left(\frac{\mathbb{P}(R)}{2\mu\theta} \right)^{\kappa} \right\}$$

$$\subseteq \left\{ h \in V : \left(\frac{diam(er(h) - \nu; \mathbb{C})}{\mu} \right)^{\kappa} > \left(\frac{\mathbb{P}(R)}{2\mu\theta} \right)^{\kappa} \right\}$$

$$\subseteq \left\{ h \in V : er(h) - \nu > \left(\frac{\mathbb{P}(R)}{2\mu\theta} \right)^{\kappa} \right\}$$

$$= \left\{ h \in V : er(h|R) - \inf_{h' \in V} er(h'|R) > \mathbb{P}(R)^{\kappa-1}(2\mu\theta)^{-\kappa} \right\}$$

$$\subseteq \left\{ h \in V : UB(h, Q, \delta/n) - \min_{h' \in V} LB(h', Q, \delta/n) > \mathbb{P}(R)^{\kappa-1}(2\mu\theta)^{-\kappa} \right\}$$

$$= \left\{ h \in V : LB(h, Q, \delta/n) - \min_{h' \in V} UB(h', Q, \delta/n) > \mathbb{P}(R)^{\kappa-1}(2\mu\theta)^{-\kappa} - 4G(|Q|, \delta/n) \right\}.$$

By definition, every $h \in V$ has $LB(h, Q, \delta/n) \leq \min_{h' \in V} UB(h', Q, \delta/n)$, so for this last set to be nonempty after step 7, we must have $\mathbb{P}(R)^{\kappa-1}(2\mu\theta)^{-\kappa} < 4G(|Q|, \delta/n)$. On the other hand, if $\left\{ h \in V : \limsup_k \mathbb{P}(h(X) \neq h^{(k)}(X)) > \frac{\mathbb{P}(R)}{2\theta} \right\} = \emptyset$, then

$$\mathbb{P}(DIS(V)) \leq \mathbb{P}(DIS(\{h \in \mathbb{C} : \limsup_k \mathbb{P}(h(X) \neq h^{(k)}(X)) \leq \mathbb{P}(R)/(2\theta)\}))$$

$$= \limsup_k \mathbb{P}(DIS(\{h \in \mathbb{C} : \mathbb{P}(h(X) \neq h^{(k)}(X)) \leq \mathbb{P}(R)/(2\theta)\})) \leq \limsup_k \theta_{h_k} \frac{\mathbb{P}(R)}{2\theta} = \frac{\mathbb{P}(R)}{2},$$

so that we will definitely satisfy the condition in step 2 on the next round. Since $|Q|$ gets reset to 0 upon reaching step 3, we have that after every execution of step 7, $\mathbb{P}(R)^{\kappa-1}(2\mu\theta)^{-\kappa} < 4G(|Q| - 1, \delta/n)$.

If $\mathbb{P}(R) \leq \frac{\epsilon}{2G(|Q|-1,\delta/n)} \leq \frac{\epsilon}{2G(|Q|,\delta/n)}$, then certainly $\beta_t \leq \epsilon$. So on any round for which $\beta_t > \epsilon$, we must have $\mathbb{P}(R) > \frac{\epsilon}{2G(|Q|-1,\delta/n)}$. Combined with the above observations, on any round for which $\beta_t > \epsilon$, $\left(\frac{\epsilon}{2G(|Q|-1,\delta/n)} \right)^{\kappa-1} (2\mu\theta)^{-\kappa} < 4G(|Q| - 1, \delta/n)$, which implies (by simple algebra)

$$|Q| \leq \left(\frac{1}{\epsilon} \right)^{\frac{2\kappa-2}{\kappa}} (6\mu\theta)^2 \left(\ln \frac{4}{\delta} + (d+1)\ln(n) \right) + 1.$$

29

Since we need to reach step 3 at most $\lceil \log(1/\epsilon) \rceil$ times before we are guaranteed some $\beta_t \leq \epsilon$ ($\mathbb{P}(R)$ is at least halved each time we reach step 3), any

$$n \geq 1 + \left(\left(\frac{1}{\epsilon} \right)^{\frac{2\kappa - 2}{\kappa}} (6\mu\theta)^2 \left(\ln \frac{4}{\delta} + (d+1)\ln(n) \right) + 1 \right) \log_2 \frac{2}{\epsilon} \tag{2.5}$$

suffices to guarantee some $\beta_t \leq \epsilon$. This implies the stated result by basic inequalities to bound the smallest value of ϵ satisfying (2.5) for a given value of n. $\qquad\square$

If the disagreement coefficient is relatively small, Theorem 2.10 can represent a significant improvement in convergence rate compared to passive learning, where we typically expect rates of order $n^{-\kappa/(2\kappa-1)}$ [Mammen and Tsybakov, 1999, Tsybakov, 2004]; this gap is especially notable when the disagreement coefficient and κ are small. In particular, the bound matches (up to log factors) the form of the minimax rate *lower bound* proven by Castro and Nowak [2006] for threshold classifiers (where $\theta = 2$). Note that, unlike the analysis of Castro and Nowak [2006], we do not require the algorithm to be given any extra information about the noise distribution, so that this result is somewhat stronger; it is also more general, as this bound applies to an arbitrary hypothesis class. In some sense, Theorem 2.10 is somewhat surprising, since the bounds UB and LB used to define the set V and the bounds β_t are not themselves adaptive to the noise conditions.

Note that, as before, n gets divided by θ^2 in the rates achieved by A^2. As before, it is not clear whether any modification to the definitions of UB and LB can reduce this exponent on θ from 2 to 1. As such, it is natural to investigate the rates achieved by Algorithm 2 under $\mathcal{T}sybakov(\mathbb{C}, \kappa, \mu)$; we know that it does improve the dependence on θ for the worst case rates over distributions with any given noise rate, so we might hope that it does the same for the rates over distributions with any given values of μ and κ. Unfortunately, we do not presently know whether the original definition of Algorithm 2 achieves this improvement. However, we now present a slight modification of the algorithm, and prove that it does indeed provide the desired improvement in dependence on θ, while maintaining the improvements in the asymptotic

dependence on n. Specifically, consider the following definition for the threshold in Algorithm 2.

$$\Delta_m(\mathcal{L}, Q, h^{(y)}, h^{(-y)}, \delta) = 3\hat{\mathcal{E}}_{\mathbb{C}}(\mathcal{L} \cup Q, \delta; \mathcal{L}), \tag{2.6}$$

where $\hat{\mathcal{E}}_{\mathbb{C}}(\cdot, \cdot; \cdot)$ is defined in Section 2.6, based on a notion of local Rademacher complexity studied by Koltchinskii [2006]. Unlike the previous definitions, these definitions are known to be adaptive to Tsybakov's noise conditions, so that we would expect them to be asymptotically tighter and therefore allow the algorithm to more aggressively prune the set of candidate hypotheses. Using these definitions, we have the following theorem; its proof is included in Section 2.7.

Theorem 2.11. *Suppose $\hat{h}_n$ is the classifier returned by Algorithm 2 with threshold as in* (2.6), *when allowed n label requests and given confidence parameter $\delta \in (0, 1/2)$. Suppose further that $\mathcal{D}_{XY} \in \mathcal{T}sybakov(\mathbb{C}, \kappa, \mu)$ for finite parameter values $\kappa \geq 1$ and $\mu > 0$ and VC class $\mathbb{C}$. Then there exists a finite (κ and μ-dependent) constant c such that, with probability $\geq 1 - \delta$, $\forall n \in \mathbb{N}$,*

$$er(\hat{h}_n) - \nu \leq \begin{cases} \frac{1}{\delta} \cdot exp\left\{ -\sqrt{\frac{n}{cd\theta \log^3(d/\delta)}} \right\}, & \text{when } \kappa = 1 \\ c\left(\frac{d\theta \log^2(dn/\delta)}{n} \right)^{\frac{\kappa}{2\kappa-2}}, & \text{when } \kappa > 1 \end{cases}.$$

Note that this does indeed improve the dependence on θ, reducing its exponent from 2 to 1; we do lose some in that there is now a square root in the exponent of the $\kappa = 1$ case, but it is likely that an improved definition of $\hat{\mathcal{E}}$ and a refined analysis can correct this. The bound in Theorem 2.11 is stated in terms of the VC dimension d. However, for certain nonparametric function classes, it is sometimes preferable to quantify the complexity of the class in terms of a constraint on the *entropy* (with bracketing) of the class $\mathcal{E}ntropy_{[]}(\mathbb{C}, \alpha, \rho)$ [see e.g., Castro and Nowak, 2007, Koltchinskii, 2006, Tsybakov, 2004, van der Vaart and Wellner, 1996].

In passive learning, it is known that empirical risk minimization achieves a rate of order $n^{-\kappa/(2\kappa+\rho-1)}$, under $\mathcal{E}ntropy_{[]}(\mathbb{C}, \alpha, \rho) \cap \mathcal{T}sybakov(\mathbb{C}, \kappa, \mu)$, and that this is sometimes tight [Koltchinskii, 2006, Tsybakov, 2004]. The following theorem gives a bound on the rate of convergence of the same version of Algorithm 2 as in Theorem 2.11, this time in terms of the entropy

with bracketing condition which, as before, is faster than the passive learning rate when the disagreement coefficient is small. The proof of this is included in Section 2.7.

Theorem 2.12. *Suppose $\hat{h}_n$ is the classifier returned by Algorithm 2 with threshold as in (2.6), when allowed n label requests and given confidence parameter $\delta \in (0, 1/2)$. Suppose further that $\mathcal{D}_{XY} \in \mathcal{E}ntropy_{[]}(\mathbb{C}, \alpha, \rho) \cap \mathcal{T}sybakov(\mathbb{C}, \kappa, \mu)$ for finite parameter values $\kappa \geq 1$, $\mu > 0$, $\alpha > 0$, and $\rho \in (0, 1)$. Then there exists a finite (κ, μ, α and ρ -dependent) constant c such that, with probability $\geq 1 - \delta$, $\forall n \in \mathbb{N}$,*

$$er(\hat{h}_n) - \nu \leq c \left(\frac{\theta \log^2(n/\delta)}{n} \right)^{\frac{\kappa}{2\kappa + \rho - 2}}.$$

Although this result is stated for Algorithm 2, it is conceivable that, by modifying Algorithm 1 to use definitions of V and β_t based on $\hat{\mathcal{E}}_{\mathbb{C}}(Q, \delta; \emptyset)$, an analogous result may be possible for Algorithm 1 as well.

2.4 Model Selection

While the previous sections address adaptation to the noise distribution, they are still restrictive in that they deal only with finite complexity hypothesis classes, where it is often unrealistic to expect convergence to the Bayes error rate to be achievable. We address this issue in this section by developing a general algorithm for learning with a sequence of nested hypothesis classes of increasing complexity, similar to the setting of Structural Risk Minimization in passive learning [Vapnik, 1982]. The starting point for this discussion is the assumption of a structure on $\mathbb{C}$, in the form of a sequence of nested hypothesis classes.

$$\mathbb{C}_1 \subset \mathbb{C}_2 \subset \cdots$$

Each class has an associated noise rate $\nu_i = \inf_{h \in \mathbb{C}_i} er(h)$, and we define $\nu_\infty = \lim_{i \to \infty} \nu_i$. We also let θ_i and d_i be the disagreement coefficient and VC dimension, respectively, for the set $\mathbb{C}_i$. We are interested in an algorithm that guarantees convergence in probability of the error rate to ν_∞.

We are particularly interested in situations where $\nu_\infty = \nu^*$, a condition which is realistic in this setting since $\mathbb{C}_i$ can be defined so that it is always satisfied [see e.g., Devroye, Györfi, and Lugosi, 1996]. Additionally, if we are so lucky as to have some $\nu_i = \nu^*$, then we would like the convergence rate achieved by the algorithm to be not significantly worse than running one of the above agnostic active learning algorithms with hypothesis class $\mathbb{C}_i$ alone. In this context, we can define a structure-dependent version of Tsybakov's noise condition by $\bigcap_{i \in I} \mathcal{T}sybakov(\mathbb{C}_i, \kappa_i, \mu_i)$, for some $I \subseteq \mathbb{N}$, and finite parameters $\kappa_i \geq 1$ and $\mu_i > 0$.

In passive learning, there are several methods for this type of model selection which are known to preserve the convergence rates of each class $\mathbb{C}_i$ under $\mathcal{T}sybakov(\mathbb{C}_i, \kappa_i, \mu_i)$. [e.g., Koltchinskii, 2006, Tsybakov, 2004]. In particular, Koltchinskii [2006] develops a method that performs this type of model selection; it turns out we can modify Koltchinskii's method to suit our present needs in the context of active learning; this results in a general active learning model selection method that preserves the types of improved rates discussed in the previous section. This modification is presented below, based on using Algorithm 2 as a subroutine. (It may also be possible to define an analogous method that uses Algorithm 1 as a subroutine instead.)

Algorithm 3
Input: nested sequence of classes $\{\mathbb{C}_i\}$, label budget n, confidence parameter δ
Output: classifier $\hat{h}_n$

---

0. For $i = \lfloor \sqrt{n/2} \rfloor, \lfloor \sqrt{n/2} \rfloor - 1, \lfloor \sqrt{n/2} \rfloor - 2, \ldots, 1$
1.   Let $\mathcal{L}_{in}$ and Q_{in} be the sets returned by Algorithm 2 run with $\mathbb{C}_i$ and the threshold in (2.6), allowing $\lfloor n/(2i^2) \rfloor$ label requests, and confidence $\delta/(2i^2)$
2.   Let $h_{in} \leftarrow \text{LEARN}_{\mathbb{C}_i}(\cup_{j \geq i} \mathcal{L}_{jn}, Q_{in})$
3.   If $h_{in} \neq \varnothing$ and $\forall j$ s.t. $i < j \leq \lfloor \sqrt{n/2} \rfloor$,
$$er_{\mathcal{L}_{jn} \cup Q_{jn}}(h_{in}) - er_{\mathcal{L}_{jn} \cup Q_{jn}}(h_{jn}) \leq \tfrac{3}{2}\hat{\mathcal{E}}_{\mathbb{C}_j}(\mathcal{L}_{jn} \cup Q_{jn}, \delta/(2j^2); \mathcal{L}_{jn})$$
4. $\hat{h}_n \leftarrow h_{in}$
5. Return $\hat{h}_n$

The function $\hat{\mathcal{E}}_{\cdot}(\cdot, \cdot; \cdot)$ is defined in Section 2.6. This method can be shown to correctly converge in probability to an error rate of ν_∞ at a rate never significantly worse than the original passive learning method of Koltchinskii [2006], as desired. Additionally, we have the following guarantee on the rate of convergence under the structure-dependent definition of Tsybakov's

noise conditions. The proof is similar in style to Koltchinskii's original proof, though some care is needed due to the altered sampling distribution and the constraint set $\mathcal{L}_{jn}$. The proof is included in Section 2.7.

Theorem 2.13. *Suppose $\hat{h}_n$ is the classifier returned by Algorithm 3, when allowed n label requests and confidence parameter $\delta \in (0, 1/2)$. Suppose further that*

$\mathcal{D}_{XY} \in \bigcap_{i \in I} \mathcal{T}sybakov(\mathbb{C}_i, \kappa_i, \mu_i)$ for some nonempty $I \subseteq \mathbb{N}$ and for finite parameter values $\kappa_i \geq 1$ and $\mu_i > 0$. Then there exist finite (κ_i and μ_i-dependent) constants c_i such that, with probability $\geq 1 - \delta$, $\forall n \geq 2$,

$$er(\hat{h}_n) - \nu_\infty \leq 3\min_{i \in I}(\nu_i - \nu_\infty) + \begin{cases} \frac{1}{\delta} \cdot exp\left\{-\sqrt{\frac{n}{c_i d_i \theta_i \log^3 \frac{d_i}{\delta}}}\right\}, & \text{if } \kappa_i = 1 \\ c_i \left(\frac{d_i \theta_i \log^2 \frac{d_i n}{\delta}}{n}\right)^{\frac{\kappa_i}{2\kappa_i - 2}}, & \text{if } \kappa_i > 1 \end{cases}.$$

In particular, if we are so lucky as to have $\nu_i = \nu^*$ for some finite $i \in I$, then the above algorithm achieves a convergence rate not significantly worse than that guaranteed by Theorem 2.11 for applying Algorithm 2 directly, with hypothesis class $\mathbb{C}_i$.

As in the case of finite-complexity $\mathbb{C}$, we can also show a variant of this result when the complexities are quantified in terms of the entropy with bracketing. Specifically, consider the following theorem; the proof is in Section 2.7. Again, this represents an improvement over known results for passive learning when the disagreement coefficient is small.

Theorem 2.14. *Suppose $\hat{h}_n$ is the classifier returned by Algorithm 3, when allowed n label requests and confidence parameter $\delta \in (0, 1/2)$. Suppose further that*

$\mathcal{D}_{XY} \in \bigcap_{i \in I} \mathcal{T}sybakov(\mathbb{C}_i, \kappa_i, \mu_i) \cap \mathcal{E}ntropy_{[]}(\mathbb{C}_i, \alpha_i, \rho_i)$ for some nonempty $I \subseteq \mathbb{N}$ and finite parameters $\mu_i > 0$, $\kappa_i \geq 1$, $\alpha_i > 0$ and $\rho_i \in (0, 1)$. Then there exist finite (κ_i, μ_i, α_i and ρ_i-dependent) constants c_i such that, with probability $\geq 1 - \delta$, $\forall n \geq 2$,

$$er(\hat{h}_n) - \nu_\infty \leq 3\min_{i \in I}(\nu_i - \nu_\infty) + c_i \left(\frac{\theta_i \log^2 \frac{in}{\delta}}{n}\right)^{\frac{\kappa_i}{2\kappa_i + \rho_i - 2}}.$$

In addition to these theorems for this structure-dependent version of Tsybakov's noise conditions, we also have the following result for a structure-independent version.

> **Theorem 2.15.** *Suppose $\hat{h}_n$ is the classifier returned by Algorithm 3, when allowed n label requests and confidence parameter $\delta \in (0, 1/2)$. Suppose further that there exists a constant $\mu > 0$ such that for all measurable $h : \mathcal{X} \to \{-1, 1\}$, $er(h) - \nu^* \geq \mu\mathbb{P}\{h(X) \neq h^*(X)\}$. Then there exists a finite (μ-dependent) constant c such that, with probability $\geq 1 - \delta$, $\forall n \geq 2$,*
>
> $$er(\hat{h}_n) - \nu^* \leq c\min_i(\nu_i - \nu^*) + exp\left\{-\sqrt{\frac{n}{cd_i\theta_i\log^3\frac{id_i}{\delta}}}\right\}.$$

The case where $er(h) - \nu^* \geq \mu\mathbb{P}\{h(X) \neq h^*(X)\}^\kappa$ for $\kappa > 1$ can be studied analogously, though the rate improvements over passive learning are more subtle.

2.5 Conclusions

Under Tsybakov's noise conditions, active learning can offer improved asymptotic convergence rates compared to passive learning when the disagreement coefficient is small. It is also possible to preserve these improved convergence rates when learning with a nested structure of hypothesis classes, using an algorithm that adapts to both the noise conditions and the complexity of the optimal classifier.

2.6 Definition of $\hat{\mathcal{E}}$

For any function $f : \mathcal{X} \to \mathbb{R}$, and $\xi_1, \xi_2, \ldots$ a sequence of independent random variables with distribution uniform in $\{-1, +1\}$, define the *Rademacher process* for f under a finite sequence of labeled examples $Q = \{(X_i', Y_i')\}$ as

$$R(f; Q) = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\xi_i f(X_i').$$

The ξ_i should be thought of as internal variables in the learning algorithm, rather than being fundamental to the learning problem.

For any two sequences of labeled examples $\mathcal{L} = \{(X_i', Y_i')\}$ and $Q = \{(X_i'', Y_i'')\}$, define
$\mathbb{C}[\mathcal{L}] = \{h \in \mathbb{C} : er_{\mathcal{L}}(h) = 0\}$,

$$\hat{\mathbb{C}}(\epsilon; \mathcal{L}, Q) = \{h \in \mathbb{C}[\mathcal{L}] : er_Q(h) - \min_{h' \in \mathbb{C}[\mathcal{L}]} er_Q(h') \leq \epsilon\},$$

let

$$\hat{D}_{\mathbb{C}}(\epsilon; \mathcal{L}, Q) = \sup_{h_1, h_2 \in \hat{\mathbb{C}}(\epsilon; \mathcal{L}, Q)} \frac{1}{|Q|} \sum_{i=1}^{|Q|} \mathbb{1}[h_1(X_i'') \neq h_2(X_i'')],$$

and define

$$\hat{\phi}_{\mathbb{C}}(\epsilon; \mathcal{L}, Q) = \frac{1}{2} \sup_{h_1, h_2 \in \hat{\mathbb{C}}(\epsilon; \mathcal{L}, Q)} R(h_1 - h_2; Q).$$

Let $\delta \in (0, 1]$, $m \in \mathbb{N}$, and define

$$s_m(\delta) = \ln \frac{20m^2 \log_2(3m)}{\delta}.$$

Let $\mathbb{Z}_\epsilon = \{j \in \mathbb{Z} : 2^j \geq \epsilon\}$, and for any sequence of labeled examples $Q = \{(X_i', Y_i')\}$, define $Q_m = \{(X_1', Y_1'), (X_2', Y_2'), \ldots, (X_m', Y_m')\}$. We use the following notation of Koltchinskii Koltchinskii [2006] with only minor modifications. For $\epsilon \in [0, 1]$, define

$$\hat{U}_{\mathbb{C}}(\epsilon, \delta; \mathcal{L}, Q) = \hat{K}\left(\hat{\phi}_{\mathbb{C}}(\hat{c}\epsilon; \mathcal{L}, Q) + \sqrt{\frac{s_{|Q|}(\delta)\hat{D}_{\mathbb{C}}(\hat{c}\epsilon; \mathcal{L}, Q)}{|Q|}} + \frac{s_{|Q|}(\delta)}{|Q|}\right)$$

$$\hat{\mathcal{E}}_{\mathbb{C}}(Q, \delta; \mathcal{L}) = \min_{m \leq |Q|} \inf\left\{\epsilon > 0 : \forall j \in \mathbb{Z}_\epsilon, \hat{U}_{\mathbb{C}}(2^j, \delta; \mathcal{L}, Q_m) \leq 2^{j-4}\right\}$$

where, for our purposes, we can take $\hat{K} = 752$, and $\hat{c} = 3/2$, though there seems to be room for improvement in these constants. We also define $\hat{\mathcal{E}}_{\mathbb{C}}(\emptyset, \delta; \mathbb{C}, \mathcal{L}) = \infty$ by convention.

2.7 Main Proofs

Let $\hat{\mathcal{E}}_{\mathbb{C}}(m, \delta) = \hat{\mathcal{E}}_{\mathbb{C}}(\mathcal{Z}_m, \delta; \emptyset)$. For each $m \in \mathbb{N}$, let $\hat{h}_m^* = \arg\min_{h \in \mathbb{C}} er_m(h)$ be the empirical risk minimizer in $\mathbb{C}$ for the *true* labels of the first m examples.

For $\epsilon > 0$, define $\mathbb{C}(\epsilon) = \{h \in \mathbb{C} : er(h) - \nu \leq \epsilon\}$. For $m \in \mathbb{N}$, let

$$\phi_{\mathbb{C}}(m, \epsilon) = \mathbb{E} \sup_{h_1, h_2 \in \mathbb{C}(\epsilon)} |(er(h_1) - er_m(h_1)) - (er(h_2) - er_m(h_2))|,$$

36

$$\tilde{U}_{\mathbb{C}}(m, \epsilon, \delta) = \tilde{K}\left(\phi_{\mathbb{C}}(m, \tilde{c}\epsilon) + \sqrt{\frac{s_m(\delta)diam(\mathbb{C}(\tilde{c}\epsilon))}{m}} + \frac{s_m(\delta)}{m}\right),$$

$$\tilde{\mathcal{E}}_{\mathbb{C}}(m, \delta) = \inf\left\{\epsilon > 0 : \forall j \in \mathbb{Z}_\epsilon, \tilde{U}_{\mathbb{C}}(m, 2^j, \delta) \leq 2^{j-4}\right\},$$

where, for our purposes, we can take $\tilde{K} = 8272$ and $\tilde{c} = 3$. We also define $\tilde{\mathcal{E}}_{\mathbb{C}}(0, \delta) = \infty$. The following lemma is crucial to all of the proofs that follow.

Lemma 2.16. *[Koltchinskii, 2006] There is an event $E_{\mathbb{C},\delta}$ with $\mathbb{P}(E_{\mathbb{C},\delta}) \geq 1 - \delta/2$ such that, on event $E_{\mathbb{C},\delta}$, $\forall m \in \mathbb{N}, \forall h \in \mathbb{C}, \forall \tau \in (0, 1/m), \forall h' \in \mathbb{C}(\tau)$,*

$$er(h) - \nu \leq \max\left\{2(er_m(h) - er_m(h') + \tau), \hat{\mathcal{E}}_{\mathbb{C}}(m, \delta)\right\}$$

$$er_m(h) - er_m(\hat{h}_n^*) \leq \tfrac{3}{2}\max\left\{(er(h) - \nu), \hat{\mathcal{E}}_{\mathbb{C}}(m, \delta)\right\},$$

$$\hat{\mathcal{E}}_{\mathbb{C}}(m, \delta) \leq \tilde{\mathcal{E}}_{\mathbb{C}}(m, \delta),$$

and for any $j \in \mathbb{Z}$ with $2^j > \hat{\mathcal{E}}_{\mathbb{C}}(m, \delta)$,

$$\sup_{h_1, h_2 \in \mathbb{C}(2^j)} |(er_m(h_1) - er(h_1)) - (er_m(h_2) - er(h_2))| \leq \hat{U}_{\mathbb{C}}(2^j, \delta; \emptyset, \mathcal{Z}_m).$$

This lemma essentially follows from details of the proof of Koltchinskii's Theorem 1, Lemma 2, and Theorem 3 [Koltchinskii, 2006][1]. We do not provide a proof of Lemma 2.16 here. The reader is referred to Koltchinskii's paper for the details.

2.7.1 Definition of r_0

If θ is bounded by a finite constant, the definition of r_0 is not too important. However, in some cases, setting $r_0 = 0$ results in a suboptimal, or even infinite, value of θ, which is undesirable. In these cases, we would like to set r_0 as large as possible while maintaining the validity of the bounds, and if we do this carefully we should be able to establish bounds that, even in the worst case when $\theta = 1/r_0$, are never worse than the bounds for some analogous passive learning

---

[1]Our $\min_{m \leq |Q|}$ modification to Koltchinskii's version of $\hat{\mathcal{E}}_{\mathbb{C}}(m, \delta)$ is not a problem, since $\phi_{\mathbb{C}}(m, \epsilon)$ and $\frac{s_m(\delta)}{m}$ are nonincreasing functions of m.

method; however, to do this requires r_0 to depend on the parameters of the learning problem: namely, n, δ, $\mathbb{C}$, and $\mathcal{D}_{XY}$.

Generally, depending on the bound we wish to prove, different values of r_0 may be appropriate. For the tightest bound in terms of θ proven below (namely, Lemma 2.18), the following definition of r_0 gives a good bound. Defining

$$\tilde{m}_{\mathbb{C}}(n, \delta, \mathcal{D}_{XY}) = \min\left\{ m \in \mathbb{N} : n \le \log_2 \frac{4m^2}{\delta} + 2e \sum_{\ell=0}^{m-1} \mathbb{P}(DIS(\mathbb{C}(2\tilde{\mathcal{E}}_{\mathbb{C}}(\ell, \delta)))) \right\}, \quad (2.7)$$

we can let $r_0 = r_{\mathbb{C}}(n, \delta, \mathcal{D}_{XY})$, where

$$r_{\mathbb{C}}(n, \delta, \mathcal{D}_{XY}) = \frac{1}{\tilde{m}_{\mathbb{C}}(n, \delta, \mathcal{D}_{XY})} \sum_{\ell=0}^{\tilde{m}_{\mathbb{C}}(n, \delta, \mathcal{D}_{XY})-1} diam(\mathbb{C}(2\tilde{\mathcal{E}}_{\mathbb{C}}(m_{\mathbb{C}}(r', n, \delta), \delta))). \quad (2.8)$$

We use this definition in all of the proofs below. In particular, with this definition, Lemma 2.18 is never significantly worse than the analogous known result for passive learning (though it can be significantly better when $\theta << 1/r_0$). For the looser bounds (namely, Theorems 2.11 and 2.12), a larger value of r_0 would be more appropriate; however, note that this same general technique can be employed to define a good value for r_0 in these looser bounds as well, simply using upper bounds on (2.8) analogous to how the theorems themselves are derived from Lemma 2.18 below.

2.7.2 Proofs Relating to Section 2.3

For $\ell \in \mathbb{N} \cup \{0\}$, let $\mathcal{L}^{(\ell)}$ and $Q^{(\ell)}$ denote the sets $\mathcal{L}$ and Q, respectively, in step 4 of Algorithm 2, when $m - 1 = \ell$; if this never happens during execution, then define $\mathcal{L}^{(\ell)} = \emptyset$, $Q^{(\ell)} = \mathcal{Z}_\ell$.

Lemma 2.17. *On event $E_{\mathbb{C},\delta}$, $\forall \ell \in \mathbb{N} \cup \{0\}$,*

$$\hat{\mathcal{E}}_{\mathbb{C}}(Q^{(\ell)} \cup \mathcal{L}^{(\ell)}, \delta; \mathcal{L}^{(\ell)}) = \hat{\mathcal{E}}_{\mathbb{C}}(\ell, \delta)$$

and

$$\forall \epsilon \ge \hat{\mathcal{E}}_{\mathbb{C}}(\ell, \delta), \ \ \hat{h}_\ell^* \in \hat{\mathbb{C}}_\ell(\epsilon; \mathcal{L}^{(\ell)}) \subseteq \hat{\mathbb{C}}_\ell(\epsilon; \emptyset).$$

Proof of Lemma 2.17. Throughout this proof, we assume the event $E_{\mathbb{C},\delta}$ occurs. We proceed by induction on ℓ, with the base case of $\ell = 0$ (which clearly holds). Suppose the statements are true

for all $\ell' < \ell$. The case $\mathcal{L}^{(\ell)} = \emptyset$ is trivial, so assume $\mathcal{L}^{(\ell)} \neq \emptyset$. For the inductive step, suppose

$$h \in \hat{\mathbb{C}}_\ell(\hat{\mathcal{E}}_\mathbb{C}(\ell, \delta); \emptyset).$$

Then for all $\ell' < \ell$, we have

$$er_\ell(h) - er_\ell(\hat{h}_\ell^*) \leq \hat{\mathcal{E}}_\mathbb{C}(\ell', \delta).$$

In particular, by Lemma 2.16, this implies

$$er(h) - \nu \leq \max\left\{2(er_\ell(h) - er_\ell(\hat{h}_\ell^*)), \hat{\mathcal{E}}_\mathbb{C}(\ell, \delta)\right\} \leq 2\hat{\mathcal{E}}_\mathbb{C}(\ell', \delta),$$

and thus for any $h' \in \mathbb{C}$,

$$er_{\ell'}(h) - er_{\ell'}(h') \leq er_{\ell'}(h) - er_{\ell'}(\hat{h}_{\ell'}^*)$$
$$\leq \frac{3}{2}\max\left\{er(h) - \nu, \hat{\mathcal{E}}_\mathbb{C}(\ell', \delta)\right\} \leq 3\hat{\mathcal{E}}_\mathbb{C}(\ell', \delta) = 3\hat{\mathcal{E}}_\mathbb{C}(Q^{(\ell')}, \delta; \mathcal{L}^{(\ell')}).$$

Thus, we must have $er_{\mathcal{L}^{(\ell)}}(h) = 0$, and therefore $h \in \hat{\mathbb{C}}_\ell(\hat{\mathcal{E}}_\mathbb{C}(\ell, \delta); \mathcal{L}^{(\ell)})$. Since this is the case for all such h, we must have that

$$\hat{\mathbb{C}}_\ell(\hat{\mathcal{E}}_\mathbb{C}(\ell, \delta); \mathcal{L}^{(\ell)}) \supseteq \hat{\mathbb{C}}_\ell(\hat{\mathcal{E}}_\mathbb{C}(\ell, \delta); \emptyset). \tag{2.9}$$

In particular, this implies that

$$\hat{U}_\mathbb{C}(\hat{\mathcal{E}}_\mathbb{C}(\ell, \delta), \delta; \mathcal{L}^{(\ell)}, Q^{(\ell)}) \geq \hat{U}_\mathbb{C}(\hat{\mathcal{E}}_\mathbb{C}(\ell, \delta), \delta; \emptyset, \mathcal{Z}_\ell) > \frac{1}{16}\hat{\mathcal{E}}_\mathbb{C}(\ell, \delta),$$

where the last inequality follows from the definition of $\hat{\mathcal{E}}_\mathbb{C}(\ell, \delta)$, (which is a power of 2). Thus, we must have $\hat{\mathcal{E}}_\mathbb{C}(Q^{(\ell)} \cup \mathcal{L}^{(\ell)}, \delta; \mathcal{L}^{(\ell)}) \geq \hat{\mathcal{E}}_\mathbb{C}(\ell, \delta)$.

The relation in (2.9) also implies that

$$\hat{h}_\ell^* \in \hat{\mathbb{C}}_\ell(\hat{\mathcal{E}}_\mathbb{C}(\ell, \delta); \mathcal{L}^{(\ell)}),$$

and therefore

$$\forall \epsilon \geq \hat{\mathcal{E}}_\mathbb{C}(\ell, \delta), \ \ \hat{\mathbb{C}}_\ell(\epsilon; \mathcal{L}^{(\ell)}) \subseteq \hat{\mathbb{C}}_\ell(\epsilon; \emptyset),$$

39

which implies

$$\forall \epsilon \geq \hat{\mathcal{E}}_{\mathbb{C}}(\ell, \delta), \ \hat{U}_{\mathbb{C}}(\epsilon, \delta; \mathcal{L}^{(\ell)}, Q^{(\ell)}) \leq \hat{U}_{\mathbb{C}}(\epsilon, \delta; \emptyset, \mathcal{Z}_\ell).$$

But this means $\hat{\mathcal{E}}_{\mathbb{C}}(Q^{(\ell)} \cup \mathcal{L}^{(\ell)}, \delta; \mathcal{L}^{(\ell)}) \leq \hat{\mathcal{E}}_{\mathbb{C}}(\ell, \delta)$. Therefore, we must have equality. Thus, the lemma follows by the principle of induction. □

Lemma 2.18. *Suppose for any $n \in \mathbb{N}$, $\hat{h}_n$ is the classifier returned by Algorithm 2 with threshold as in (2.6), when allowed n label requests and given confidence parameter $\delta > 0$, and suppose further that m_n is the value of $|Q| + |\mathcal{L}|$ when Algorithm 2 returns. Then there is an event $H_{\mathbb{C},\delta}$ such that $\mathbb{P}(H_{\mathbb{C},\delta} \cap E_{\mathbb{C},\delta}) \geq 1 - \delta$, such that on $H_{\mathbb{C},\delta} \cap E_{\mathbb{C},\delta}$, $\forall n \in \mathbb{N}$,*

$$er(\hat{h}_n) - \nu \leq \tilde{\mathcal{E}}_{\mathbb{C}}(m_n, \delta),$$

and

$$n \leq \min \left\{ m_n, \log_2 \frac{4m_n^2}{\delta} + 4e\theta \sum_{\ell=0}^{m_n-1} diam(\mathbb{C}(2\tilde{\mathcal{E}}_{\mathbb{C}}(\ell, \delta))) \right\}.$$

Proof of Lemma 2.18. Once again, assume event $E_{\mathbb{C},\delta}$ occurs. By Lemma 2.16, $\forall \tau > 0$,

$$er(\hat{h}_n) - \nu \leq \max \left\{ 2(er_{m_n}(\hat{h}_n) - er_{m_n}(\hat{h}^*_{m_n}) + \tau), \hat{\mathcal{E}}_{\mathbb{C}}(m_n, \delta) \right\}.$$

Letting $\tau \to 0$, and noting that $er_{\mathcal{L}}(\hat{h}^*_{m_n}) = 0$ (Lemma 2.17) implies $er_{m_n}(\hat{h}_n) = er_{m_n}(\hat{h}^*_{m_n})$, we have

$$er(\hat{h}_n) - \nu \leq \hat{\mathcal{E}}_{\mathbb{C}}(m_n, \delta) \leq \tilde{\mathcal{E}}_{\mathbb{C}}(m_n, \delta),$$

where the last inequality is also due to Lemma 2.16. Note that this $\hat{\mathcal{E}}_{\mathbb{C}}(m_n, \delta)$ represents an interesting data-dependent bound.

To get the bound on the number of label requests, we proceed as follows. For any $m \in \mathbb{N}$, and nonnegative integer $\ell < m$, let I_ℓ be the indicator for the event that Algorithm 2 requests the label $Y_{\ell+1}$ and let $N_m = \sum_{\ell=0}^{m-1} I_\ell$. Additionally, let I'_ℓ be independent Bernoulli random variables with

$$\mathbb{P}[I'_\ell = 1] = \mathbb{P}\left\{ DIS(\mathbb{C}(2\tilde{\mathcal{E}}_{\mathbb{C}}(\ell, \delta))) \right\}.$$

40

Let $N'_m = \sum_{\ell=0}^{m-1} I'_\ell$. We have that

$$\mathbb{P}\left[\{I_\ell = 1\} \cap E_{\mathbb{C},\delta}\right] \leq \mathbb{P}\left[\{X_{\ell+1} \in DIS(\hat{\mathbb{C}}_\ell(\hat{\mathcal{E}}_{\mathbb{C}}(Q^{(\ell)} \cup \mathcal{L}^{(\ell)}, \delta; \mathcal{L}_i^{(\ell)}); \mathcal{L}^{(\ell)}))\} \cap E_{\mathbb{C},\delta}\right]$$

$$\leq \mathbb{P}\left[\{X_{\ell+1} \in DIS(\hat{\mathbb{C}}_\ell(\tilde{\mathcal{E}}_{\mathbb{C}}(\ell, \delta); \emptyset))\} \cap E_{\mathbb{C},\delta}\right] \leq \mathbb{P}\left[DIS(\mathbb{C}(2\tilde{\mathcal{E}}_{\mathbb{C}}(\ell, \delta)))\right] = \mathbb{P}[I'_\ell = 1].$$

The second inequality is due to Lemmas 2.17 and 2.16, while the third inequality is due to Lemma 2.16. Note that

$$\mathbb{E}[N'_m] = \sum_{\ell=0}^{m-1} \mathbb{P}[I'_\ell = 1] = \sum_{\ell=0}^{m-1} \mathbb{P}\left\{DIS(\mathbb{C}(2\tilde{\mathcal{E}}_{\mathbb{C}}(\ell, \delta)))\right\}$$

Let us name this last quantity q_m. Thus, by union and Chernoff bounds,

$$\mathbb{P}\left[\left\{\exists m \in \mathbb{N} : N_m > \max\left\{2eq_m, q_m + \log_2 \frac{4m^2}{\delta}\right\}\right\} \cap E_{\mathbb{C},\delta}\right]$$

$$\leq \sum_{m\in\mathbb{N}} \mathbb{P}\left[\left\{N_m > \max\left\{2eq_m, q_m + \log_2 \frac{4m^2}{\delta}\right\}\right\} \cap E_{\mathbb{C},\delta}\right]$$

$$\leq \sum_{m\in\mathbb{N}} \mathbb{P}\left[\left\{N'_m > \max\left\{2eq_m, q_m + \log_2 \frac{4m^2}{\delta}\right\}\right\}\right] \leq \sum_{m\in\mathbb{N}} \frac{\delta}{4m^2} \leq \frac{\delta}{2}.$$

For any n, we know $n \leq m_n \leq 2^n$. Therefore, we have that on an event (which includes $E_{\mathbb{C},\delta}$) occuring with probability $\geq 1 - \delta$, for every $n \in \mathbb{N}$,

$$n \leq \max\{N_{m_n}, \log_2 m_n\} \leq \max\left\{2eq_{m_n}, q_{m_n} + \log_2 \frac{4m_n^2}{\delta}\right\}$$

$$\leq \log_2 \frac{4m_n^2}{\delta} + 2e \sum_{\ell=0}^{m_n-1} \mathbb{P}\{DIS(\mathbb{C}(2\tilde{\mathcal{E}}_{\mathbb{C}}(\ell, \delta)))\}.$$

In particular, this implies $\tilde{m}_n = \tilde{m}_{\mathbb{C}}(n, \delta, \mathcal{D}_{XY}) \leq m_n$ (where $\tilde{m}_{\mathbb{C}}(n, \delta, \mathcal{D}_{XY})$ is defined in (2.7)). We now use the definition of θ with the r_0 in (2.8).

$$n \leq \log_2 \frac{4\tilde{m}_n^2}{\delta} + 2e \sum_{\ell=0}^{\tilde{m}_n-1} \mathbb{P}\{DIS(\mathbb{C}(2\tilde{\mathcal{E}}_{\mathbb{C}}(\ell, \delta)))\}$$

$$\leq \log_2 \frac{4\tilde{m}_n^2}{\delta} + 2e\theta \sum_{\ell=0}^{\tilde{m}_n-1} \max\{diam(\mathbb{C}(2\tilde{\mathcal{E}}_{\mathbb{C}}(\ell, \delta))), r_{\mathbb{C}}(n, \delta, \mathcal{D}_{XY})\}$$

$$\leq \log_2 \frac{4\tilde{m}_n^2}{\delta} + 4e\theta \sum_{\ell=0}^{\tilde{m}_n-1} diam(\mathbb{C}(2\tilde{\mathcal{E}}_{\mathbb{C}}(\ell, \delta))) \leq \log_2 \frac{4m_n^2}{\delta} + 4e\theta \sum_{\ell=0}^{m_n-1} diam(\mathbb{C}(2\tilde{\mathcal{E}}_{\mathbb{C}}(\ell, \delta))).$$

$$\square$$

Lemma 2.19. *On event $H_{\mathbb{C},\delta} \cap E_{\mathbb{C},\delta}$ (where $H_{\mathbb{C},\delta}$ is from Lemma 2.18), under*

$\mathfrak{T}sybakov(\mathbb{C}, \kappa, \mu)$, $\forall n \geq 2$,

$$\tilde{\mathcal{E}}_{\mathbb{C}}(m_n, \delta) \leq \begin{cases} \frac{1}{\delta} \cdot exp\left\{-\sqrt{\frac{n}{cd\theta \log^3 \frac{d}{\delta}}}\right\}, & if \, \kappa = 1 \\ c\left(\frac{d\theta \log^2(nd/\delta)}{n}\right)^{\frac{\kappa}{2\kappa-2}}, & if \, \kappa > 1 \end{cases},$$

for some finite constant c (depending on κ and μ), and under

$\mathcal{E}ntropy_{[]}(\mathbb{C}, \alpha, \rho) \cap \mathfrak{T}sybakov(\mathbb{C}, \kappa, \mu)$, $\forall n \in \mathbb{N}$,

$$\tilde{\mathcal{E}}_{\mathbb{C}}(m_n, \delta) \leq c\left(\frac{\theta \log^2(n/\delta)}{n}\right)^{\frac{\kappa}{2\kappa+\rho-2}},$$

for some finite constant c (depending on κ, μ, ρ, and α).

Proof of Lemma 2.19. We begin with the first case ($\mathfrak{T}sybakov(\mathbb{C}, \kappa, \mu)$ only).

We know that

$$\omega_{\mathbb{C}}(m, \epsilon) \leq K\sqrt{\frac{\epsilon d \log \frac{2}{\epsilon}}{m}}$$

for some constant K [see e.g., Massart and Élodie Nédélec, 2006]. Noting that $\phi_{\mathbb{C}}(m, \epsilon) \leq \omega_{\mathbb{C}}(m, diam(\mathbb{C}(\epsilon)))$, we have that

$$\tilde{U}_{\mathbb{C}}(m, \epsilon, \delta) \leq \tilde{K}\left(K\sqrt{\frac{diam(\mathbb{C}(\tilde{c}\epsilon))d \log \frac{2}{diam(\mathbb{C}(\tilde{c}\epsilon))}}{m}} + \sqrt{\frac{s_m(\delta)diam(\mathbb{C}(\tilde{c}\epsilon))}{m}} + \frac{s_m(\delta)}{m}\right)$$

$$\leq K' \max\left\{\sqrt{\frac{\epsilon^{1/\kappa}d \log \frac{1}{\epsilon}}{m}}, \sqrt{\frac{s_m(\delta)\epsilon^{1/\kappa}}{m}}, \frac{s_m(\delta)}{m}\right\}.$$

Taking any $\epsilon \geq K''\left(\frac{d \log \frac{m}{\delta}}{m}\right)^{\frac{\kappa}{2\kappa-1}}$, for some constant $K'' > 0$, suffices to make this latter quantity $\leq \frac{\epsilon}{16}$. So for some appropriate constant K (depending on μ and κ), we must have that

$$\tilde{\mathcal{E}}_{\mathbb{C}}(m, \delta) \leq K\left(\frac{d \log \frac{m}{\delta}}{m}\right)^{\frac{\kappa}{2\kappa-1}}. \tag{2.10}$$

Plugging this into the query bound, we have that

$$n \leq \log_2 \frac{4m_n^2}{\delta} + 2e\theta\left(2 + \int_1^{m_n-1} \mu(2K')^{\frac{1}{\kappa}}\left(\frac{d \log \frac{x}{\delta}}{x}\right)^{\frac{1}{2\kappa-1}}\right). \tag{2.11}$$

42

If $\kappa > 1$, (2.11) is at most $K'' \theta m_n^{\frac{2\kappa-2}{2\kappa-1}} d \log \frac{m_n}{\delta}$, for some constant K'' (depending on κ and μ). This implies

$$m_n \geq K^{(3)} \left(\frac{n}{\theta d \log \frac{n}{\delta}} \right)^{\frac{2\kappa-1}{2\kappa-2}},$$

for some constant $K^{(3)}$. Plugging this into (2.10) and using Lemma 2.18 completes the proof for this case.

On the other hand, if $\kappa = 1$, (2.11) is at most $K'' \theta d \log^2 \frac{m_n}{\delta}$, for some constant K'' (depending on κ and μ). This implies

$$m_n \geq \delta exp \left\{ K^{(3)} \sqrt{\frac{n}{\theta d}} \right\},$$

for some constant $K^{(3)}$. Plugging this into (2.10), using Lemma 2.18, and simplifying the expression with a bit of algebra completes this case.

For the bound in terms of ρ, Koltchinskii [2006] proves that

$$\tilde{\mathcal{E}}_{\mathbb{C}}(m, \delta) \leq K' \max \left\{ m^{-\frac{\kappa}{2\kappa+\rho-1}}, \left(\frac{\log \frac{m}{\delta}}{m} \right)^{\frac{\kappa}{2\kappa-1}} \right\} \leq K' \left(\frac{\log \frac{m}{\delta}}{m} \right)^{\frac{\kappa}{2\kappa+\rho-1}}, \tag{2.12}$$

for some constant K' (depending on μ, α, and κ). Plugging this into the query bound, we have that

$$n \leq \log_2 \frac{4m_n^2}{\delta} + 2e\theta \left(2 + \int_1^{m_n-1} \mu(2K')^{\frac{1}{\kappa}} \left(\frac{\log \frac{x}{\delta}}{x} \right)^{\frac{1}{2\kappa+\rho-1}} \right) \leq K'' \theta m_n^{\frac{2\kappa+\rho-2}{2\kappa+\rho-1}} \log \frac{m_n}{\delta},$$

for some constant K'' (depending on κ, μ, α, and ρ). This implies

$$m_n \geq K^{(3)} \left(\frac{n}{\theta \log \frac{n}{\delta}} \right)^{\frac{2\kappa+\rho-1}{2\kappa+\rho-2}},$$

for some constant $K^{(3)}$. Plugging this into (2.12) and using Lemma 2.18 completes the proof of this case. $\qquad \square$

Proofs of Theorem 2.11 and Theorem 2.12. These theorems now follow directly from Lemmas 2.18 and 2.19. $\qquad \square$

2.7.3 Proofs Relating to Section 2.4

> **Lemma 2.20.** *For $i \in \mathbb{N}$, let $\delta_i = \delta/(2i^2)$ and $m_{in} = |\mathcal{L}_{in}| + |Q_{in}|$ (for $i > \sqrt{n/2}$, define $\mathcal{L}_{in} = Q_{in} = \emptyset$). For each n, let $\hat{i}_n$ denote the smallest index i satisfying the condition on h_{in} in step 3 of Algorithm 3. Let $\tau_n = 2^{-n}$ and define*
> $$i_n^* = \min \left\{ i \in \mathbb{N} : \forall i' \geq i, \forall j \geq i', \forall h \in \mathbb{C}_{i'}(\tau_n), er_{\mathcal{L}_{jn}}(h) = 0 \right\},$$
> *and*
> $$j_n^* = \arg\min_{j \in \mathbb{N}} \nu_j + \hat{\mathcal{E}}_{\mathbb{C}_j}(m_{jn}, \delta_j).$$
> *Then on the event $\bigcap_{i=1}^{\infty} E_{\mathbb{C}_i, \delta_i}$,*
> $$\forall n \in \mathbb{N}, \max \left\{ i_n^*, \hat{i}_n \right\} \leq j_n^*.$$

Proof of Lemma 2.20. Continuing the notation from the proof of Lemma 2.17, for $\ell \in \mathbb{N} \cup \{0\}$, let $\mathcal{L}_{in}^{(\ell)}$ and $Q_{in}^{(\ell)}$ denote the sets $\mathcal{L}$ and Q, respectively, in step 4 of Algorithm 2, when $m - 1 = \ell$, when run with class $\mathbb{C}_i$, label budget $\lfloor n/(2i^2) \rfloor$, confidence parameter δ_i, and threshold as in (2.6); if $m - 1$ is never ℓ during execution, then define $\mathcal{L}_{in}^{(\ell)} = \emptyset$ and $Q_{in}^{(\ell)} = \mathcal{Z}_\ell$.

Assume the event $\bigcap_{i=1}^{\infty} E_{\mathbb{C}_i, \delta_i}$ occurs. Suppose, for the sake of contradiction, that $j = j_n^* < i_n^*$ for some $n \in \mathbb{N}$. Then there is some $i \geq i_n^* - 1$ such that, for some $\ell < m_{in}$, we have some $h' \in \mathbb{C}_{i_n^*-1}(\tau_n) \cap \{h \in \mathbb{C}_i : er_{\mathcal{L}_{in}^{(\ell)}}(h) = 0\}$ but

$$er_\ell(h') - \min_{h \in \mathbb{C}_i} er_\ell(h) \geq er_\ell(h') - \min_{h \in \mathbb{C}_i : er_{\mathcal{L}_{in}^{(\ell)}}(h)=0} er_\ell(h) > 3\hat{\mathcal{E}}_{\mathbb{C}_i}(\mathcal{L}_{in}^{(\ell)} \cup Q_{in}^{(\ell)}, \delta_i; \mathcal{L}_{in}^{(\ell)}) = 3\hat{\mathcal{E}}_{\mathbb{C}_i}(\ell, \delta_i),$$

where the last equality is due to Lemma 2.17. Lemma 2.16 implies this will not happen for $i = i_n^* - 1$, so we can assume $i \geq i_n^*$. We therefore have (by Lemma 2.16) that

$$3\hat{\mathcal{E}}_{\mathbb{C}_i}(\ell, \delta_i) < er_\ell(h') - \min_{h \in \mathbb{C}_i} er_\ell(h) \leq \frac{3}{2} \max \left\{ \tau_n + \nu_{i_n^*-1} - \nu_i, \hat{\mathcal{E}}_{\mathbb{C}_i}(\ell, \delta_i) \right\}.$$

In particular, this implies that

$$3\hat{\mathcal{E}}_{\mathbb{C}_i}(m_{in}, \delta_i) \leq 3\hat{\mathcal{E}}_{\mathbb{C}_i}(\ell, \delta_i) < \frac{3}{2} \left(\tau_n + \nu_{i_n^*-1} - \nu_i \right) \leq \frac{3}{2} \left(\tau_n + \nu_j - \nu_i \right).$$

Therefore,

$$\hat{\mathcal{E}}_{\mathbb{C}_j}(m_{jn}, \delta_j) + \nu_j \leq \hat{\mathcal{E}}_{\mathbb{C}_i}(m_{in}, \delta_i) + \nu_i \leq \frac{1}{2} \left(\tau_n + \nu_j - \nu_i \right) + \nu_i \leq \frac{\tau_n}{2} + \nu_j.$$

This would imply that $\hat{\mathcal{E}}_{\mathbb{C}_j}(m_{jn}, \delta_j) \leq \tau_n/2 < \frac{1}{m_{jn}}$ (due to the second return condition in Algorithm 2), which by definition is not possible, so we have a contradiction. Therefore, we must have that every $j_n^* \geq i_n^*$. In particular, we have that $\forall n \in \mathbb{N}, h_{j_n^* n} \neq \varnothing$.

Now pick an arbitrary $i \in \mathbb{N}$ with $i > j = j_n^*$, and let $h' \in \mathbb{C}_j(\tau_n)$. Then

$$er_{\mathcal{L}_{in} \cup Q_{in}}(h_{jn}) - er_{\mathcal{L}_{in} \cup Q_{in}}(h_{in}) = er_{m_{in}}(h_{jn}) - er_{m_{in}}(h_{in})$$

$$\leq \quad er_{m_{in}}(h_{jn}) - \min_{h \in \mathbb{C}_i} er_{m_{in}}(h)$$

$$\leq \quad \frac{3}{2} \max \left\{ er(h_{jn}) - \nu_i, \hat{\mathcal{E}}_{\mathbb{C}_i}(m_{in}, \delta_i) \right\} \quad \text{(Lemma 2.16)}$$

$$= \quad \frac{3}{2} \max \left\{ er(h_{jn}) - \nu_j + \nu_j - \nu_i, \hat{\mathcal{E}}_{\mathbb{C}_i}(m_{in}, \delta_i) \right\}$$

$$\leq \quad \frac{3}{2} \max \begin{cases} 2(er_{m_{jn}}(h_{jn}) - er_{m_{jn}}(h') + \tau_n) + \nu_j - \nu_i \\ \hat{\mathcal{E}}_{\mathbb{C}_j}(m_{jn}, \delta_j) + \nu_j - \nu_i \\ \hat{\mathcal{E}}_{\mathbb{C}_i}(m_{in}, \delta_i) \end{cases}$$

$$= \quad \frac{3}{2} \max \begin{cases} \hat{\mathcal{E}}_{\mathbb{C}_j}(m_{jn}, \delta_j) + \nu_j - \nu_i \\ \hat{\mathcal{E}}_{\mathbb{C}_i}(m_{in}, \delta_i) \end{cases} \quad \text{(since } j \geq i_n^*\text{)}$$

$$= \quad \frac{3}{2} \hat{\mathcal{E}}_{\mathbb{C}_i}(m_{in}, \delta_i) \quad \text{(by definition of } j_t^*\text{)}$$

$$= \quad \frac{3}{2} \hat{\mathcal{E}}_{\mathbb{C}}(\mathcal{L}_{in} \cup Q_{in}, \delta_i; \mathcal{L}_{in}) \quad \text{(by Lemma 2.17).}$$

$\square$

Lemma 2.21. *On the event* $\bigcap_{i=1}^{\infty} E_{\mathbb{C}_i, \delta_i}$, $\forall n \in \mathbb{N}$,

$$er(h_{\hat{i}_n n}) - \nu_\infty \leq 3 \min_{i \in \mathbb{N}} \left(\nu_i - \nu_\infty + \tilde{\mathcal{E}}_{\mathbb{C}_i}(m_{in}, \delta_i) \right).$$

Proof of Lemma 2.21. Let $h'_n \in \mathbb{C}_{j_n^*}(\tau_n)$ for $\tau_n \in (0, 2^{-n})$, $n \in \mathbb{N}$.

45

$$er(\hat{h}_n) - \nu_\infty = er(h_{\hat{i}_n n}) - \nu_\infty$$

$$= \nu_{j_n^*} - \nu_\infty + er(h_{\hat{i}_n n}) - \nu_{j_n^*}$$

$$\leq \nu_{j_n^*} - \nu_\infty + \max \begin{cases} 2(er_{m_{j_n^* n}}(h_{\hat{i}_n n}) - er_{m_{j_n^* n}}(h_n') + \tau_n) \\ \hat{\mathcal{E}}_{\mathbb{C}_{j_n^*}}(m_{j_n^* n}, \delta_{j_n^*}) \end{cases}$$

$$\leq \nu_{j_n^*} - \nu_\infty + \max \begin{cases} 2(er_{\mathcal{L}_{j_n^* n} \cup Q_{j_n^* n}}(h_{\hat{i}_n n}) - er_{\mathcal{L}_{j_n^* n} \cup Q_{j_n^* n}}(h_{j_n^* n})) + \tau_n) \\ \hat{\mathcal{E}}_{\mathbb{C}_{j_n^*}}(m_{j_n^* n}, \delta_{j_n^*}) \end{cases}$$

The first inequality follows from Lemma 2.16. The second inequality is due to Lemma 2.20 (i.e., $j_n^* \geq i_n^*$). In this last line, we can let $\tau_n \to 0$, and using the definition of $\hat{i}_n$ show that it is at most

$$\nu_{j_n^*} - \nu_\infty + \max \left\{ 2\left(\frac{3}{2}\hat{\mathcal{E}}_{\mathbb{C}_{j_n^*}}(\mathcal{L}_{j_n^* n} \cup Q_{j_n^* n}, \delta_{j_n^*}; \mathcal{L}_{j_n^* n})\right), \hat{\mathcal{E}}_{\mathbb{C}_{j_n^*}}(m_{j_n^* n}, \delta_{j_n^*}) \right\}$$

$$= \nu_{j_n^*} - \nu_\infty + 3\hat{\mathcal{E}}_{\mathbb{C}_{j_n^*}}(m_{j_n^* n}, \delta_{j_n^*}) \qquad \text{(Lemma 2.17)}$$

$$\leq 3 \min_i \left(\nu_i - \nu_\infty + \hat{\mathcal{E}}_{\mathbb{C}_i}(m_{in}, \delta_i)\right) \qquad \text{(by definition of } j_n^*)$$

$$\leq 3 \min_i \left(\nu_i - \nu_\infty + \tilde{\mathcal{E}}_{\mathbb{C}_i}(m_{in}, \delta_i)\right) \qquad \text{(Lemma 2.16).}$$

$\square$

We are now ready for the proof of Theorems 2.13 and 2.14.

Proofs of Theorem 2.13 and Theorem 2.14. These theorems now follow directly from Lemmas 2.21 and 2.19. That is, Lemma 2.21 gives a bound in terms of the $\tilde{\mathcal{E}}$ quantities, holding on event $\bigcap_{i=1}^\infty E_{\mathbb{C}_i, \delta_i}$, and Lemma 2.19 bounds these $\tilde{\mathcal{E}}$ quantities as desired, on event $\bigcap_{i=1}^\infty H_{\mathbb{C}_i, \delta_i} \cap E_{\mathbb{C}_i, \delta_i}$. Noting that, by the union bound, $\mathbb{P}\left[\bigcap_{i=1}^\infty H_{\mathbb{C}_i, \delta_i} \cap E_{\mathbb{C}_i, \delta_i}\right] \geq 1 - \sum_{i=1}^\infty \delta_i \geq 1 - \delta$ completes the proof. $\square$

Define $\mathring{c} = \tilde{c} + 1$, $\mathring{D}(\epsilon) = \lim_{j\to\infty} diam(\mathbb{C}_j(\epsilon))$, and

$$\mathring{U}_{\mathbb{C}_i}(m, \epsilon, \delta_i) = \tilde{K}\left(\omega_{\mathbb{C}_i}(m, \mathring{D}(\mathring{c}\epsilon)) + \sqrt{\frac{s_m(\delta_i)\mathring{D}(\mathring{c}\epsilon)}{m}} + \frac{s_m(\delta_i)}{m}\right)$$

and

$$\mathring{\mathcal{E}}_{\mathbb{C}_i}(m, \delta_i) = \inf \left\{ \epsilon > 0 : \forall j \in \mathbb{Z}_\epsilon, \mathring{U}_{\mathbb{C}_i}(m, 2^j, \delta_i) \leq 2^{j-4} \right\}.$$

Lemma 2.22. *For any $m, i \in \mathbb{N}$,*

$$\tilde{\mathcal{E}}_{\mathbb{C}_i}(m, \delta_i) \leq \max \left\{ \mathring{\mathcal{E}}_{\mathbb{C}_i}(m, \delta_i), \nu_i - \nu_\infty \right\}.$$

Proof of Lemma 2.22. For $\epsilon > \nu_i - \nu_\infty$,

$$\tilde{U}_{\mathbb{C}_i}(m, \epsilon, \delta_i) = \tilde{K} \left(\phi_{\mathbb{C}_i}(m, \tilde{c}\epsilon) + \sqrt{\frac{s_m(\delta_i) diam(\mathbb{C}_i(\tilde{c}\epsilon))}{m}} + \frac{s_m(\delta_i)}{m} \right)$$

$$\leq \tilde{K} \left(\omega_{\mathbb{C}_i}(m, diam(\mathbb{C}_i(\tilde{c}\epsilon))) + \sqrt{\frac{s_m(\delta_i) diam(\mathbb{C}_i(\tilde{c}\epsilon))}{m}} + \frac{s_m(\delta_i)}{m} \right).$$

But $diam(\mathbb{C}_i(\tilde{c}\epsilon)) \leq \mathring{D}(\tilde{c}\epsilon + (\nu_i - \nu_\infty)) \leq \mathring{D}(\mathring{c}\epsilon)$, so the above line is at most

$$\tilde{K} \left(\omega_{\mathbb{C}_i}(m, \mathring{D}(\mathring{c}\epsilon)) + \sqrt{\frac{s_m(\delta_i) \mathring{D}(\mathring{c}\epsilon)}{m}} + \frac{s_m(\delta_i)}{m} \right) = \mathring{U}_{\mathbb{C}_i}(m, \epsilon, \delta_i).$$

In particular, this implies that

$$\begin{aligned}
\tilde{\mathcal{E}}_{\mathbb{C}_i}(m, \delta_i) = \quad & \inf \left\{ \epsilon > 0 : \forall j \in \mathbb{Z}_\epsilon, \tilde{U}_{\mathbb{C}_i}(m, 2^j, \delta_i) \leq 2^{j-4} \right\} \\
\leq \quad & \inf \left\{ \epsilon > (\nu_i - \nu_\infty) : \forall j \in \mathbb{Z}_\epsilon, \tilde{U}_{\mathbb{C}_i}(m, 2^j, \delta_i) \leq 2^{j-4} \right\} \\
\leq \quad & \inf \left\{ \epsilon > (\nu_i - \nu_\infty) : \forall j \in \mathbb{Z}_\epsilon, \mathring{U}_{\mathbb{C}_i}(m, 2^j, \delta_i) \leq 2^{j-4} \right\} \\
\leq \quad & \max \left\{ \inf \left\{ \epsilon > 0 : \forall j \in \mathbb{Z}_\epsilon, \mathring{U}_{\mathbb{C}_i}(m, 2^j, \delta_i) \leq 2^{j-4} \right\}, (\nu_i - \nu_\infty) \right\} \\
= \quad & \max \left\{ \mathring{\mathcal{E}}_{\mathbb{C}_i}(m, \delta_i), \nu_i - \nu_\infty \right\}.
\end{aligned}$$

$\square$

Proof of Theorem 2.15. By the same argument that lead to (2.10), we have that

$$\mathring{\mathcal{E}}_{\mathbb{C}_i}(m, \delta_i) \leq K_2 \frac{d_i \log \frac{mi}{\delta}}{m},$$

47

for some constant K_2 (depending on μ).

Now assume the event $\bigcap_{i=1}^{\infty} H_{\mathbb{C}_i,\delta_i} \cap E_{\mathbb{C}_i,\delta_i}$ occurs. In particular, Lemma 2.21 implies that $\forall i, n \in \mathbb{N}$,

$$er(\hat{h}_n) - \nu^* \le \min\left\{1, 3\min_{i\in\mathbb{N}}\left(2(\nu_i - \nu_\infty) + \mathring{\mathcal{E}}_{\mathbb{C}_i}(m_{in}, \delta_i)\right)\right\}$$

$$\le K_3 \min_{i\in\mathbb{N}}\left((\nu_i - \nu^*) + \min\left\{1, \frac{d_i \log \frac{m_{in} i}{\delta}}{m_{in}}\right\}\right),$$

for some constant K_3.

Now take $i \in \mathbb{N}$. The label request bound of Lemma 2.18, along with Lemma 2.22, implies that

$$\lfloor n/(2i^2) \rfloor \le \log \frac{8 m_{in}^2 i^2}{\delta} + K_4 \theta_i \left(2 + \int_1^{m_{in}-1} \max\left\{\nu_i - \nu^*, \frac{d_i \log \frac{xi}{\delta}}{x}\right\} dx\right)$$

$$\le K_5 \theta_i \max\left\{(\nu_i - \nu^*)m_{in}, d_i \log^2(m_{in}) \log \frac{i}{\delta}\right\}$$

Let $\gamma_i(n) = \sqrt{\frac{n}{i^2 \theta_i d_i \log \frac{i}{\delta}}}$. Then

$$\frac{d_i \log \frac{m_{in} i}{\delta}}{m_{in}} \le K_6 \left((\nu_i - \nu^*)\frac{1 + \gamma_i(n)}{\gamma_i(n)^2} + d_i \log \frac{i}{\delta}(1 + \gamma_i(n)) exp\left\{-c_2 \gamma_i(n)\right\}\right).$$

Thus,

$$\min\left\{1, \frac{d_i \log \frac{m_{in} i}{\delta}}{m_{in}}\right\} \le \min\left\{1, K_7\left((\nu_i - \nu^*) + d_i \log \frac{i}{\delta}(1 + \gamma_i(t)) exp\left\{-c_2\gamma_i(n)\right\}\right)\right\}.$$

The result follows from this by some simple algebra. $\qquad\square$

2.8 Time Complexity of Algorithm 2

It is worth making a few remarks about the time complexity of Algorithm 2 when used with the (2.6) threshold. Clearly the LEARN$_\mathbb{C}$ subroutine could be at least as computationally hard as empirical risk minimization (ERM) over $\mathbb{C}$. For most interesting hypothesis classes, this is known to be NP-Hard – though interestingly, there are some efficient special cases [e.g.,

Kalai, Klivans, Mansour, and Servedio, 2005]. Additionally, there is the matter of calculating $\hat{\mathcal{E}}_m(\delta; \mathbb{C}, \mathcal{L})$. The challenge here is due to the localization $\hat{\mathbb{C}}(\epsilon; \mathcal{L})$ in the empirical Rademacher process calculation and the empirical diameter calculation.

However, using a trick similar to that in Bartlett, Bousquet, and Mendelson [2005], we can calculate or bound these quantities via an efficient reduction to minimization of a *weighted* empirical error. That is, the only possibly difficult step in calculating $\hat{\phi}_m(\epsilon; \mathbb{C}, \mathcal{L})$ requires only that we identify $h_1 = \operatorname*{argmin}_{h \in \hat{\mathbb{C}}_m(\epsilon; \mathcal{L})} er_m(h, \xi)$ and $h_2 = \operatorname*{argmin}_{h \in \hat{\mathbb{C}}_m(\epsilon; \mathcal{L})} er_m(h, -\xi)$, where $er_m(h, \xi) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}[h(X_i) \neq \xi_i]$ and $er_m(h, -\xi)$ is the same but with $-\xi_i$. Similarly, letting $\hat{h}_{\mathcal{L}} = $ LEARN$_{\mathbb{C}}(\mathcal{L}, Q)$ for $\mathcal{L} \cup Q$ generated from the first m unlabeled examples, we can bound $\hat{D}_m(\epsilon; \mathbb{C}, \mathcal{L})$ within a factor of 2 by $2er_m(h', \hat{h}_{\mathcal{L}})$ where $h' = \operatorname*{argmin}_{h \in \hat{\mathbb{C}}_m(\epsilon; \mathcal{L})} er_m(h, -\hat{h}_{\mathcal{L}})$ and $er_m(f, g) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}[f(X_i) \neq g(X_i)]$. All that remains is to specify how this optimization for h_1, h_2, and h' can be performed. Taking the h_1 case for example, we can solve the optimization as follows. We find

$$\hat{h}_{(\lambda)} = \arg\min_{h \in \mathbb{C}} \sum_{i=1}^{m} \mathbb{1}[h(X_i) \neq \xi_i] + \sum_{(x,y) \in Q} \lambda \mathbb{1}[h(x) \neq y] + \sum_{(x,y) \in \mathcal{L}} 2 \max\{1, \lambda\} m \mathbb{1}[h(x) \neq y],$$

where λ is a Lagrange multiplier; we can calculate $\hat{h}_{(\lambda)}$ for $O(m^2)$ values of λ in a discrete grid, and from these choose the one with smallest $er_m(\hat{h}_{(\lambda)}, \xi)$ among those with $er_{\mathcal{L} \cup Q}(\hat{h}_{(\lambda)}) - er_{\mathcal{L} \cup Q}(\hat{h}_{\mathcal{L}}) \leq \epsilon$. The third term guarantees the solution satisfies $er_{\mathcal{L}}(\hat{h}_{(\lambda)}) = 0$, while the value of λ specifies the trade-off between $er_{\mathcal{L} \cup Q}(\hat{h}_{(\lambda)})$ and $er_m(\hat{h}_{(\lambda)}, \xi)$. The calculation for h_2 and h' is analogous. Additionally, we can clearly formulate the LEARN subroutine as such a weighted ERM problem as well.

For each of these weighted ERM problems, a further polynomial reduction to (unweighted) empirical risk minimization is possible. In particular, we can replicate the examples a number of times proportional to the weights, generating an ERM problem on $O(m^2)$ examples. Thus, for processing any finite number of unlabeled examples m, the time complexity of Algorithm 2 (substituting the above 2-approximation for $\hat{D}_m(\epsilon; \mathbb{C}, \mathcal{L})$, which only changes constant factors in the results of Section 2.3.4) should be no more than a polynomial factor worse than the time

complexity of empirical risk minimization with $\mathbb{C}$, for the worst case over all samples of size $O(m^2)$.

2.9   A Refined Analysis of PAC Learning Via the Disagreement Coefficient

Throughout this section, we will work in $\mathcal{Realizable}(\mathbb{C})$ and denote $\mathcal{D} = \mathcal{D}_{XY}[\mathcal{X}]$. In particular, there is always a target function $f \in \mathbb{C}$ with $er(f) = 0$.

Note that the known general upper bound for this problem is that, if the VC dimension of $\mathbb{C}$ is d, then with probability $1 - \delta$, every classifier in $\mathbb{C}$ consistent with n random samples has error rate at most

$$4 \frac{d \ln(2en/d) + \ln(4/\delta)}{n}. \tag{2.13}$$

This is due to Vapnik [1982]. There is a slightly different bound (for a different learning strategy) of

$$\propto \frac{d \log(1/\delta)}{n} \tag{2.14}$$

proven by Haussler, Littlestone, and Warmuth [1994]. It is also known that one cannot get a distribution-free bound smaller than

$$\propto \frac{d + \log(1/\delta)}{n}$$

for any concept space [Vapnik, 1982]. The question we are concerned with here is deriving upper bounds that are closer to this lower bound than either (2.13) or (2.14) in some cases.

For our purposes, throughout this section we will take $r_0 = \frac{d + \log(1/\delta)}{n}$ in the definition of the disagreement coefficient. In particular, recall that $\theta_f \leq \frac{1}{r_0}$ always, and this will imply a fallback guarantee no worse than those above for our analysis below. However, it is sometimes much smaller, or even constant, in which case our analysis here may be better than those mentioned above.

50

2.9.1 Error Rates for Any Consistent Classifier

For simplicity and to focus on the nontrivial cases, the results in this section will be stated for the case where $\mathbb{P}(DIS(\mathbb{C})) > 0$. The $\mathbb{P}(DIS(\mathbb{C})) = 0$ case is trivial, since every $h \in \mathbb{C}$ has $er(h) = 0$ there.

Theorem 2.23. *Let d be the VC dimension of concept space $\mathbb{C}$, and let*

$V_n = \{h \in \mathbb{C} : \forall i \leq n, h(x_i) = f(x_i)\}$, *where $f \in \mathbb{C}$ is the target function (i.e., $er(f) = 0$),*

and $(x_1, x_2, \ldots, x_n) \sim \mathcal{D}^n$ is a sequence of i.i.d. training examples. Then for any $\delta \in (0, 1)$,

with probability $\geq 1 - \delta$, $\forall h \in V_n$,

$$er(h) \leq \frac{24}{n} \left(d \ln(880\theta_f) + \ln \frac{12}{\delta} \right). \tag{2.15}$$

Proof. Since $\mathbb{P}(DIS(\mathbb{C})) > 0$ by assumption, $\theta_f > 0$ (and $d > 0$ also follows). As above, let $V_m = \{h \in \mathbb{C} : \forall i \leq m, h(x_i) = f(x_i)\}$, and define $radius(V_m) = \sup_{h \in V_m} er(h)$. We will prove the result by induction on n. As a base case, note that the result clearly holds for $n \leq d$, as we always have $er(h) \leq 1$.

Now suppose $n \geq d + 1 \geq 2$, and suppose the result holds for any $m < n$; in particular, consider $m = \lfloor n/2 \rfloor$. Thus, for any $\delta \in (0, 1)$, with probability $\geq 1 - \delta/3$,

$$radius(V_m) \leq \frac{24}{m} \left(d \ln(880\theta_f) + \ln \frac{36}{\delta} \right).$$

Note that $r_n < r_m$, so we can take this inequality to hold for the θ_f defined with r_n as well. If $radius(V_m) \leq r_n$, then we clearly have (2.15) (and (2.16) below) since $radius(V_n) \leq radius(V_m)$, so suppose $radius(V_m) > r_n$. Likewise, if $\mathbb{P}(DIS(V_m)) < \frac{8}{m} \ln \frac{3}{\delta} \leq \frac{24}{n} \ln \frac{3}{\delta}$, then (2.15) is valid (as is (2.16) below) since $radius(V_n) \leq radius(V_m) \leq \mathbb{P}(DIS(V_m))$. Otherwise, by a Chernoff bound, with probability $\geq 1 - \delta/3$, we have

$$|\{x_{m+1}, x_{m+2}, \ldots, x_n\} \cap DIS(V_m)| \geq \mathbb{P}(DIS(V_m))\lceil n/2 \rceil/2 =: N.$$

When this is the case, the first N samples in $\{x_{m+1}, x_{m+2}, \ldots, x_n\} \cap DIS(V_m)$ represent an iid sample from the conditional given $DIS(V_m)$, so that (2.13) tells us that given this event, with

probability $\geq 1 - \delta/3$,

$$radius(V_n) = \mathbb{P}(DIS(V_m))radius(V_n|DIS(V_m))$$

$$\leq \mathbb{P}(DIS(V_m))\frac{4}{N}\left(d\ln\frac{2eN}{d} + \ln\frac{12}{\delta}\right) \leq \frac{16}{n}\left(d\ln\frac{2e\mathbb{P}(DIS(V_m))n}{4d} + \ln\frac{12}{\delta}\right)$$

$$\leq \frac{16}{n}\left(d\ln\frac{e\theta_f radius(V_m)n}{2d} + \ln\frac{12}{\delta}\right).$$

Applying the inductive hypothesis for $radius(V_m)$, combined with a union bound over these 3 failure events (each of probability $\delta/3$), we have that with probability $\geq 1 - \delta$,

$$radius(V_n) \leq \frac{16}{n}\left(d\ln\left(48e\theta_f\left(\ln(880\theta_f) + \frac{1}{d}\ln\frac{36}{\delta}\right)\right) + \ln\frac{12}{\delta}\right). \qquad (2.16)$$

If $d \geq \frac{1}{e}\ln\frac{12}{\delta}$, then the right side of (2.16) is at most

$$\frac{16}{n}\left(d\ln\left(\theta_f 48e\ln(880 \cdot 3 \cdot e^e\theta_f)\right) + \ln\frac{12}{\delta}\right)$$

$$\leq \frac{16}{n}\left(d\ln\left(\theta_f 48e\ln(40008\theta_f)\right) + \ln\frac{12}{\delta}\right)$$

$$\leq \frac{16}{n}\left(d\ln\left(26099\theta_f^{3/2}\right) + \ln\frac{12}{\delta}\right) \leq \frac{24}{n}\left(d\ln(880\theta_f) + \ln\frac{12}{\delta}\right).$$

Otherwise $d < \frac{1}{e}\ln\frac{12}{\delta}$, so that the right side of (2.16) is at most

$$\frac{16}{n}\left(d\ln\left(\theta_f 48e\ln(880 \cdot 3\theta_f)\frac{1}{d}\ln\frac{12}{\delta}\right) + \ln\frac{12}{\delta}\right)$$

$$\leq \frac{16}{n}\left(d\ln\left(6705\theta_f^{3/2}\right) + d\ln\left(\frac{1}{d}\ln\frac{12}{\delta}\right) + \ln\frac{12}{\delta}\right)$$

$$\leq \frac{24}{n}\left(d\ln(356\theta_f) + \frac{2}{3}\left(\frac{1}{e} + 1\right)\ln\frac{12}{\delta}\right) \leq \frac{24}{n}\left(d\ln(880\theta_f) + \ln\frac{12}{\delta}\right).$$

The theorem now follows by the principle of induction. $\qquad \square$

With this result in hand, we can immediately get some interesting results, such as the following corollary.

Corollary 2.24. *Suppose $\mathbb{C}$ is the space of linear separators in d dimensions that pass through the origin, and suppose the distribution is uniform on the surface of the origin-centered unit sphere. Then with probability $\geq 1 - \delta$, any $h \in \mathbb{C}$ consistent with the n i.i.d. training examples has (for some finite universal c)*

$$er(h) \leq c \frac{d \log d + \log \frac{1}{\delta}}{n}.$$

Proof. [Hanneke, 2007b] proves that $\sup\limits_{f \in \mathbb{C}} \theta_f \leq \pi \sqrt{d}$ for this problem. □

This improves over the best previously known bound for consistent classifiers for this problem in its dependence on n, which was $\propto \frac{d\sqrt{\log(n/d)} + \log(1/\delta)}{n}$ [Li and Long, 2007] (though we picked up an extra $\log d$ factor in the process).

2.9.2   Specializing to Particular Algorithms

The above analysis is for arbitrary algorithms that select a classifier consistent with the training data. However, we can modify the disagreement coefficient to be more interesting for more specific algorithms. Specifically, suppose there are sets $\mathbb{C}_f$ such that with high probability algorithm $\mathcal{A}$ will output a classifier in $\mathbb{C}_f$ when f is the target function. Then we only need to worry about the regions of disagreement within these $\mathbb{C}_f$ sets, which may be significantly smaller than within the full space $\mathbb{C}$.

To give a concrete example, consider the Closure algorithm: output the $h \in \mathbb{C}$ with smallest $\mathbb{P}(h(X) = +1)$ that is consistent with the data. For intersection-closed $\mathbb{C}$, the sets are $\mathbb{C}_f = \{h \in \mathbb{C} : h(x) = +1 \Rightarrow f(x) = +1\}$. So effectively, this becomes our concept space, and the disagreement coefficient of f with respect to $\mathbb{C}_f$ and $\mathcal{D}$ can be significantly smaller than it is with respect to the full space $\mathbb{C}$. For instance, if $\mathbb{C}$ is axis-aligned rectangles, then the disagreement coefficient of any $f \in \mathbb{C}$ with respect to $\mathbb{C}_f$ and $\mathcal{D}$ is at most d. This implies a bound

$$\propto \frac{d \log d + \log(1/\delta)}{n}.$$

We already have better bounds than this for using Closure with this concept space. However, if the d upper bound on disagreement coefficient with respect to $\mathbb{C}_f$ is true for *general* intersection-closed spaces $\mathbb{C}$, this would match the best known bounds for general intersection-closed spaces [Auer and Ortner, 2004].

Chapter 3

Significance of the Verifiable/Unverifiable Distinction in Realizable Active Learning

This chapter describes and explores a new perspective on the label complexity of active learning in the fixed-distribution realizable case. In many situations where it was generally thought that active learning does not help, we show that active learning does help in the limit, often with exponential improvements in label 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 labels asymptotically smaller than this. These new insights arise from a subtle variation on the traditional definition of label complexity, not previously recognized in the active learning literature.

Remark 3.1. *The results in this chapter are taken from [Balcan, Hanneke, and Wortman, 2008], joint work with Maria-Florina Balcan and Jennifer Wortman.*

3.1 Introduction

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 [Balcan et al., 2006, 2007, Cohn et al., 1994, Dasgupta, 2004, 2005, Dasgupta et al., 2005, 2007, Hanneke, 2007a,b]. 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 [Balcan et al., 2006, 2007, Dasgupta, 2005, Dasgupta et al., 2005, 2007]. 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 label complexities have indicated that perhaps active learning does not help at all [Dasgupta, 2005].

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 label 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
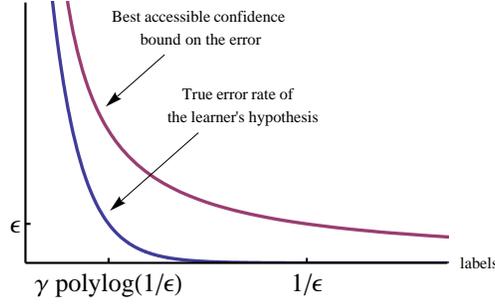
Figure 3.1: Active learning can often achieve exponential improvements, though in many cases the amount of improvement cannot be detected from information available to the learning algorithm. Here γ may be a target-dependent constant.

previous literature, active learning can achieve exponential[1] improvements over passive learning with respect to dependence on ϵ. This situation is characterized in Figure 3.1.

To our knowledge, this is the first work to address this subtle point in the context of active learning. Though several previous papers have studied bounds on this latter type of label complexity [Castro and Nowak, 2007, Dasgupta et al., 2005, 2007], their results were *no stronger* than the results one could prove in the traditional analysis. As such, it seems this large gap between the two types of label complexities has gone unnoticed until now.

3.1.1 A Simple Example: Intervals

To get some intuition about when these types of label complexity are different, consider the following example. Suppose that $\mathbb{C}$ is the class of all intervals over $[0, 1]$ and $\mathcal{D}$ is 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 $[0, 2\epsilon], [2\epsilon, 4\epsilon], \ldots$, requiring $\Omega(1/\epsilon)$ total label requests.

---

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

However, no matter what the target function is, we can *find* an ϵ-good classifier with only a logarithmic label complexity via the following extremely simple 2-phase learning algorithm. The algorithm will be allowed to make t label requests, and then we will find a value of t that is sufficiently large to guarantee learning. We start with a large $(\Omega(2^t))$ 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 either observe a $+1$ label, at which point we enter the second phase, or we use all t label requests. 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. Once we use all t label requests, we output a smallest interval consistent with the observed 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, so any $t \geq 0$ suffices in this case. 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 $O(\log(1/\epsilon))$ additional queries are required to run the binary search to reach error rate ϵ, it suffices to have $t \geq O(1/w + \log(1/\epsilon)) = O(\log(1/\epsilon))$. So in general, the label complexity is at worst $O(\log(1/\epsilon))$. Thus, we see a sharp distinction between the label complexity required to *find* a good classifier (logarithmic) and the label 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.

3.1.2 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 label complexity. The traditional definition, which we refer to as *verifiable label 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 *label 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 label complexity asymptotically smaller than the label complexity of passive learning for nontrivial targets. A simple corollary of this is that finite VC dimension implies $o(1/\epsilon)$ active learning label 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 3.11). 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.

3.2 Background and Notation

In various places throughout this chapter, we will need notation for a *countable dense subset* of a hypothesis class V. For any set of classifiers V, we will denote by $\tilde{V}$ a countable (or possibly finite) subset of V s.t. $\forall \alpha > 0$, $\forall h \in V$, $\exists h' \in \tilde{V}$ with $\mathbb{P}_{\mathcal{D}_{XY}[\mathcal{X}]}(h(X) \neq h'(X)) \leq \alpha$. Such a set is guaranteed to exist under mild conditions; in particular, finite VC dimension suffices to guarantee its existence. We introduce this notion to avoid certain degenerate behaviors, such as when $DIS(B(h,0)) = \mathcal{X}$. For instance, the hypothesis class of classifiers on the $[0,1]$ interval that label exactly one point positive has this property under any nonatomic density function.

Since all of the results in this chapter are for the fixed-distribution realizable case, it will be convenient to introduce the following short-hand notation.

Definition 3.1. *A function $\Lambda(\epsilon, \delta, h^*)$ is a* label complexity *for a pair $(\mathbb{C}, \mathcal{D})$ if there exists an active learning algorithm $\mathcal{A}$ achieving label complexity $\Lambda(\epsilon, \delta, \mathcal{D}_{XY}) = \Lambda(\epsilon, \delta, h^*_{\mathcal{D}_{XY}})$ for all $\mathcal{D}_{XY} \in \mathcal{R}ealizable(\mathbb{C}, \mathcal{D})$, where $\mathcal{D}$ is a distribution over $\mathcal{X}$ and $h^*_{\mathcal{D}_{XY}}$ is the target function under $\mathcal{D}_{XY}$.*

Definition 3.2. *A function $\Lambda(\epsilon, \delta, h^*)$ is a* verifiable label complexity *for a pair $(\mathbb{C}, \mathcal{D})$ if there exists an active learning algorithm $\mathcal{A}$ achieving verifiable label complexity $\Lambda(\epsilon, \delta, \mathcal{D}_{XY}) = \Lambda(\epsilon, \delta, h^*_{\mathcal{D}_{XY}})$ for all $\mathcal{D}_{XY} \in \mathcal{R}ealizable(\mathbb{C}, \mathcal{D})$, where $\mathcal{D}$ is a distribution over $\mathcal{X}$ and $h^*_{\mathcal{D}_{XY}}$ is the target function under $\mathcal{D}_{XY}$.*

Let us take a moment to reflect on the difference between the two definitions of label complexity: namely, verifiable and unverifiable. The distinction may appear quite subtle. Both definitions allow the label complexity to depend both on the target function and on the input distribution. The only distinction is whether or not there is an *accessible guarantee* or *confidence bound* 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. As an illustration of this distinction, consider again the problem of learning intervals. As described above, if the target h^* is an interval of width w, then after seeing $O(1/w + \log(1/\epsilon))$ labels, with

high probability it is possible for an algorithm to *guarantee* that it can output a function with error less than ϵ. In this case, for sufficiently small ϵ, the verifiable label complexity $\Lambda(\epsilon, \delta, h^*)$ is proportional to $\log(1/\epsilon)$. However, if h^* is the all-negative function, then the verifiable label complexity is at least proportional to $1/\epsilon$ for *all* values of ϵ because *a high-confidence guarantee can never be made* without observing $\Omega(1/\epsilon)$ labels; for completeness, a formal proof of this fact is included in Section 3.7. In contrast, as we have seen, the label complexity is $O(\log(1/\epsilon))$ for *any* target in the class of intervals when no such guarantee is required.

A common alternative formulation of verifiable label 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 [Dasgupta, 2005]. This alternative definition is almost equivalent (an algorithm for either definition can be modified to fit the other definition without significant loss in the verifiable label complexity values), 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]

3.2.1 The Verifiable Label Complexity

To date, there has been a significant amount of work studying the verifiable label complexity (though typically under the aforementioned alternative formulation). It is clear from standard results in passive learning that verifiable label 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 la-

---

[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 [Castro and Nowak, 2007]), 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 label complexity up to constants.

bel complexity *smaller* than this, and in particular, when the verifiable label complexity is a polylogarithmic function of $1/\epsilon$ (representing exponential improvements over passive learning).

As discussed in previous chapters, there have been a few quantities proposed to measure the verifiable label complexity of active learning on any given concept class and distribution. Dasgupta's *splitting index* [Dasgupta, 2005], 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 the *disagreement coefficient* [Hanneke, 2007b], defined in Chapter 2.

The disagreement coefficient is often a useful quantity for analyzing the verifiable label complexity of active learning algorithms. For example, as we saw in Chapter 2, Algorithm 0 achieves a verifiable label complexity at most $\theta_{h^*} d \cdot \mathrm{polylog}(1/(\epsilon\delta))$ when run with hypothesis class $\mathbb{C}$ for target function $h^* \in \mathbb{C}$. We will use it in several of the results below. In all of the relevant results of this chapter, we will simply take $r_0 = 0$ in the definition of the disagreement coefficient.

We will see that both the disagreement coefficient and splitting index are also useful quantities for analyzing unverifiable label complexities, though their use in that case is less direct.

3.2.2 The True Label Complexity

This chapter focuses on situations where true label complexities are significantly smaller than verifiable label complexities. In particular, we show that many common pairs $(\mathbb{C}, \mathcal{D})$ have label 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 label complexity [Dasgupta, 2004, 2005, Freund et al., 1997, Hanneke, 2007a]. 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 [Antos and Lugosi, 1998].

Definition 3.3. *We say that* $(\mathbb{C}, \mathcal{D})$ *is* actively learnable at an exponential rate *if there exists an active learning algorithm achieving label complexity*

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

for all $h^* \in \mathbb{C}$*, where* γ_{h^*} *is a finite constant that may depend on* h^* *and* $\mathcal{D}$ *but is independent of* ϵ *and* δ.

3.3 Strict Improvements of Active Over Passive

In this section, we describe conditions under which active learning can achieve a label 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 3.4. *A function* $\Lambda(\epsilon, \delta, h^*)$ *is a* passive learning *label complexity for a pair* $(\mathbb{C}, \mathcal{D})$ *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* $h_{t,\delta}$*, such that for any target function* $h^* \in \mathbb{C}, \epsilon \in (0, 1/2), \delta \in (0, 1)$*, for any* $t \geq \Lambda(\epsilon, \delta, h^*)$,

$$\mathbb{P}_{\mathcal{D}}(er(h_{t,\delta}) \leq \epsilon) \geq 1 - \delta.$$

Thus, a passive learning label complexity corresponds to a restriction of an active learning label 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 $O\left(1/\epsilon\right)$ passive learning label complexity [Haussler et al., 1994]. Furthermore, this is often (though not always) tight, in the sense that for any passive algorithm, there exist targets for which the corresponding passive learning label complexity is $\Omega\left(1/\epsilon\right)$ [Antos and Lugosi, 1998]. The following theorem states that for any passive learning label complexity, there exists an achievable active learning label complexity with a strictly slower asymptotic rate of growth. Its proof is included in Section 3.11.

Remark 3.2. *This result is superceded by a stronger result in Chapter 4; however, the result in*

Chapter 4 is proven for a different algorithm, so that Theorem 3.5 is not entirely redundant. I have therefore chosen to include the result, since the construction of the algorithm may be of independent interest, even if the stated theorem is itself weaker than later results.

Theorem 3.5. *Suppose $\mathbb{C}$ has finite VC dimension, and let $\mathcal{D}$ be any distribution on $\mathcal{X}$. For any passive learning label complexity $\Lambda_p(\epsilon, \delta, h)$ for $(\mathbb{C}, \mathcal{D})$, there exists an active learning algorithm achieving a label complexity $\Lambda_a(\epsilon, \delta, h)$ such that, for all $\delta \in (0, 1/4)$ and targets $h^* \in \mathbb{C}$ for which $\Lambda_p(\epsilon, \delta, h^*) = \omega(1)$,*

$$\Lambda_a(\epsilon, \delta, h^*) = o\left(\Lambda_p(\epsilon/4, \delta, h^*)\right).$$

In particular, this implies the following simple corollary.

Corollary 3.6. *For any $\mathbb{C}$ with finite VC dimension, and any distribution $\mathcal{D}$ over $\mathcal{X}$, there is an active learning algorithm that achieves a label complexity $\Lambda(\epsilon, \delta, h^*)$ such that for $\delta \in (0, 1/4)$,*

$$\Lambda(\epsilon, \delta, h^*) = o\left(1/\epsilon\right)$$

for all targets $h \in \mathbb{C}$.

Proof. Let d be the VC dimension of $\mathbb{C}$. The passive learning algorithm of Haussler, Littlestone & Warmuth [Haussler et al., 1994] is known to achieve a label complexity no more than $(kd/\epsilon)\log(1/\delta)$, for some universal constant $k < 200$. Applying Theorem 3.5 now implies the result. $\qquad\square$

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

3.4   Decomposing Hypothesis Classes

Let us return once more to the simple example of learning the class of intervals over $[0, 1]$ under the uniform distribution. As discussed above, it is well known that the verifiable label complexity of the all-negative classifier in this class is $\Omega(1/\epsilon)$. However, consider the more limited class $\mathbb{C}' \subset \mathbb{C}$ containing only the intervals h of width w_h strictly greater than 0. Using the simple algorithm described in Section 3.1.1, this restricted class can be learned with a (verifiable) label complexity of only $O(1/w_h + \log(1/\epsilon))$. Furthermore, the remaining set of classifiers $\mathbb{C}'' = \mathbb{C} \setminus \mathbb{C}'$ consists of only a single function (the all-negative classifier) and thus can be learned with verifiable label complexity 0. Here we have that $\mathbb{C}$ can be decomposed into two subclasses $\mathbb{C}'$ and $\mathbb{C}''$, where both $(\mathbb{C}', \mathcal{D})$ and $(\mathbb{C}'', \mathcal{D})$ are learnable at an exponential rate. It is natural to wonder if the existence of such a decomposition is enough to imply that $\mathbb{C}$ itself is learnable at an exponential rate.

More generally, suppose that we are given a distribution $\mathcal{D}$ and a hypothesis class $\mathbb{C}$ such that we can construct a sequence of subclasses $\mathbb{C}_i$ with label complexity $\Lambda_i(\epsilon, \delta, h)$, with $\mathbb{C} = \cup_{i=1}^{\infty} \mathbb{C}_i$. Thus, if we knew *a priori* that the target h^* was a member of subclass $\mathbb{C}_i$, it would be straightforward to achieve $\Lambda_i(\epsilon, \delta, h^*)$ label complexity. It turns out that it is possible to learn *any* target h^* in *any* class $\mathbb{C}_i$ with label complexity only $O(\Lambda_i(\epsilon/2, \delta/2, h^*))$, even without knowing which subclass the target belongs to in advance. This can be accomplished by using a simple aggregation algorithm, such as the one given below. Here a set of active learning algorithms (for example, multiple instances of Dasgupta's splitting algorithm [Dasgupta, 2005] or CAL) are run on individual subclasses $\mathbb{C}_i$ in parallel. The output of one of these algorithms is selected according to a sequence of comparisons.

Using this algorithm, we can show the following label complexity bound. The proof appears in Section 3.8.

Algorithm 1 Algorithm 4 : The Aggregation Procedure. Here it is assumed that $\mathbb{C} = \cup_{i=1}^{\infty} \mathbb{C}_i$, and that for each i, A_i is an algorithm achieving label complexity at most $\Lambda_i(\epsilon, \delta, h)$ for the pair $(\mathbb{C}_i, \mathcal{D})$. Both the main aggregation procedure and each algorithm A_i take a number of labels t and a confidence parameter 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 $\mathbb{P}_{\mathcal{D}}(h_i(x) \neq h_j(x)) > 0$ **then**

        Let R_{ij} be the first $\lceil 72 \ln(4k/\delta) \rceil$ elements x in the sequence $\{x_{2n}\}_{n=1}^{\infty}$ s.t. $h_i(x) \neq h_j(x)$

        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 $i = \operatorname*{argmin}_{i \in \{1,2,\ldots,k\}} \max_{j \in \{1,2,\ldots,k\}} m_{ij}$

Theorem 3.7. *For any distribution $\mathcal{D}$, let $\mathbb{C}_1, \mathbb{C}_2, \ldots$ be a sequence of classes such that for each i, the pair $(\mathbb{C}_i, \mathcal{D})$ has label complexity at most $\Lambda_i(\epsilon, \delta, h)$ for all $h \in \mathbb{C}_i$. Let $\mathbb{C} = \cup_{i=1}^{\infty} \mathbb{C}_i$. Then $(\mathbb{C}, \mathcal{D})$ has a label complexity at most*

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

for any $h \in \mathbb{C}$. In particular, Algorithm 4 achieves this when given as input the algorithms A_i that each achieve label complexity $\Lambda_i(\epsilon, \delta, h)$ on class $(\mathbb{C}_i, \mathcal{D})$.

A particularly interesting implication of Theorem 3.7 is that the ability to decompose $\mathbb{C}$ into a sequence of classes $\mathbb{C}_i$ with each pair $(\mathbb{C}_i, \mathcal{D})$ learnable at an exponential rate is enough to imply that $(\mathbb{C}, \mathcal{D})$ is also learnable at an exponential rate. Since the *verifiable* label complexity of active learning has received more attention and is therefore better understood, it is often be useful to apply this result when there exist known bounds on the verifiable label complexity; the approach loses nothing in generality, as suggested by the following theorem. The proof of this theorem. is included in Section 3.9.

Theorem 3.8. *For any $(\mathbb{C}, \mathcal{D})$ learnable at an exponential rate, there exists a sequence $\mathbb{C}_1, \mathbb{C}_2, \ldots$ with $\mathbb{C} = \cup_{i=1}^{\infty} \mathbb{C}_i$, and a sequence of active learning algorithms $A_1, A_2, \ldots$ such that the algorithm A_i achieves* verifiable *label complexity at most $\gamma_i polylog_i (1/(\epsilon\delta))$ for the pair $(\mathbb{C}_i, \mathcal{D})$, where γ_i is a constant independent of ϵ and δ. In particular, the aggregation algorithm (Algorithm 4) achieves exponential rates when used with these algorithms.*

Note that decomposing a given $\mathbb{C}$ into a sequence of $\mathbb{C}_i$ subsets that have good verifiable label complexities is not always a simple task. One might be tempted to think a simple decomposition based on increasing values of verifiable label complexity with respect to $(\mathbb{C}, \mathcal{D})$ 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 $(\mathbb{C}, \mathcal{D})$ to construct a good decomposition. We have included in Section 3.10 a simple heuristic approach that can be quite effective, and in particular yields good label complexities for every $(\mathbb{C}, \mathcal{D})$ described in Section 3.5.

Since it is more abstract and allows us to use known active learning algorithms as a black box, we frequently rely on the decompositional view introduced here throughout the remainder of the chapter.

3.5 Exponential Rates

The results in Section 3.3 tell us that the label complexity of active learning can be made strictly superior to any passive learning label complexity when the VC dimension is finite. We now ask

how much better that label 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* label complexity.

3.5.1 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 label complexity is actually $O(1)$.

Clearly if $|\mathcal{X}| < \infty$ or $|\mathbb{C}| < \infty$, we can always achieve exponential rates. In the former case, we may simply request the label of every x in the support of $\mathcal{D}$, and thereby perfectly identify the target. The corresponding $\gamma = |\mathcal{X}|$. In the latter case, Algorithm 0 can achieve exponential learning with $\gamma = |\mathbb{C}|$ since each queried label will reduce the size of the version space by at least one.

Less obvious is the fact that a similar argument can be applied to any *countably infinite* hypothesis class $\mathbb{C}$. In this case we can impose an ordering $h_1, h_2, \cdots$ over the classifiers in $\mathbb{C}$, and set $\mathbb{C}_i = \{h_i\}$ for all i. By Theorem 3.7, applying the aggregation procedure to this sequence yields an algorithm with label complexity $\Lambda(\epsilon, \delta, h_i) = 2i^2 \lceil 72 \ln(4i/\delta) \rceil = O(1)$.

3.5.2 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* label complexity is $\Omega(1/\epsilon)$. As we see in Section 3.5.3, 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 $\mathbb{C}^{(k)}$

denote the class of unions of at most k intervals. In other words, $\mathbb{C}^{(k)}$ 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, define $\mathbb{C}_1$ to be the set containing the all-negative function along with any functions that are equivalent given the distribution $\mathcal{D}$. Formally,

$$\mathbb{C}_1 = \{ h \in \mathbb{C}^{(k)} : \mathbb{P}(h(X) = +1) = 0 \} \ .$$

Clearly $\mathbb{C}_1$ has verifiable label complexity 0. For $i = 2, 3, \ldots, k + 1$, let $\mathbb{C}_i$ be the set containing all functions that can be represented as unions of $i - 1$ intervals but cannot be represented as unions of fewer intervals. More formally, we can inductively define each $\mathbb{C}_i$ as

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

For $i > 1$, within each subclass $\mathbb{C}_i$, for each $h \in \mathbb{C}_i$ the disagreement coefficient wrt $\tilde{\mathbb{C}}_i$ is bounded by something proportional to $k + 1/w(h)$, where $w(h)$ is the weight of the smallest positive or negative interval with nonzero weight. Thus running Algorithm 0 with $\tilde{\mathbb{C}}_i$ achieves polylogarithmic (verifiable) label complexity for any $h \in \mathbb{C}_i$. Since $\mathbb{C}^{(k)} = \cup_{i=1}^{k+1} \mathbb{C}_i$, by Theorem 3.7, $\mathbb{C}^{(k)}$ is learnable at an exponential rate.

Ordinary Binary Classification Trees: Let $\mathcal{X}$ be the cube $[0, 1]^n$, $\mathcal{D}$ be the uniform distribution on $\mathcal{X}$, and $\mathbb{C}$ be the class of binary decision trees using a finite number of axis-parallel splits (see e.g., Devroye et al. [Devroye et al., 1996], Chapter 20). In this case, in the same spirit as the previous example, we let $\mathbb{C}_i$ be the set of decision trees in $\mathbb{C}$ distance zero from a tree with i leaf nodes, not contained in any $\mathbb{C}_j$ for $j < i$. For any i, the disagreement coefficient for any $h \in \mathbb{C}_i$ (with respect to $(\tilde{\mathbb{C}}_i, \mathcal{D})$) is a finite constant, and we can choose $\tilde{\mathbb{C}}_i$ to have finite VC dimension, so each $(\mathbb{C}_i, \mathcal{D})$ is learnable at an exponential rate (by running Algorithm 0 with $\tilde{\mathbb{C}}_i$). By Theorem 3.7, $(\mathbb{C}, \mathcal{D})$ is learnable at an exponential rate.

Linear Separators

> **Theorem 3.9.** *Let $\mathbb{C}$ be the concept class of linear separators in n dimensions, and let $\mathcal{D}$ be the uniform distribution over the surface of the unit sphere. The pair $(\mathbb{C}, \mathcal{D})$ is learnable at an exponential rate.*

Proof. 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. Let $\mathbb{C}_1$ be the set containing only the separators h with $\lambda(h) = 0$, let $\mathbb{C}_2 = \{h \in \mathbb{C} : \lambda(h) = 1/2\}$, and let $\mathbb{C}_3 = \mathbb{C} \setminus (\mathbb{C}_1 \cup \mathbb{C}_2)$. As before, we can use a black box active learning algorithm such as CAL to learn within the class $\mathbb{C}_3$. To prove that we indeed get the desired exponential rate of active learning, we show that the disagreement coefficient of any separator $h \in \mathbb{C}_3$ with respect to $(\mathbb{C}_3, \mathcal{D})$ is finite. The results concerning Algorithm 0 from Chapter 2 then immediately imply that $\mathbb{C}_3$ is learnable at an exponential rate. Since $\mathbb{C}_1$ trivially has label complexity 1, and $(\mathbb{C}_2, \mathcal{D})$ is known to be learnable at an exponential rate [e.g., Balcan, Broder, and Zhang, 2007, Dasgupta, 2005, Dasgupta, Kalai, and Monteleoni, 2005, Hanneke, 2007b] combined with Theorem 3.7, this would imply the result.

Below, we will restrict the discussion to hypotheses in $\mathbb{C}_3$, which will be implicit in notation such as $B(h, r)$, etc. First note that, to show $\theta_h < \infty$, it suffices to show that

$$\lim_{r \to 0} \frac{\mathbb{P}(DIS(B(h, r)))}{r} < \infty, \tag{3.1}$$

so we will focus on this.

For any h, there exists $r_h > 0$ s.t. $\forall h' \in B(h, r), \mathbb{P}(h'(X) = +1) \leq 1/2 \Leftrightarrow \mathbb{P}(h(X) = +1) \leq 1/2$, or in other words the minority class is the same among all $h' \in B(h, r)$. Now consider any $h' \in B(h, r)$ for $0 < r < \min\{r_h, \lambda(h)/2\}$. Clearly $\mathbb{P}(h(X) \neq h'(X)) \geq |\lambda(h) - \lambda(h')|$. Suppose $h(x) = sign(w \cdot x + b)$ and $h'(x) = sign(w' \cdot x + b')$ (where, without loss, we assume $\|w\| = 1$), and $\alpha(h, h') \in [0, \pi]$ is the angle between w and w'. If $\alpha(h, h') = 0$ or if the minority regions of h and h' do not intersect, then clearly $\mathbb{P}(h(X) \neq h'(X)) \geq \frac{2\alpha(h,h')}{\pi} \min\{\lambda(h), \lambda(h')\}$. Otherwise, consider the classifiers $\bar{h}(x) = sign(w \cdot x + \bar{b})$ and $\bar{h}'(x) =$
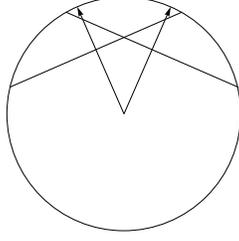
Figure 3.2: Projection of $\bar{h}$ and $\bar{h}'$ into the plane defined by w and w'.

$sign(w' \cdot x + \bar{b}')$, where $\bar{b}$ and $\bar{b}'$ are chosen s.t. $\mathbb{P}(\bar{h}(X) = +1) = \mathbb{P}(\bar{h}'(X) = +1)$ and $\lambda(\bar{h}) = \min\{\lambda(h), \lambda(h')\}$. That is, $\bar{h}$ and $\bar{h}'$ are identical to h and h' except that we adjust the bias term of the one with larger minority class probability to reduce its minority class probability to be equal to the other's. If $h \neq \bar{h}$, then most of the probability mass of $\{x : h(x) \neq \bar{h}(x)\}$ is contained in the majority class region of h' (or vice versa if $h' \neq \bar{h}'$), and in fact every point in $\{x : h(x) \neq \bar{h}(x)\}$ is labeled by $\bar{h}$ according to the majority class label (and similarly for h' and $\bar{h}'$). Therefore, we must have $\mathbb{P}(h(X) \neq h'(X)) \geq \mathbb{P}(\bar{h}(X) \neq \bar{h}'(X))$.

We also have that $\mathbb{P}(\bar{h}(X) \neq \bar{h}'(X)) \geq \frac{2\alpha(h,h')}{\pi}\lambda(\bar{h})$. To see this, consider the projection onto the 2-dimensional plane defined by w and w', as in Figure 3.5.2. Because the two decision boundaries must intersect inside the acute angle, the probability mass contained in each of the two wedges (both with $\alpha(h, h')$ angle) making up the projected region of disagreement between $\bar{h}$ and $\bar{h}'$ must be at least an $\alpha(h, h')/\pi$ fraction of the total minority class probability for the respective classifier, implying the union of these two wedges has probability mass at least $\frac{2\alpha(h,h')}{\pi}\lambda(\bar{h})$. Thus, we have $\mathbb{P}(h(X) \neq h'(X)) \geq \max\left\{|\lambda(h) - \lambda(h')|, \frac{2\alpha(h,h')}{\pi}\min\{\lambda(h), \lambda(h')\}\right\}$. In particular,

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

The region of disagreement of this set is at most

$$DIS\left(\left\{h' : \frac{2\alpha(h, h')}{\pi}(\lambda(h) - r) \leq r \wedge |\lambda(h) - \lambda(h')| \leq r\right\}\right)$$

$$\subseteq DIS(\{h' : w' = w \wedge |\lambda(h') - \lambda(h)| \leq r\}) \cup DIS(\{h' : \alpha(h, h') \leq \pi r/\lambda(h) \wedge |\lambda(h) - \lambda(h')| = r\}),$$

71

where this last line follows from the following reasoning. Take y_{maj} to be the majority class of h (arbitrary if $\lambda(h) = 1/2$). For any h' with $|\lambda(h) - \lambda(h')| < r$, the h'' with $\alpha(h, h'') = \alpha(h, h')$ having $\mathbb{P}(h(X) = y_{maj}) - \mathbb{P}(h''(X) = y_{maj}) = r$ disagrees with h on a set of points containing $\{x : h'(x) \neq h(x) = y_{maj}\}$; likewise, the one having $\mathbb{P}(h(X) = y_{maj}) - \mathbb{P}(h''(X) = y_{maj}) = -r$ disagrees with h on a set of points containing $\{x : h'(x) \neq h(x) = -y_{maj}\}$. So any point in disagreement between h and some h' with $|\lambda(h) - \lambda(h')| < r$ and $\alpha(h, h') \leq \pi r/\lambda(h)$ is also disagreed upon by some h'' with $|\lambda(h) - \lambda(h'')| = r$ and $\alpha(h, h'') \leq \pi r/\lambda(h)$.

Some simple trigonometry shows that $DIS(\{h' : \alpha(h, h') \leq \pi r/\lambda(h) \wedge |\lambda(h) - \lambda(h')| = r\})$ is contained in the set of points within distance $sin(\pi r/\lambda(h)) \leq \pi r/\lambda$ of the two hyperplanes representing $h_1(x) = sign(w \cdot x + b_1)$ and $h_2(x) = sign(w \cdot x + b_2)$ defined by the property that $\lambda(h_1) - \lambda(h) = \lambda(h) - \lambda(h_2) = r$, so that the total region of disagreement is contained within

$$\{x : h_1(x) \neq h_2(x)\} \cup \{x : \min\{|w \cdot x + b_1|, |w \cdot x + b_2|\} \leq \pi r/\lambda(h)\}.$$

Clearly, $\mathbb{P}(\{x : h_1(x) \neq h_2(x)\}) = 2r$. Using previous results [Balcan et al., 2006, Hanneke, 2007b], we know that $\mathbb{P}(\{x : \min\{|w \cdot x + b_1|, |w \cdot x + b_2|\} \leq \pi r/\lambda(h)\}) \leq 2\pi\sqrt{n}r/\lambda(h)$ (since the probability mass contained within this distance of a hyperplane is maximized when the hyperplane passes through the origin). Thus, the probability of the entire region of disagreement is at most $(2 + 2\pi\sqrt{n}/\lambda(h))r$, so that (3.1) holds, and therefore the disagreement coefficient is finite. $\square$

3.5.3 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 $(\mathbb{C}, \mathcal{D})$ is learnable at an exponential rate, then for any distribution $\mathcal{D}'$ such that for all measurable $A \subseteq \mathcal{X}$, $\lambda\mathbb{P}_{\mathcal{D}}(A) \leq \mathbb{P}_{\mathcal{D}'}(A) \leq (1/\lambda)\mathbb{P}_{\mathcal{D}}(A)$ for some $\lambda \in (0, 1]$, $(\mathbb{C}, \mathcal{D}')$ is also learnable at an exponential rate. In particular, we can simply use the algorithm
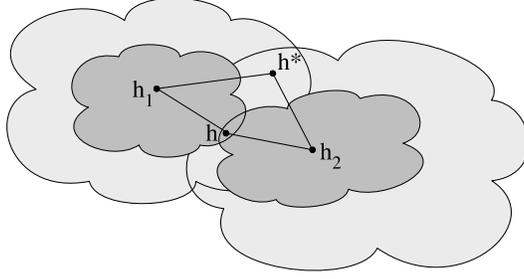
Figure 3.3: Illustration of the proof of Theorem 3.10. The dark gray regions represent $B_{\mathcal{D}_1}(h_1, 2r)$ and $B_{\mathcal{D}_2}(h_2, 2r)$. The function h that gets returned is in the intersection of these. The light gray regions represent $B_{\mathcal{D}_1}(h_1, \epsilon/3)$ and $B_{\mathcal{D}_2}(h_2, \epsilon/3)$. The target function h^* is in the intersection of these. We therefore must have $r \leq \epsilon/3$, and by the triangle inequality $\mathrm{er}(h) \leq \epsilon$.

for $(\mathbb{C}, \mathcal{D})$, filter the examples from $\mathcal{D}'$ so that they appear like examples from $\mathcal{D}$, and then any t large enough to find an $\epsilon\lambda$-good classifier with respect to $\mathcal{D}$ is large enough to find an ϵ-good classifier with respect to $\mathcal{D}'$.

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

Theorem 3.10. *Let $\mathbb{C}$ be an arbitrary hypothesis class. Assume that the pairs $(\mathbb{C}, \mathcal{D}_1)$ and $(\mathbb{C}, \mathcal{D}_2)$ have label complexities $\Lambda_1(\epsilon, \delta, h^*)$ and $\Lambda_2(\epsilon, \delta, h^*)$ respectively, where $\mathcal{D}_1$ and $\mathcal{D}_2$ have density functions $\mathcal{P}r_{\mathcal{D}_1}$ and $\mathcal{P}r_{\mathcal{D}_2}$ respectively. Then for any $\alpha \in [0, 1]$, the pair $(\mathbb{C}, \alpha\mathcal{D}_1 + (1 - \alpha)\mathcal{D}_2)$ has label complexity at most*
$$2 \lceil \max\{\Lambda_1(\epsilon/3, \delta/2, h^*), \Lambda_2(\epsilon/3, \delta/2, h^*)\} \rceil.$$

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, $\mathcal{D}_1$, and $\mathcal{D}_2$, which achieves this label complexity bound.

Suppose algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ achieve the stated label complexities under $\mathcal{D}_1$ and $\mathcal{D}_2$ re-

73

spectively. 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 $\mathcal{D}_1$, and one distributed according to $\mathcal{D}_2$, 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

$$u_i < \frac{\alpha \mathcal{P}r_{\mathcal{D}_1}(x_i)}{\alpha \mathcal{P}r_{\mathcal{D}_1}(x_i) + (1 - \alpha)\mathcal{P}r_{\mathcal{D}_2}(x_i)},$$

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 $\mathcal{D}_1$, so after $t/2$ queries, the current error of h_1 with respect to $\mathcal{D}_1$ is, with probability $1 - \delta/2$, at most $\inf\{\epsilon' : \Lambda_1(\epsilon', \delta/2, h^*) \leq t/2\}$. A similar argument applies to the error of h_2 with respect to $\mathcal{D}_2$.

Finally, let

$$r = \inf\{r : B_{\mathcal{D}_1}(h_1, r) \cap B_{\mathcal{D}_2}(h_2, r) \neq \emptyset\},$$

where

$$B_{\mathcal{D}_i}(h_i, r) = \{h \in \mathbb{C} : \mathcal{P}r_{\mathcal{D}_i}(h(x) \neq h_i(x)) \leq r\}.$$

Define the output of the algorithm to be any $h \in B_{\mathcal{D}_1}(h_1, 2r) \cap B_{\mathcal{D}_2}(h_2, 2r)$. If a total of $t \geq 2\lceil \max\{\Lambda_1(\epsilon/3, \delta/2, h^*), \Lambda_2(\epsilon/3, \delta/2, h^*)\}\rceil$ 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 3.3 for an illustration of these ideas.) □

3.5.4 Lower Bounds

Given the previous discussion, one might suspect that *any* pair $(\mathbb{C}, \mathcal{D})$ is learnable at an exponential rate, under some mild condition such as finite VC dimension. However, we show in the
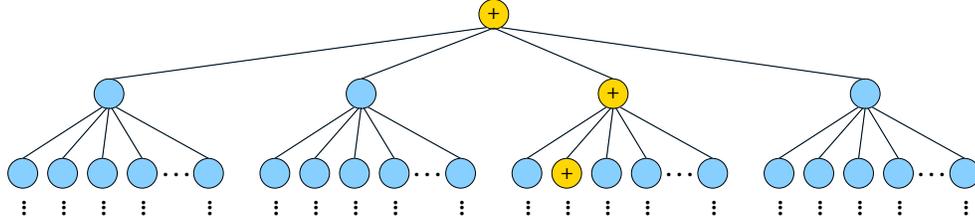
74

Figure 3.4: 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. For any $\phi(\epsilon) = o(1/\epsilon)$, when the number of children and probability mass of each node at each subsequent level are set in a certain way, label complexities of $o(\phi(\epsilon))$ are not achievable for all targets.

following that this is *not* the case, even for some simple geometric concept classes when the distribution is especially nasty.

Theorem 3.11. *For any positive function $\phi(\epsilon) = o(1/\epsilon)$, there exists a pair $(\mathbb{C}, \mathcal{D})$, with the VC dimension of $\mathbb{C}$ equal 1, such that for any achievable label complexity $\Lambda(\epsilon, \delta, h)$ for $(\mathbb{C}, \mathcal{D})$, for any $\delta \in (0, 1/4)$,*

$$\exists h \in \mathbb{C} \text{ s.t. } \Lambda(\epsilon, \delta, h) \neq o(\phi(\epsilon)).$$

In particular, taking $\phi(\epsilon) = 1/\sqrt{\epsilon}$ (for example), this implies that there exists a $(\mathbb{C}, \mathcal{D})$ that is not learnable at an exponential rate (in the sense of Definition 3.3).

Proof. If we can prove this for any such $\phi(\epsilon) \neq O(1)$, then clearly this would imply the result holds for $\phi(\epsilon) = O(1)$ as well, so we will focus on $\phi(\epsilon) \neq O(1)$ case. Let T be a fixed infinite tree in which each node at depth i has c_i children; c_i is defined shortly below. We consider learning the hypothesis class $\mathbb{C}$ where each $h \in \mathbb{C}$ 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 $h \in \mathbb{C}$ there is precisely one node on each level of the tree labeled 1 by h (i.e. one node at each depth). $\mathbb{C}$ 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 3.4.

Now we define $\mathcal{D}$, a "bad" distribution for $\mathbb{C}$. Let $\{\ell_i\}_{i=1}^{\infty}$ be any sequence of positive numbers s.t. $\sum_{i=1}^{\infty} \ell_i = 1$. ℓ_i will bound the total probability of all nodes on level i according to $\mathcal{D}$. Assume all nodes on level i have the same probability according to $\mathcal{D}$, and call this p_i. We define the values of p_i and c_i recursively as follows. For each $i \geq 1$, we define p_i as any positive number s.t. $p_i \lceil \phi(p_i) \rceil \prod_{j=0}^{i-2} c_j \leq \ell_i$ and $\phi(p_i) \geq 4$, and define $c_{i-1} = \lceil \phi(p_i) \rceil$. We are guaranteed that such a value of p_i exists by the assumptions that $\phi(\epsilon) = o(1/\epsilon)$, meaning $\lim_{\epsilon \to 0} \epsilon \phi(\epsilon) = 0$, and that $\phi(\epsilon) \neq O(1)$. Letting $p_0 = 1 - \sum_{i \geq 1} p_i \prod_{j=0}^{i-1} c_j$ completes the definition of $\mathcal{D}$.

With this definition of the parameters above, since $\sum_i p_i \leq 1$, we know that for any $\epsilon_0 > 0$, there exists some $\epsilon < \epsilon_0$ such that for some level j, $p_j = \epsilon$ and thus $c_{j-1} \geq \phi(p_j) = \phi(\epsilon)$. We will use this fact to show that $\propto \phi(\epsilon)$ labels are needed to learn with error less than ϵ for these values of ϵ. To complete the proof, we must prove the existence of a "difficult" target function, customized to challenge the particular learning algorithm being used. To accomplish this, we will use the probabilistic method to prove the existence of a point in each level i such that any target function labeling that point positive would have a label complexity $\geq \phi(p_i)/4$. The difficult target function simply strings these points together.

To begin, we define $x_0 =$ the root node. Then for each $i \geq 1$, recursively define x_i as follows. Suppose, for any h, the set R_h and the classifier $\hat{h}_h$ are, respectively, the random variable representing the set of examples the learning algorithm would request, and the classifier the learning algorithm would output, when h is the target and its label request budget is set to $t = \lfloor \phi(p_i)/2 \rfloor$. For any node x, we will let Children(x) denote the set of children of x, and Subtree(x) denote the set of x along with all descendants of x. Additionally, let h_x denote any classifier in

$\mathbb{C}$ s.t. $h_x(x) = +1$. Now note that

$$\max_{x \in \text{Children}(x_{i-1})} \inf_{h \in \mathbb{C}: h(x)=+1} \mathbb{P}\{\mathbb{P}_{\mathcal{D}}(h(X) \neq \hat{h}_h(X)) > p_i\}$$

$$\geq \frac{1}{c_{i-1}} \sum_{x \in \text{Children}(x_{i-1})} \inf_{h \in \mathbb{C}: h(x)=+1} \mathbb{P}\{\mathbb{P}_{\mathcal{D}}(h(X) \neq \hat{h}_h(X)) > p_i\}$$

$$\geq \frac{1}{c_{i-1}} \sum_{x \in \text{Children}(x_{i-1})} \mathbb{P}\{\forall h \in \mathbb{C} : h(x) = +1, \text{Subtree}(x) \cap R_h = \emptyset \wedge \mathbb{P}_{\mathcal{D}}(h(X) \neq \hat{h}_h(X)) > p_i\}$$

$$= \mathbb{E}\left[\frac{1}{c_{i-1}} \sum_{x \in \text{Children}(x_{i-1}):\text{Subtree}(x) \cap R_{h_x}=\emptyset} \mathbb{I}\left[\forall h \in \mathbb{C} : h(x) = +1, \mathbb{P}_{\mathcal{D}}\left(h(X) \neq \hat{h}_h(X)\right) > p_i\right]\right]$$

$$\geq \mathbb{E}\left[\min_{x' \in \text{Children}(x_{i-1})} \frac{1}{c_{i-1}} \sum_{x \in \text{Children}(x_{i-1}):\text{Subtree}(x) \cap R_{h_x}=\emptyset} \mathbb{I}[x' \neq x]\right]$$

$$\geq \frac{1}{c_{i-1}}(c_{i-1} - t - 1) = \frac{1}{\lfloor \phi(p_i) \rfloor}(\lfloor \phi(p_i) \rfloor - \lfloor \phi(p_i)/2 \rfloor - 1) \geq \frac{1}{\lfloor \phi(p_i) \rfloor}(\lfloor \phi(p_i) \rfloor/2 - 1) \geq 1/4.$$

The expectations above are over the unlabeled examples and any internal random bits used by the algorithm. The above inequalities imply there exists some $x \in \text{Children}(x_{i-1})$ such that every $h \in \mathbb{C}$ that has $h(x) = +1$ has $\Lambda(p_i, \delta, h) \geq \lfloor \phi(p_i)/2 \rfloor \geq \phi(p_i)/4$; we will take x_i to be this value of x. We now simply take the target function h^* to be the classifier that labels x_i positive for all i, and labels every other point negative. By construction, we have $\forall i, \Lambda(p_i, \delta, h^*) \geq \phi(p_i)/4$, and therefore

$$\forall \epsilon_0 > 0, \exists \epsilon < \epsilon_0 : \Lambda(\epsilon, \delta, h^*) \geq \phi(\epsilon)/4,$$

so that $\Lambda(\epsilon, \delta, h^*) \neq o(\phi(\epsilon))$. $\qquad\square$

Note that this implies that the $o(1/\epsilon)$ guarantee of Corollary 3.6 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 $\mathbb{C}_1$ and $\mathbb{C}_2$, such that both $(\mathbb{C}_1, \mathcal{D})$ and $(\mathbb{C}_2, \mathcal{D})$ are learnable at an exponential rate, but $(\mathbb{C}, D)$ is not, where $\mathbb{C} = \{h_1 \cap h_2 : h_1 \in \mathbb{C}_1, h_2 \in \mathbb{C}_2\}$.

3.6 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 chapter. 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 $\mathbb{C}$ in our analysis generates another interpretation of these results. Specifically, Dasgupta [2005] 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 chapter can be interpreted as reflecting a preference partial-ordering of the hypotheses, similar to ideas explored in the passive learning literature [Balcan and Blum, Shawe-Taylor et al., 1998, Vapnik, 1998]. 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 label 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 label 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 label complexity could typically be no better than the label complexity of passive learning; in particular, this observation lead Dasgupta to formulate his splitting index bounds as both target- and distribution-dependent [Dasgupta, 2005]. This fact also applies to bounds on the true label complexity as well. Indeed, the entire distinction between verifiable and true label complexities collapses if we remove the dependence on these unobservable quantities.

One might wonder what the practical implications of the true label complexity of active learning might be since the theoretical improvements we provide are for an unverifiable complexity measure and therefore they do not actually inform the user (or algorithm) of how many labels to allow the algorithm to request. However, there might still be implications for the design of practical algorithms. In some sense, this is the same issue faced in the analysis of universally consistent learning rules in passive learning [Devroye et al., 1996]. There is typically no way to verify how close to the Bayes error rate a classifier is (verifiable complexity is infinite), yet we still want learning rules whose error rates provably converge to the Bayes error in the limit (true complexity is a finite function of epsilon and the distribution of (X, Y)), and we often find such methods quite effective in practice (e.g., k-nearest neighbor methods). So this is one instance where an unverifiable label complexity seems to be a useful guide in algorithm design. In active learning with finite-complexity hypothesis classes we are more fortunate, since the verifiable complexity is finite – and we certainly want algorithms with small verifiable label complexity; however, an analysis of unverifiable complexities still seems relevant, particularly when the verifiable complexity is large. In general, it seems desirable to design algorithms for any given active learning problem that achieve both a verifiable label complexity that is near optimal and a true label complexity that is asymptotically better than passive learning.

Open Questions: There are many interesting open problems within this framework. Perhaps the most interesting of these would be formulating general necessary and sufficient conditions for learnability at an exponential rate, and determining for what types of algorithms Theorem 3.5

can be extended to the agnostic case or to infinite capacity hypothesis classes. We will discuss some progress on this latter problem in the next chapter.

3.7 The Verifiable Label Complexity of the Empty Interval

Let h_- denote the all-negative interval. In this section, we lower bound the verifiable labels complexities achievable for this classifier, with respect to the hypothesis class $\mathbb{C}$ of interval classifiers under a uniform distribution on $[0, 1]$. Specifically, suppose there exists an algorithm A that achieves a verifiable label complexity $\Lambda(\epsilon, \delta, h)$ such that for some $\epsilon \in (0, 1/4)$ and some $\delta \in (0, 1/4)$,

$$\Lambda(\epsilon, \delta, h_-) < \left\lfloor \frac{1}{24\epsilon} \right\rfloor .$$

We prove that this would imply the existence of some interval h' for which the value of $\Lambda(\epsilon, \delta, h')$ is *not valid* under Definition 3.2. We proceed by the probabilistic method.

Consider the subset of intervals

$$H_\epsilon = \left\{ [3i\epsilon, 3(i+1)\epsilon] : i \in \left\{ 0, 1, \ldots, \left\lfloor \frac{1-3\epsilon}{3\epsilon} \right\rfloor \right\} \right\} .$$

Let $s = \lceil \Lambda(\epsilon, \delta, h_-) \rceil$. For any $f \in \mathbb{C}$, let R_f, $\hat{h}_f$, and $\hat{\epsilon}_f$ denote the random variables representing, respectively, the set of examples (x, y) for which $A(s, \delta)$ requests labels (including their $y = f(x)$ labels), the classifier $A(s, \delta)$ outputs, and the confidence bound $A(s, \delta)$ outputs, when f is the target function. Let $\mathbb{I}$ be an indicator function that is 1 if its argument is true and 0 otherwise. Then

$$\max_{f \in H_\epsilon} \mathbb{P}\left(\mathbb{P}_X\left(\hat{h}_f(X) \neq f(X)\right) > \hat{\epsilon}_f\right)$$

$$\geq \frac{1}{|H_\epsilon|} \sum_{f \in H_\epsilon} \mathbb{P}\left(\mathbb{P}_X\left(\hat{h}_f(X) \neq f(X)\right) > \hat{\epsilon}_f\right)$$

$$\geq \frac{1}{|H_\epsilon|} \sum_{f \in H_\epsilon} \mathbb{P}\left((R_f = R_{h_-}) \wedge \left(\mathbb{P}_X\left(\hat{h}_f(X) \neq f(X)\right) > \hat{\epsilon}_f\right)\right)$$

$$= \mathbb{E}\left[\frac{1}{|H_\epsilon|} \sum_{f \in H_\epsilon : R_f = R_{h_-}} \mathbb{I}\left[\mathbb{P}_X\left(\hat{h}_f(X) \neq f(X)\right) > \hat{\epsilon}_f\right]\right]$$

$$\geq \mathbb{E}\left[\frac{1}{|H_\epsilon|} \sum_{f \in H_\epsilon : R_f = R_{h_-}} \mathbb{I}\left[\left(\mathbb{P}_X\left(\hat{h}_f(X) = +1\right) \leq \epsilon\right) \wedge (\hat{\epsilon}_f \leq \epsilon)\right]\right] \tag{3.2}$$

$$= \mathbb{E}\left[\frac{1}{|H_\epsilon|} \sum_{f \in H_\epsilon : R_f = R_{h_-}} \mathbb{I}\left[\left(\mathbb{P}_X\left(\hat{h}_{h_-}(X) \neq h_-(X)\right) \leq \epsilon\right) \wedge (\hat{\epsilon}_{h_-} \leq \epsilon)\right]\right] \tag{3.3}$$

$$\geq \mathbb{E}\left[\left(\frac{|H_\epsilon| - s}{|H_\epsilon|}\right) \mathbb{I}\left[\mathbb{P}_X\left(\hat{h}_{h_-}(X) \neq h_-(X)\right) \leq \hat{\epsilon}_{h_-} \leq \epsilon\right]\right] \tag{3.4}$$

$$= \left(\frac{|H_\epsilon| - s}{|H_\epsilon|}\right) \mathbb{P}\left(\mathbb{P}_X\left(\hat{h}_{h_-}(X) \neq h_-(X)\right) \leq \hat{\epsilon}_{h_-} \leq \epsilon\right)$$

$$\geq \left(\frac{|H_\epsilon| - s}{|H_\epsilon|}\right)(1 - \delta) > \delta.$$

All expectations are over the draw of the unlabeled examples and any additional random bits used by the algorithm. Line 3.2 follows from the fact that all intervals $f \in H_\epsilon$ are of width 3ϵ, so if $\hat{h}_f$ labels less than a fraction ϵ of the points as positive, it must make an error of at least 2ϵ with respect to f, which is more than $\hat{\epsilon}_f$ if $\hat{\epsilon}_f \leq \epsilon$. Note that, for any fixed sequence of unlabeled examples and additional random bits used by the algorithm, the sets R_f are completely determined, and any f and f' for which $R_f = R_{f'}$ must have $\hat{h}_f = \hat{h}_{f'}$ and $\hat{\epsilon}_f = \hat{\epsilon}_{f'}$. In particular, any f for which $R_f = R_{h_-}$ will yield identical outputs from the algorithm, which implies line 3.3. Furthermore, the only classifiers $f \in H_\epsilon$ for which $R_f \neq R_{h_-}$ are those for which some $(x, -1) \in R_{h_-}$ has $f(x) = +1$ (i.e., x is in the f interval). But since there is zero probability that any unlabeled example is in more than one of the intervals in H_ϵ, with probability 1 there are at most s intervals $f \in H_\epsilon$ with $R_f \neq R_{h_-}$, which explains line 3.4.

This proves the existence of some target function $h^* \in \mathbb{C}$ such that $\mathbb{P}(er(h_{s,\delta}) > \hat{\epsilon}_{s,\delta}) > \delta$,

which contradicts the conditions of Definition 3.2.

3.8   Proof of Theorem 3.7

First note that the total number of label requests used by the aggregation procedure in Algorithm 4 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 the true target h^* is a member of $\mathbb{C}_i$. We must show that for any input t such that

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

the aggregation procedure outputs a hypothesis $\hat{h}_t$ such that $er(\hat{h}_t) \leq \epsilon$ 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 $t/(4i^2) \geq \lceil \Lambda_i(\epsilon/2, \delta/2, h^*) \rceil$, 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 $j^* = \operatorname{argmin}_j er(h_j)$. Since $er(h_{j*}) \leq er(h_\ell)$ for any ℓ, we would expect h_{j*} to make no more errors that h_ℓ on points where the two functions disagree. It then follows from Hoeffding's inequality, with probability at least $1 - \delta/4$, for all ℓ,

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

and thus

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

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

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

the probability that h_ℓ mislabels a point x given that $h_\ell(x) \neq h_{j^*}(x)$ is less than $2/3$, 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 $1 - \delta$,

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

3.9   Proof of Theorem 3.8

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

For each i, let

$$\mathbb{C}_i = \{h \in \mathbb{C} : \gamma_h \leq i, k_h \leq i\} \,.$$

Define an algorithm A_i that achieves the required polylog verifiable label complexity on $(\mathbb{C}_i, \mathcal{D})$ as follows. First, run the algorithm A to obtain a function h_A. Then, output the classifier in $\mathbb{C}_i$ that is *closest to* h_A, i.e., the classifier that minimizes the probability of disagreement with 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.

It can be guaranteed that with probability at least $1 - \delta$, the function output by A_i has error no more than $\hat{\epsilon}_t = (2/\delta)\exp\{-(t/i)^{1/i}\}$, which is no more than ϵ, implying that the expression above is a *verifiable* label complexity.

Combining this with Theorem 3.7 yields the desired result.

3.10 Heuristic Approaches to Decomposition

As mentioned, decomposing purely based on verifiable complexity with respect to $(\mathbb{C}, \mathcal{D})$ 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 label complexity may itself have high verifiable complexity.

Although we have not yet found a general method that can provably always find a good decomposition when one exists (other than the trivial method in the proof of Theorem 3.8), we find that a heuristic recursive technique is frequently effective. To begin, define $\mathbb{C}_1 = \mathbb{C}$. Then for $i > 1$, recursively define $\mathbb{C}_i$ as the set of all $h \in \mathbb{C}_{i-1}$ such that $\theta_h = \infty$ with respect to $(\mathbb{C}_{i-1}, \mathcal{D})$. (Here θ_h is the disagreement coefficient of h.) Suppose that for some N, $\mathbb{C}_{N+1} = \emptyset$. Then for the decomposition $\mathbb{C}_1, \mathbb{C}_2, \dots, \mathbb{C}_N$, every $h \in \mathbb{C}$ has $\theta_h < \infty$ with respect to at least one of the sets in which it is contained, which implies that the verifiable label complexity of h with respect to that set is $O(\text{polylog}(1/\epsilon\delta))$, and the aggregation algorithm can be used to achieve polylog label complexity.

We could alternatively perform a similar decomposition using a suitable definition of splitting index [Dasgupta, 2005], or more generally using

$$\limsup_{\epsilon \to 0} \frac{\Lambda_{\mathbb{C}_{i-1}}(\epsilon, \delta, h)}{\left(\log\left(\frac{1}{\epsilon\delta}\right)\right)^k}$$

for some fixed constant $k > 0$.

This procedure does not always generate a good decomposition. However, if $N < \infty$ exists, then it creates a decomposition for which the aggregation algorithm, combined with an appropriate sequence of algorithms $\{A_i\}$, could achieve exponential rates. In particular, this is the case for all of the $(\mathbb{C}, \mathcal{D})$ described in Section 3.5. In fact, even if $N = \infty$, as long as every $h \in \mathbb{C}$ does end up in *some* set $\mathbb{C}_i$ for finite i, this decomposition would still provide exponential rates.

3.11 Proof of Theorem 3.5

We now finally prove Theorem 3.5. This section is mostly self-contained, though we do make use of Theorem 3.7 from Section 3.4 in the final step of the proof.

The proof proceeds according to the following outline. We begin in Lemma 3.12 by describing special conditions under which a CAL-like algorithm has the property that the more unlabeled examples it considers, the smaller the fraction of them it asks to be labeled. Since CAL is able to identify the target's true label on any example it considers (either the label of the example is requested or the example is not in the region of disagreement and therefore the label is already known), we end up with a set of labeled examples growing strictly faster than the number of label requests used to obtain it. This set of labeled examples can be used as a training set in any passive learning algorithm. However, the special conditions under which this happens are rather limiting. In Lemma 3.13, 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, and a simple procedure that boosts the confidence level, extends Lemma 3.12 to the general conditions of Theorem 3.5.

Before jumping into Lemma 3.12, it is useful to define some additional notation. For any $V \subseteq \mathbb{C}$ and $h \in \mathbb{C}$, define the *boundary* of h with respect to $\mathcal{D}$ and V, denoted $\partial_V h$, as

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

Lemma 3.12. *Suppose $(\mathbb{C}, \mathcal{D})$ is such that $\mathbb{C}$ has finite VC dimension d, and $\forall h \in \mathbb{C}, \mathbb{P}(\partial_{\tilde{\mathbb{C}}} h) = 0$. Then for any passive learning label complexity $\Lambda_p(\epsilon, \delta, h)$ for $(\mathbb{C}, \mathcal{D})$ which is nondecreasing as $\epsilon \to 0$, there exists an active learning algorithm achieving a label complexity $\Lambda_a(\epsilon, \delta, h)$ such that, for any $\delta > 0$ and any target function $h^* \in \mathbb{C}$ with $\Lambda_p(\epsilon, \delta, h^*) = \omega(1)$ and $\forall \epsilon > 0, \Lambda_p(\epsilon, \delta, h^*) < \infty$,*

$$\Lambda_a(\epsilon, 2\delta, h^*) = o(\Lambda_p(\epsilon, \delta, h^*)) .$$

Proof. Recall that t is the "budget" of the active learning algorithm, and our goal in this proof is to define an active learning algorithm A_a and a function $\Lambda_a(\epsilon, \delta, h^*)$ such that, if $t \geq \Lambda_a(\epsilon, \delta, h^*)$ and $h^* \in \mathbb{C}$ is the target function, then $A_a(t, \delta)$ will, with probability $1 - \delta$, output an ϵ-good classifier; furthermore, we require that $\Lambda_a(\epsilon, 2\delta, h^*) = o(\Lambda_p(\epsilon, \delta, h^*))$ under the conditions on h^* in the lemma statement.

To construct this algorithm, we perform the learning in two phases. The first is a passive phase, where we focus on reducing a version space, to shrink the region of disagreement; the second is a phase where we construct a labeled training set, which is much larger than the number of label requests used to construct it since all classifiers in the version space agree on many of the examples' labels.

To begin the first phase, we simply request the labels of $x_1, x_2, \ldots, x_{\lfloor t/2 \rfloor}$, and let

$$V = \{h \in \tilde{\mathbb{C}} : \forall i \leq \lfloor t/2 \rfloor, h(x_i) = h^*(x_i)\} .$$

In other words, V is the set of all hypotheses in $\tilde{\mathbb{C}}$ that correctly label the first $\lfloor t/2 \rfloor$ examples. By standard consistency results [Blumer et al., 1989, Devroye et al., 1996, Vapnik, 1982], there is a universal constant $c > 0$ such that, with probability at least $1 - \delta/2$,

$$\sup_{h \in V} er(h) \leq c \left(\frac{d \ln t + \ln \frac{1}{\delta}}{t} \right) .$$

This implies that

$$V \subseteq B_{\tilde{\mathbb{C}}} \left(h^*, c \left(\frac{d \ln t + \ln \frac{1}{\delta}}{t} \right) \right) ,$$

and thus $\mathbb{P}(\mathrm{DIS}(V)) \leq \Delta_t$ where

$$\Delta_t = \mathbb{P} \left(\mathrm{DIS} \left(B_{\tilde{\mathbb{C}}} \left(h^*, c \left(\frac{d \ln t + \ln \frac{1}{\delta}}{t} \right) \right) \right) \right) .$$

Clearly, Δ_t goes to 0 as t grows, by the assumption on $\mathbb{P}(\partial_{\tilde{\mathbb{C}}} h^*)$.

Next, in the second phase of the algorithm, we will actively construct a set of labeled examples to use with the passive learning algorithm. 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 $n_t = \lfloor t/(24\mathbb{P}(\mathrm{DIS}(V))\ln(4/\delta)) \rfloor$, and suppose $t \geq 2$. By a Chernoff bound, with probability at least $1 - \delta/2$, in the sequence of examples $x_{\lfloor t/2 \rfloor+1}, x_{\lfloor t/2 \rfloor+2}, \ldots, x_{\lfloor t/2 \rfloor+n_t}$, at most $t/2$ 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 $x_i' = x_{\lfloor t/2 \rfloor+i}$, and y_i' is either the label agreed upon by all the elements of V, or it is the $h^*(x_{\lfloor t/2 \rfloor+i})$ 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.

Suppose A is the passive learning algorithm that guarantees $\Lambda_p(\epsilon, \delta, h)$ passive label complexities. Then let h_t be the classifier returned by $A(\mathcal{L}, \delta)$. This is the classifier the active learning algorithm outputs.

Note that if $n_t \geq \Lambda_p(\epsilon, \delta, h^*)$, then with probability at least $1 - \delta$ over the draw of $\mathcal{L}$, $er(h_t) \leq \epsilon$. Define

$$\Lambda_a(\epsilon, 2\delta, h^*) = 1 + \inf \left\{ s : s \geq 144\ln(4/\delta)\Lambda_p(\epsilon, \delta, h^*)\Delta_s \right\}.$$

This is well-defined when $\Lambda_p(\epsilon, \delta, h^*) < \infty$ because Δ_s is nonincreasing in s, so some value of s will satisfy the inequality. Note that if $t \geq \Lambda_a(\epsilon, 2\delta, h^*)$, then (with probability at least $1 - \delta/2$)

$$\Lambda_p(\epsilon, \delta, h^*) \leq \frac{t}{144\ln(4/\delta)\Delta_t} \leq n_t .$$

So, by a union bound over the possible failure events listed above ($\delta/2$ for $\mathbb{P}(\mathrm{DIS}(V)) > \Delta_t$, $\delta/2$ for more than $t/2$ examples of $\mathcal{L}$ in $\mathrm{DIS}(V)$, and δ for $er(h_t) > \epsilon$ when the previous failures do not occur), if $t \geq \Lambda_a(\epsilon, 2\delta, h^*)$, then with probability at least $1 - 2\delta$, $er(h_t) \leq \epsilon$. So $\Lambda_a(\epsilon, \delta, h^*)$ is a valid label complexity function, achieved by the described algorithm. Furthermore,

$$\Lambda_a(\epsilon, 2\delta, h^*) \leq 1 + 144\ln(4/\delta)\Lambda_p(\epsilon, \delta, h^*)\Delta_{\Lambda_a(\epsilon, 2\delta, h^*)-2}.$$

If $\Lambda_a(\epsilon, 2\delta, h^*) = O(1)$, then since $\Lambda_p(\epsilon, \delta, h^*) = \omega(1)$, the result is established. Otherwise, since $\Lambda_a(\epsilon, \delta, h^*)$ is nondecreasing as $\epsilon \to 0$, $\Lambda_a(\epsilon, 2\delta, h^*) = \omega(1)$, so we know that $\Delta_{\Lambda_a(\epsilon, 2\delta, h^*)-2} =$

$o(1)$. Thus, $\Lambda_a(\epsilon, 2\delta, h^*) = o\left(\Lambda_p(\epsilon, \delta, h^*)\right)$. $\qquad\square$

As an interesting aside, it is also true (by essentially the same argument) that under the conditions of Lemma 3.12, the *verifiable* label complexity of active learning is strictly smaller than the *verifiable* label complexity of passive learning in this same sense. In particular, this implies a verifiable label complexity that is $o\left(1/\epsilon\right)$ under these conditions. For instance, with some effort one can show that these conditions are satisfied when the VC dimension of $\mathbb{C}$ is 1, or when the support of $\mathcal{D}$ 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 3.5. Toward this end, we again turn to the idea of a decomposition of $\mathbb{C}$, this time decomposing it into subsets satisfying the condition in Lemma 3.12.

Lemma 3.13. *For any* $(\mathbb{C}, \mathcal{D})$ *where* $\mathbb{C}$ *has finite VC dimension* d, *there exists a countably infinite sequence* $\mathbb{C}_1, \mathbb{C}_2, \ldots$ *such that* $\mathbb{C} = \cup_{i=1}^{\infty} \mathbb{C}_i$ *and* $\forall i, \forall h \in \mathbb{C}_i, \mathbb{P}(\partial_{\tilde{\mathbb{C}}_i} h) = 0$.

Proof. The case of $d = 0$ is clear, so assume $d > 0$. A decomposition procedure is given below. We will show that, if we let $\mathbb{H} = \text{Decompose}(\mathbb{C})$, 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 $\mathcal{H}_{\infty} = \{h \in \mathcal{H} : \mathbb{P}(\partial_{\tilde{\mathcal{H}}} h) = 0\}$

if $\mathcal{H}_{\infty} = \mathcal{H}$ **then**

    Return $\{\mathcal{H}\}$

else

    For $i \in \{1, 2, \ldots\}$, let $\mathcal{H}_i = \left\{h \in \mathcal{H} : \mathbb{P}(\partial_{\tilde{\mathcal{H}}} h) \in ((1 + 2^{-(d+3)})^{-i}; (1 + 2^{-(d+3)})^{1-i}]\right\}$

    Return $\bigcup_{i \in \{1,2,\ldots\}} \text{Decompose}(\mathcal{H}_i) \cup \{\mathcal{H}_{\infty}\}$

end if

For the sake of contradiction, suppose that the maximum recursion depth of Decompose($\mathbb{C}$) 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

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

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

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

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

$$\mathbb{P}(\text{DIS}(B_{\tilde{\mathcal{H}}^{(j-1)}}(h_{d+1}, r))) \in \left((1 + 2^{-(d+3)})^{-i_j}, (1 + 2^{-(d+2)})(1 + 2^{-(d+3)})^{-i_j} \right]. \quad (3.5)$$

In particular, by (3.5), each $h \in B_{\tilde{\mathcal{H}}^{(j)}}(h_{d+1}, r/2)$ has

$$\mathbb{P}(\partial_{\tilde{\mathcal{H}}^{(j-1)}} h) > (1 + 2^{-(d+3)})^{-i_j} \geq (1 + 2^{-(d+2)})^{-1} \mathbb{P}(DIS(B_{\tilde{\mathcal{H}}^{(j-1)}}(h_{d+1}, r))),$$

though by definition of $\partial_{\tilde{\mathcal{H}}^{(j-1)}} h$ and the triangle inequality,

$$\mathbb{P}(\partial_{\tilde{\mathcal{H}}^{(j-1)}} h \setminus DIS(B_{\tilde{\mathcal{H}}^{(j-1)}}(h_{d+1}, r))) = 0.$$

Recall that in general, for sets Q and $R_1, R_2, \ldots, R_k$, if $\mathbb{P}(R_i \setminus Q) = 0$ for all i, then $\mathbb{P}(\bigcap_i R_i) \geq \mathbb{P}(Q) - \sum_{i=1}^{k}(\mathbb{P}(Q) - \mathbb{P}(R_i))$. Thus, for any j, any set of $\leq 2^{d+1}$ classifiers $T \subset B_{\tilde{\mathcal{H}}^{(j)}}(h_{d+1}, r/2)$ must have

$$\mathbb{P}(\cap_{h \in T} \partial_{\tilde{\mathcal{H}}^{(j-1)}} h) \geq (1 - 2^{d+1}(1 - (1 + 2^{-(d+2)})^{-1}))\mathbb{P}(DIS(B_{\tilde{\mathcal{H}}^{(j-1)}}(h_{d+1}, r))) > 0.$$

That is, any set of 2^{d+1} classifiers in $\tilde{\mathcal{H}}^{(j)}$ within distance $r/2$ of h_{d+1} will have boundaries with respect to $\mathcal{H}^{(j-1)}$ which have a nonzero probability overlap. The remainder of the proof will hinge on this fact that these boundaries overlap.

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 $d + 1$). Let $h_d \in$

89

$B_{\tilde{\mathcal{H}}^{(d)}}(h_{d+1}, r/4)$ 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 d). Define $A_d = \{x : h_d(x) \neq h_{d+1}(x)\}$, and let $\Delta_d = 2^{-(d+2)}\mathbb{P}(A_d)$. Now for each $\ell \in \{d-1, d-2, \ldots, 0\}$ in decreasing order, we define the ℓ level of the tree as follows. Let $T_{\ell+1}$ denote the nodes at the $\ell + 1$ level in the tree, and let $A'_\ell = \bigcap_{h \in T_{\ell+1}} \partial_{\tilde{\mathcal{H}}^{(\ell)}} h$. We iterate over the elements of $T_{\ell+1}$ in left-to-right order, and for each one h, we find $h' \in B_{\tilde{\mathcal{H}}^{(\ell)}}(h, \Delta_{\ell+1})$ with

$$\mathbb{P}_{\mathcal{D}}(h(x) \neq h'(x) \wedge x \in A'_\ell) > 0 .$$

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

$$A'_\ell \leftarrow A'_\ell \cap \{x : h(x) \neq h'(x)\} .$$

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

Note that for $\ell \in \{0, 1, 2, \ldots, d\}$, every node in the left subtree of any h at level $\ell + 1$ is strictly within distance $2\Delta_\ell$ of h, and every node in the right subtree of any h at level $\ell + 1$ is strictly within distance $2\Delta_\ell$ of the right child of h. Thus,

$$\mathbb{P}(\exists h' \in T_\ell, h'' \in Subtree(h') : h'(x) \neq h''(x)) < 2^{d+1} 2\Delta_\ell.$$

Since

$$2^{d+1} 2\Delta_\ell = \mathbb{P}(A_\ell) = \mathbb{P}(x \in \bigcap_{h' \in T_{\ell+1}} \partial_{\tilde{\mathcal{H}}^{(\ell)}} h' \text{ and } \forall \text{ siblings } h_1, h_2 \in T_\ell, h_1(x) \neq h_2(x)),$$

there must be some set

$$A^*_\ell = \{x \in \bigcap_{h' \in T_{\ell+1}} \partial_{\tilde{\mathcal{H}}^{(\ell)}} h' \text{ s.t. } \forall \text{siblings } h_1, h_2 \in T_\ell, h_1(x) \neq h_2(x)$$

$$\text{and } \forall h \in T_\ell, h' \in Subtree(h), h(x) = h'(x)\} \subseteq A_\ell$$

with $\mathbb{P}(A_\ell^*) > 0$. That is, for every h at level $\ell + 1$, every node in its left subtree agrees with h on every $x \in A_\ell^*$ and every node in its right subtree disagrees with h on every $x \in A_\ell^*$. Therefore, taking any $\{x_0, x_1, x_2, \ldots, x_d\}$ such that each $x_\ell \in A_\ell^*$ creates a shatterable set (shattered by the set of leaf nodes in the tree). This contradicts VC dimension d, so we must have the desired claim that the maximum recursion depth is at most d. $\square$

Before completing the proof of Theorem 3.5, we have two additional minor concerns to address. The first is that the confidence level in Lemma 3.12 is slightly smaller than needed for the theorem. The second is that Lemma 3.12 only applies when $\Lambda_p(\epsilon, \delta, h^*) < \infty$ for all $\epsilon > 0$. We can address both of these concerns with the following lemma.

Lemma 3.14. *Suppose* $(\mathbb{C}, \mathcal{D})$ *is such that* $\mathbb{C}$ *has finite VC dimension d, and suppose* $\Lambda'_a(\epsilon, \delta, h^*)$ *is a label complexity for* $(\mathbb{C}, \mathcal{D})$. *Then there is a label complexity* $\Lambda_a(\epsilon, \delta, h^*)$ *for* $(\mathbb{C}, \mathcal{D})$ *s.t. for any* $\delta \in (0, 1/4)$ *and* $\epsilon \in (0, 1/2)$,

$$\Lambda_a(\epsilon, \delta, h^*) \leq (k+2) \max \begin{cases} \min \left\{ \Lambda'_a(\epsilon/2, 4\delta, h^*), \frac{16 d \log(26/\epsilon) + 8 \log(4/\delta)}{\epsilon} \right\} \\ (k+1)^2 72 \log(4(k+1)^2/\delta) \end{cases},$$

where $k = \lceil \log(\delta/2)/\log(4\delta) \rceil$.

Proof. Suppose A'_a is the algorithm achieving $\Lambda'_a(\epsilon, \delta, h^*)$. Then we can define a new algorithm A_a as follows. Suppose t is the budget of label requests allowed of A_a and δ is its confidence argument. We partition the indices of the unlabeled sequence into $k + 2$ infinite subsequences. For $i \in \{1, 2, \ldots, k\}$, let $h_i = A'_a(t/(k+2), 4\delta)$, each time running A'_a on a different one of these subsequence, rather than on the full sequence. From one of the remaining two subsequences, we request the labels of the first $t/(k+2)$ unlabeled examples and let h_{k+1} denote any classifier in $\mathbb{C}$ consistent with these labels. From the remaining subsequence, for each $i, j \in \{1, 2, \ldots, k+1\}$ s.t. $\mathbb{P}(h_i(X) \neq h_j(X)) > 0$, we find the first $\lfloor t/((k+2)(k+1)k) \rfloor$ examples x s.t. $h_i(x) \neq h_j(x)$, request their labels and let m_{ij} denote the number of mistakes made by h_i on these labels (if $\mathbb{P}(h_i(X) \neq h_j(X)) = 0$, we let $m_{ij} = 0$). Now take as the return value of A_a the classifier $h_{\hat{i}}$

where $\hat{i} = \arg\min_i \max_j m_{ij}$.

Suppose $t \geq \Lambda_a(\epsilon, \delta, h^*)$. First note that, by a Hoeffding bound argument (similar to the proof of Theorem 3.7), t is large enough to guarantee with probability $\geq 1 - \delta/2$ that $er(h_{\hat{i}}) \leq 2\min_i er(h_i)$. So all that remains is to show that, with probability $\geq 1 - \delta/2$, at least one of these h_i has $er(h_i) \leq \epsilon/2$.

If $\Lambda_a'(\epsilon/2, 4\delta, h^*) > \frac{16d\log(26/\epsilon) + 8\log(4/\delta)}{\epsilon}$, then the classic results for consistent classifiers (e.g., [Blumer et al., 1989, Devroye et al., 1996, Vapnik, 1982]) guarantee that, with probability $\geq 1 - \delta/2$, $er(h_{k+1}) \leq \epsilon/2$. Otherwise, we have $t \geq (k+2)\Lambda_a'(\epsilon/2, 4\delta, h^*)$. In this case, each of $h_1, \ldots, h_k$ has an independent $\geq 1 - 4\delta$ probability of having $er(h_i) \leq \epsilon/2$. The probability at least one of them achieves this is therefore at least $1 - (4\delta)^k \geq 1 - \delta/2$. $\qquad \square$

We are now ready to combine these lemmas to prove Theorem 3.5.

Theorem 3.5. Theorem 3.5 now follows by a simple combination of Lemmas 3.12 and 3.13, along with Theorem 3.7 and Lemma 3.14. That is, the passive learning algorithm achieving passive learning label complexity $\Lambda_p(\epsilon, \delta, h)$ on $(\mathbb{C}, \mathcal{D})$ also achieves passive label complexity $\bar{\Lambda}_p(\epsilon, \delta, h) = \min_{\epsilon' \leq \epsilon} \lceil \Lambda_p(\epsilon', \delta, h) \rceil$ on any $(\mathbb{C}_i, \mathcal{D})$, where $\mathbb{C}_1, \mathbb{C}_2, \ldots$ is the decomposition from Lemma 3.13. So Lemma 3.12 guarantees the existence of active learning algorithms $A_1, A_2, \ldots$ such that A_i achieves a label complexity $\Lambda_i(\epsilon, 2\delta, h) = o(\bar{\Lambda}_p(\epsilon, \delta, h))$ on $(\mathbb{C}_i, \mathcal{D})$ for all $\delta > 0$ and $h \in \mathbb{C}_i$ s.t. $\bar{\Lambda}_p(\epsilon, \delta, h)$ is finite and $\omega(1)$. Then Theorem 3.7 tells us that this implies the existence of an active learning algorithm based on these A_i combined with Algorithm 4 , achieving label complexity $\Lambda_a'(\epsilon, 4\delta, h) = o(\bar{\Lambda}_p(\epsilon/2, \delta, h))$ on $(\mathbb{C}, \mathcal{D})$, for any $\delta > 0$ and h s.t. $\bar{\Lambda}_p(\epsilon/2, \delta, h)$ is always finite and is $\omega(1)$. Lemma 3.14 then implies the existence of an algorithm achieving label complexity $\Lambda_a(\epsilon, \delta, h) \in O(\min\{\Lambda_a(\epsilon/2, 4\delta, h), \log(1/\epsilon)/\epsilon\}) \subseteq o(\bar{\Lambda}_p(\epsilon/4, \delta, h)) \subseteq o(\Lambda_p(\epsilon/4, \delta, h))$ for all $\delta \in (0, 1/4)$ and all $h \in \mathbb{C}$. $\qquad \square$

Note there is nothing special about 4 in Theorem 3.5. Using a similar argument, it can be made arbitrarily close to 1.

Chapter 4

Activized Learning: Transforming Passive to Active With Improved Label Complexity

In this chapter, we prove that, in the realizable case, virtually any passive learning algorithm can be transformed into an active learning algorithm with asymptotically strictly superior label complexity, in many cases without significant loss in computational efficiency. We further explore the problem of learning with label noise, and find that even under arbitrary noise distributions, we can still guarantee strict improvements over the known results for passive learning. These are the most general results proven to date regarding the advantages of active learning over passive learning.

4.1 Definitions and Notation

As in previous chapters, all of our asymptotics notation in this chapter will be interpretted as $\epsilon \searrow 0$, when stated for a function of ϵ, the desired excess error, or as $n \to \infty$ when stated for a function of n, the allowed number of label requests. In particular, recall that for two functions ϕ_1 and ϕ_2, we say $\phi_1(\epsilon) = o(\phi_2(\epsilon))$ iff $\lim_{\epsilon \searrow 0} \frac{\phi_1(\epsilon)}{\phi_2(\epsilon)} = 0$. Throughout the chapter, the o notation, as well as "O," "Ω," "ω," "$\ll$," and "$\gg$," where used, should be interpreted purely in terms of the

asymptotic dependence on ϵ or n, with all other quantities held constant, including $\mathcal{D}_{XY}$, δ, and $\mathbb{C}$, where appropriate.

Definition 4.1. *Define the set of functions polynomial in the logarithm of $1/\epsilon$ as follows.*

$$Polylog(1/\epsilon) = \{\phi : [0,1] \to [0,\infty] | \exists k \in [0,\infty) \text{ s.t. } \phi(\epsilon) = O(\log^k(1/\epsilon))\}.$$

Definition 4.2. *We say an active meta-algorithm A_a activizes a passive algorithm A_p for $\mathbb{C}$ under $\mathbb{D}$ if, for any label complexity $\bar{\Lambda}_p$ achieved by A_p, $A_a(A_p, \cdot)$ achieves label complexity $\bar{\Lambda}_a$ such that for all $\mathcal{D} \in \mathbb{D}$,*
$\bar{\Lambda}_p(\epsilon + \nu(\mathbb{C}, \mathcal{D}), \mathcal{D}) \in Polylog(1/\epsilon) \Rightarrow \bar{\Lambda}_a(\epsilon + \nu(\mathbb{C}, \mathcal{D}), \mathcal{D}) \in Polylog(1/\epsilon)$, and if
$\bar{\Lambda}_p(\epsilon + \nu(\mathbb{C}, \mathcal{D}), \mathcal{D}) \ll \infty$ and $\bar{\Lambda}_p(\epsilon + \nu(\mathbb{C}, \mathcal{D}), \mathcal{D}) \notin Polylog(1/\epsilon)$, then there exists a finite constant c such that

$$\bar{\Lambda}_a(c\epsilon + \nu(\mathbb{C}, \mathcal{D}), \mathcal{D}) = o(\bar{\Lambda}_p(\epsilon + \nu(\mathbb{C}, \mathcal{D}), \mathcal{D})).$$

Note that, in keeping with the reductions spirit, we only require the meta-algorithm to successfully improve over the passive algorithm under conditions for which the passive algorithm is itself a reasonable learning algorithm ($\bar{\Lambda}_p \ll \infty$). Given a meta-algorithm satisfying this condition, it is a trivial matter to strengthen it to successfully improve over the passive algorithm even when the passive algorithm is not itself a reasonable method, simply by replacing the passive algorithm with an aggregate of the passive algorithm and some reasonable general-purpose method, such as empiricial error minimization. For simplicity, we do not discuss this matter further.

We will generally refer to any meta-algorithm A_a that activizes *every* passive algorithm A_p for $\mathbb{C}$ under $\mathbb{D}$ as a *general activizer* for $\mathbb{C}$ under $\mathbb{D}$. As we will see, such general activizers do exist under $\mathcal{Realizable}(\mathbb{C})$, under mild conditions on $\mathbb{C}$. However, we will also see that this is typically *not* true for the noisy settings.

4.2 A Basic Activizer

In the following, we adopt the convention that any set of classifiers V *shatters* $\{\}$ iff $V \neq \{\}$ (and otherwise, shattering is defined as in [Vapnik, 1998], as usual). Furthermore, for convenience, we will define $\mathcal{X}^0 = \{\{\}\}$.

Let us begin by motivating the approach we will take below. Similarly to Chapter 3, define the *boundary* as $\partial_{\mathbb{C}} \mathcal{D}_{XY} = \lim_{r \searrow 0} DIS(\mathbb{C}(r))$. If $\mathbb{P}(\partial_{\mathbb{C}} \mathcal{D}_{XY}) = 0$, then methods based on sampling in the region of disagreement and inferring the labels of examples not in the region of disagreement should be effective for activizing (in the realizable case). On the other hand, if $\mathbb{P}(\partial_{\mathbb{C}} \mathcal{D}_{XY}) > 0$, then such methods will fail to focus the sampling region beyond a constant fraction of $\mathcal{X}$, so alternative methods are needed. To cope with such situations, we might exploit the fact that the region of disagreement of the set of classifiers with relatively small empirical error rates on a labeled sample (call this set $\hat{\mathbb{C}}(\tau)$) converges to $\partial_{\mathbb{C}} \mathcal{D}_{XY}$ (up to measure-zero differences). So, for a large enough labeled sample, a random point $x \in DIS(\hat{\mathbb{C}}(\tau))$ will probably be in the boundary region. We can exploit this fact by using x to split $\hat{\mathbb{C}}(\tau)$ into two subsets: $V_+ = \{h \in \hat{\mathbb{C}}(\tau) : h(x) = +1\}$ and $V_- = \{h \in \hat{\mathbb{C}}(\tau) : h(x) = -1\}$. Now, if $x \in \partial_{\mathbb{C}} \mathcal{D}_{XY}$, then $\inf_{h \in V_+} er(h) = \inf_{h \in V_-} er(h) = \nu(\mathbb{C}, \mathcal{D}_{XY})$. So, for almost every point $x' \in \mathcal{X} \setminus DIS(V_+)$, we can infer a label for this point, which will agree with some classifier whose error rate is arbitrarily close to $\nu(\mathbb{C}, \mathcal{D}_{XY})$, and similarly for V_-. In particular, in the realizable case, this inferred label is the target function's label, and in the benign noise case, it is the Bayes optimal classifier's label (when $\eta(x') \neq 1/2$). We can therefore infer the label of points not in the region $DIS(V_+) \cap DIS(V_-)$, thus effectively reducing the region we must request labels in. Similarly, this region converges to a region $\partial_{V_+} \mathcal{D}_{XY} \cap \partial_{V_-} \mathcal{D}_{XY}$. If this region has zero probability, then sampling from $DIS(V_+) \cap DIS(V_-)$ effectively focuses the sampling distribution, as needed. Otherwise, we can repeat this argument; for large enough sample sizes, a random point from $DIS(V_+) \cap DIS(V_-)$ will likely be in $\partial_{V_+} \mathcal{D}_{XY} \cap \partial_{V_-} \mathcal{D}_{XY}$, and therefore splits $\hat{\mathbb{C}}(\tau)$ into four sets with $\nu(\mathbb{C}, \mathcal{D}_{XY})$ optimal error rates, and we can further focus the sampling region in this

way. We can repeat this process as needed until we get a partition of $\hat{\mathbb{C}}(\tau)$ with a shrinking intersection of regions of disagreement. Note that this argument can be written more concisely in terms of shattering. That is, a point in $DIS(\hat{\mathbb{C}}(\tau))$ is simply a point that $\hat{\mathbb{C}}(\tau)$ can shatter. Similarly, a point $x' \in DIS(V_+) \cap DIS(V_-)$ is simply a point s.t. $\hat{\mathbb{C}}(\tau)$ shatters $\{x, x'\}$, etc.

The above simple argument leads to a natural algorithm, which effectively improves label complexity for confidence-bounded error in the realizable case. However, to achieve improvements in the label complexity for expected error, it is not sufficient to merely have the probability of a random point in $DIS(\hat{\mathbb{C}}(\tau))$ being in the boundary converging to 1, as this could happen at a slow rate. To resolve this, we can replace the single sample x with multiple samples, and then take a majority vote over whether to infer the label, and which label to infer if we do.

The following meta-algorithm, based on these observations, is central to the results of this chapter. It depends on several parameters, and two types of estimators: $\hat{\Delta}^{(k)}(\cdot, \cdot)$ and $\hat{\Gamma}^{(k)}(\cdot, \cdot, \cdot)$; one possible definition for these is given immediately after the meta-algorithm, along with a discussion of the roles of these various parameters and estimators.

Meta-Algorithm 5 : $Activizer(\mathcal{A}_p, n)$
Input: passive algorithm $\mathcal{A}_p$, label budget n
Output: classifier $\hat{h}$

---

0. Request the first $\lfloor n/3 \rfloor$ labels and let Q denote these $\lfloor n/3 \rfloor$ labeled examples
1. Let $V = \{h \in \mathbb{C} : er_Q(h) - \min_{h' \in \mathbb{C}} er_Q(h') \leq \tau\}$
2. Let $\mathcal{U}_1$ be the next m_n unlabeled examples, and $\mathcal{U}_2$ the next m_n examples after that
3. For $k = 1, 2, \ldots, d+1$
4.   Let $\mathcal{L}_k$ denote the next $\lfloor n/(6 \cdot 2^k \hat{\Delta}^{(k)}(\mathcal{U}_1, \mathcal{U}_2)) \rfloor$ unlabeled examples,
5.   For each $x \in \mathcal{L}_k$,
6.   If $\hat{\Delta}^{(k)}(x, \mathcal{U}_2) \geq 1 - \gamma$, and we've requested $< \lfloor n/(3 \cdot 2^k) \rfloor$ labels in $\mathcal{L}_k$ so far,
7.   Request the label of x and replace it in $\mathcal{L}_k$ by the labeled one
8.   Else, label x with $\arg\max_{y \in \{-1,+1\}} \hat{\Gamma}^{(k)}(x, y, \mathcal{U}_2)$ and replace it in $\mathcal{L}_k$ by the labeled one
9. Return $ActiveSelect(\{\mathcal{A}_p(\mathcal{L}_1), \mathcal{A}_p(\mathcal{L}_2), \ldots, \mathcal{A}_p(\mathcal{L}_{d+1})\}, \lfloor n/3 \rfloor)$

Subroutine: $ActiveSelect(\{h_1, h_2, \ldots, h_N\}, m)$
0. For each $j, k \in \{1, 2, \ldots, N\} : j < k$,
1.   Take the next $\lfloor m/\binom{N}{2} \rfloor$ examples x s.t. $h_j(x) \neq h_k(x)$ (if such examples exist)
2.   Let m_{jk} and m_{kj} respectively denote the number of mistakes h_j and h_k make on these
3. Return $h_{\hat{k}}$, where $\hat{k} = \arg\min_k \max_j m_{kj}$

The meta-algorithm has several parameters to be specified below.

As with Algorithm 0 and the agnostic generalizations thereof, the set V can be represented implicitly by simply performing each step on the full space $\mathbb{C}$, subject to the constraint given in the definition of V, so that we can more easily adapt algorithms that are designed to manipulate $\mathbb{C}$. Note that, since this is the realizable case, the choice of $\tau = 0$ is sufficient, and furthermore enables the possibility of an efficient reduction to the passive algorithm for many interesting concept spaces. The choice of γ is fairly arbitrary; generally, the proof requires only that $\gamma \in (0, 1)$.

The design of the estimators $\hat{\Delta}^{(k)}(\mathcal{U}_1, \mathcal{U}_2)$, $\hat{\Delta}^{(k)}(x, \mathcal{U}_2)$, and $\hat{\Gamma}^{(k)}(x, y, \mathcal{U}_2)$ can be done in a variety of ways. Generally, the only important feature seems to be that they be converging estimators of an appropriate limiting values. For our purposes, given any $m \in \mathbb{N}$ and sequences $\mathcal{U}_1 = \{z_1, \ldots, z_m\} \in \mathcal{X}^m$ and $\mathcal{U}_2 = \{z_{m+1}, z_{m+2}, \ldots, z_{2m}\} \in \mathcal{X}^m$, the following definitions for $\hat{\Delta}^{(k)}(\mathcal{U}_1, \mathcal{U}_2)$, $\hat{\Delta}^{(k)}(z, \mathcal{U}_2)$, and $\hat{\Gamma}^{(k)}(x, y, \mathcal{U}_2)$ will suffice. Generally, we define

$$\hat{\Delta}^{(k)}(\mathcal{U}_1, \mathcal{U}_2) = \frac{1}{m^{1/3}} + \frac{1}{m} \sum_{z \in \mathcal{U}_1} \mathbb{1}[\hat{\Delta}^{(k)}(z, \mathcal{U}_2) \geq 1 - \gamma]. \tag{4.1}$$

For the others, there are two cases to consider. If $k = 1$, the definitions are quite simple:

$$\hat{\Gamma}^{(1)}(x, y, \mathcal{U}_2) = \mathbb{1}[\forall h \in V, h(x) = y],$$

$$\hat{\Delta}^{(1)}(z, \mathcal{U}_2) = \mathbb{1}[z \in DIS(V)].$$

For the other case, namely $k \geq 2$, we first partition $\mathcal{U}_2$ into subsets of size $k - 1$, and record how many of those subsets are shattered by V: for $i \in \{1, 2, \ldots, \lfloor m/(k-1) \rfloor\}$, define $S_i^{(k)} = \{z_{m+1+(i-1)(k-1)}, \ldots, z_{m+i(k-1)}\}$, and let $M_k = \max\left\{1, \sum_{i=1}^{\lfloor m/(k-1) \rfloor} \mathbb{1}\left[V \text{ shatters } S_i^{(k)}\right]\right\}$. Then define $V_{(x,y)} = \{h \in V : h(x) = y\}$, and

$$\hat{\Gamma}^{(k)}(x, y, \mathcal{U}_2) = \sum_{i=1}^{\lfloor m/(k-1) \rfloor} \mathbb{1}\left[V \text{ shatters } S_i^{(k)} \text{ and } V_{(x,-y)} \text{ does not shatter } S_i^{(k)}\right]. \tag{4.2}$$

$\hat{\Delta}^{(k)}(z, \mathcal{U}_2)$ simply estimates the probability that $S \cup \{z\}$ is shatterable by V given S shatterable

by V, as follows.

$$\hat{\Delta}^{(k)}(z, \mathcal{U}_2) = \frac{1}{M_k^{1/3}} + \frac{1}{M_k} \sum_{i=1}^{\lfloor m/(k-1) \rfloor} \mathbb{1}[V \text{ shatters } S_i^{(k)} \cup \{z\}]. \tag{4.3}$$

The following theorem is the main result on activized learning in the realizable case for this chapter.

Theorem 4.3. *Suppose $\mathbb{C}$ is a VC class, $0 \leq \tau = o(1)$, $m_n \geq n$, and $\gamma \in (0, 1)$ is constant. Let $\hat{\Delta}^{(k)}$ and $\hat{\Gamma}^{(k)}$ be defined as in (4.1), (4.3), and (4.2).*

For any passive algorithm $\mathcal{A}_p$, Meta-Algorithm 5 activizes $\mathcal{A}_p$ for $\mathbb{C}$ under $\mathcal{R}ealizable(\mathbb{C})$.

More concisely, Theorem 4.3 states that Meta-Algorithm 5 is a *general activizer* for $\mathbb{C}$. We can also prove the following result on the fixed-confidence version of label complexity.[1]

Theorem 4.4. *Suppose the conditions of Theorem 4.3 hold, and that $\mathcal{A}_p$ achieves a label complexity Λ_p. Then $Activizer(\mathcal{A}_p, \cdot)$ achieves a label complexity Λ_a such that, for any $\delta \in (0, 1)$ and $\mathcal{D} \in \mathcal{R}ealizable(\mathbb{C})$, there is a finite constant c such that $\Lambda_p(\epsilon, c\delta, \mathcal{D}) = O(1) \Rightarrow \Lambda_a(c\epsilon, c\delta, \mathcal{D}) = O(1)$ and*

$$\Lambda_p(\epsilon, \delta, \mathcal{D}) = \omega(1) \Rightarrow \Lambda_a(c\epsilon, c\delta, \mathcal{D}) = o(\Lambda_p(\epsilon, \delta, \mathcal{D})).$$

The proof of Theorems 4.3 and 4.4 are deferred to Section 4.4.

For a more concrete implication, we immediately get the following simple corollary.

Corollary 4.5. *For any VC class $\mathbb{C}$, there exist active learning algorithms that achieve label complexities Λ_a and $\bar{\Lambda}_a$, respectively, such that for all $\mathcal{D}_{XY} \in \mathcal{R}ealizable(\mathbb{C})$,*

$$\bar{\Lambda}_a(\epsilon, \mathcal{D}_{XY}) = o(1/\epsilon), \quad and \quad \forall \delta \in (0, 1), \Lambda_a(\epsilon, \delta, \mathcal{D}_{XY}) = o(1/\epsilon).$$

Proof. For $d = 0$, the result is trivial. For $d \geq 1$, Haussler, Littlestone, and Warmuth [1994] propose passive learning algorithms achieving respective label complexities $\bar{\Lambda}_p(\epsilon, \mathcal{D}_{XY}) = \frac{d}{\epsilon}$ and $\Lambda_p(\epsilon, \delta, \mathcal{D}_{XY}) \leq \frac{70d}{\epsilon} \ln \frac{8}{\delta}$. Plugging this into Theorems 4.3 and 4.4 implies that applying Meta-Algorithm 5 to these passive algorithms yield combined active learning algorithms with the stated behaviors for $\bar{\Lambda}_a$ and Λ_a. $\qquad\square$

---

[1]In fact, this result even holds for a much simpler variant of the algorithm, where $\hat{\Gamma}^{(k)}$ and $\hat{\Delta}^{(k)}$ can be replaced by an estimator that uses a single random $S \in \mathcal{X}^{k-1}$ shattered by V, rather than repeated samples.

For practical reasons, it is interesting to note that all of the label requests in Meta-Algorithm 5 can be performed in three batches: the initial $n/3$, the requests during the $d+1$ iterations (which can all be requested in a single batch), and the requests for the $ActiveSelect$ procedure. However, because of this, we should not expect Meta-Algorithm 5 to have optimal label complexities. In particular, to get exponential rates, we should expect to need $\Theta(n)$ batches. That said, it should be possible to construct the sets $\mathcal{L}_k$ sequentially, updating V after each example added to $\mathcal{L}_k$, and requesting labels as needed while constructing the set, analogous to Algorithm 0. Some care in the choice of stopping criterion on each round is needed to make sure the set $\mathcal{L}_k$ still represents an i.i.d. sample. Such a modification should significantly improve the label complexities compared to Meta-Algorithm 5, while still maintaining the validity of the results proven here.

Note: The restriction to VC classes is not necessary for positive results in activized learning. For instance, even if the concept space $\mathbb{C}$ has infinite VC dimension, but can be decomposed into a countable sequence of VC class subsets, we can still construct an activizer for $\mathbb{C}$ using an aggregation technique similar to that introduced in Chapter 3.

4.3   Toward Agnostic Activized Learning

We might wonder whether it is possible to state a result as general as Theorem 4.3, even for the most general setting $\mathcal{A}gnostic$. However, one can construct VC classes $\mathbb{C}$, and passive algorithms $\mathcal{A}_p$ that cannot be activized for $\mathbb{C}$, even under bounded noise distributions ($\mathcal{T}sybakov(\mathbb{C}, 1, \mu)$), let alone $\mathcal{A}gnostic$. These algorithms tend to have a peculiar dependence on the noise distribution, so that if the noise distribution and h^* align in just the right way, the algorithm becomes very good, and is otherwise not very good; the effect is that we cannot lose much information about the noise distribution if we hope to get these extremely fast rates for these particular distributions, so that the problem becomes more like regression than classification. However, as mentioned, these passive algorithms are not very interesting for most distributions, which leads to an informal conjecture that any *reasonable* passive algorithm can be activized for $\mathbb{C}$ under

$\mathcal{A}gnostic$. More formally, I have the following specific conjecture.

Recall that we say h is a minimizer of the empirical error rate for a labeled sample $\mathcal{L}$ iff $h \in \arg\min_{h' \in \mathbb{C}} er_{\mathcal{L}}(h')$.

> **Conjecture 4.6.** *For any VC class $\mathbb{C}$, there exists a passive algorithm $\mathcal{A}_p$ that outputs a minimizer of the empirical error rate on its training sample such that some active meta-algorithm $\mathcal{A}_a$ activizes $\mathcal{A}_p$ for $\mathbb{C}$ under $\mathcal{A}gnostic$.*

Although, at this writing, this conjecture remains open, the rest of this section may serve as evidence in its favor.

4.3.1   Positive Results

First, we have the following simple lemma, which allows us to restrict the discussion to the $\mathcal{B}enignNoise(\mathbb{C})$ case.

> **Lemma 4.7.** *For any $\mathbb{C}$, if there exists an active algorithm $\mathcal{A}_a$ achieving label complexities $\bar{\Lambda}_a$ and Λ_a, then there exists an active algorithm $\mathcal{A}'_a$ achieving label complexities $\bar{\Lambda}'_a$ and Λ'_a such that, $\forall \mathcal{D} \in \mathcal{A}gnostic$ and $\delta \in (0,1)$, for some functions $\bar{\lambda}(\epsilon, \mathcal{D}), \lambda(\epsilon, \delta, \mathcal{D}) \in Polylog(1/\epsilon)$, If $\mathcal{D} \in \mathcal{B}enignNoise(\mathbb{C})$, then*
>
> $$\bar{\Lambda}'_a(\epsilon + \nu(\mathbb{C}, \mathcal{D}), \mathcal{D}) \leq \max\{2\lceil\bar{\Lambda}_a(\epsilon/2 + \nu(\mathbb{C}, \mathcal{D}), \mathcal{D})\rceil, \bar{\lambda}(\epsilon, \mathcal{D})\},$$
>
> $$\Lambda'_a(\epsilon + \nu(\mathbb{C}, \mathcal{D}), \delta, \mathcal{D}) \leq \max\{2\lceil\Lambda_a(\epsilon + \nu(\mathbb{C}, \mathcal{D}), \delta/2, \mathcal{D})\rceil, \lambda(\epsilon, \delta, \mathcal{D})\},$$
>
> *and if $\mathcal{D} \notin \mathcal{B}enignNoise(\mathbb{C})$, then*
>
> $$\bar{\Lambda}'_a(\epsilon + \nu(\mathbb{C}, \mathcal{D}), \mathcal{D}) \leq \bar{\lambda}(\epsilon, \mathcal{D}),$$
>
> $$\Lambda'_a(\epsilon + \nu(\mathbb{C}, \mathcal{D}), \delta, \mathcal{D}) \leq \lambda(\epsilon, \delta, \mathcal{D}).$$

Proof. Consider a universally consistent passive learning algorithm $\mathcal{A}_u$. Then $\mathcal{A}_u$ achieves label complexities Λ_u and $\bar{\Lambda}_u$ such that for any distribution $\mathcal{D}$ on $\mathcal{X} \times \{-1, +1\}$, $\forall \epsilon, \delta \in (0,1)$, $\bar{\Lambda}_u(\epsilon/2 + \beta(\mathcal{D}), \mathcal{D})$ and $\Lambda_u(\epsilon/2 + \beta(\mathcal{D}), \delta/2, \mathcal{D})$ are both finite. In particular, if $\beta(\mathcal{D}) < \nu(\mathbb{C}, \mathcal{D})$,

then $\bar{\Lambda}_u(\epsilon/2 + \nu(\mathbb{C}, \mathcal{D}), \mathcal{D}) = O(1)$ and $\Lambda_u(\epsilon/2 + \nu(\mathbb{C}, \mathcal{D}), \delta/2, \mathcal{D}) = O(1)$.

Now we simply run $\mathcal{A}_a(\lfloor n/2 \rfloor)$, to get a classifier h_a, and run $\mathcal{A}_u(\mathcal{Z}_{\lfloor n/3 \rfloor})$ (after requesting those first $\lfloor n/3 \rfloor$ labels), to get a classifier h_u. Take the next $n - \lfloor n/2 \rfloor - \lfloor n/3 \rfloor$ unlabeled examples and request their labels; call this set $\mathcal{L}$. If $er_{\mathcal{L}}(h_a) - er_{\mathcal{L}}(h_u) > n^{-1/3}$, return $\hat{h} = h_u$; otherwise, return $\hat{h} = h_a$. I claim that this method achieves the stated result, for the following reasons.

First, let us examine the final step of this algorithm. By Hoeffding's inequality, the probability that $er(\hat{h}) \neq \min\{er(h_a), er(h_u)\}$ is at most $2exp\{-n^{1/3}/24\}$.

Consider the case where $\mathcal{D} \in \mathcal{B}enign\mathcal{N}oise(\mathbb{C})$. For any $n \geq 2\lceil \bar{\Lambda}_a(\epsilon/2 + \nu(\mathbb{C}, \mathcal{D}), \mathcal{D}) \rceil$, $\mathbb{E}[er(h_a)] \leq \nu(\mathbb{C}, \mathcal{D}) + \epsilon/2$, so $\mathbb{E}[er(\hat{h})] \leq \nu(\mathbb{C}, \mathcal{D}) + \epsilon/2 + 2exp\{-n^{1/3}/24\}$, which is at most $\nu(\mathbb{C}, \mathcal{D}) + \epsilon$ if $n \geq 24^3 \ln^3 \frac{4}{\epsilon}$. Also, for any $n \geq 2\lceil \Lambda_a(\epsilon + \nu(\mathbb{C}, \mathcal{D}), \delta/2, \mathcal{D}) \rceil$, with probability at least $1 - \delta/2$, $er(h_a) \leq \nu(\mathbb{C}, \mathcal{D}) + \epsilon$. If additionally, $n \geq 24^3 \ln^3 \frac{4}{\delta}$, then a union bound implies that with probability $\geq 1 - \delta$, $er(\hat{h}) \leq er(h_a) \leq \nu(\mathbb{C}, \mathcal{D}) + \epsilon$.

On the other hand, if $\mathcal{D} \notin \mathcal{B}enign\mathcal{N}oise(\mathbb{C})$, then for any $n \geq 3\lceil \bar{\Lambda}_u(\nu(\mathbb{C}, \mathcal{D}) + \epsilon/2, \mathcal{D}) \rceil$, $\mathbb{E}[er(\hat{h})] \leq \mathbb{E}[\min\{er(h_a), er(h_u)\}] + 2exp\{-n^{1/3}/24\} \leq \mathbb{E}[er(h_u)] + 2exp\{-n^{1/3}/24\} \leq \nu(\mathbb{C}, \mathcal{D}) + \epsilon/2 + 2exp\{-n^{1/3}/24\}$. Again, this is at most $\nu(\mathbb{C}, \mathcal{D}) + \epsilon$ if $n \geq 24^3 \ln^3 \frac{4}{\epsilon}$. Similarly, for any $n \geq 3\lceil \Lambda_u(\nu(\mathbb{C}, \mathcal{D}) + \epsilon, \delta/2, \mathcal{D}) \rceil = O(1)$, with probability $\geq 1 - \delta/2$, $er(h_u) \leq \nu(\mathbb{C}, \mathcal{D}) + \epsilon$. If additionally, $n \geq 24^3 \ln^3 \frac{4}{\delta}$, then a union bound implies that with probability $\geq 1 - \delta$, $er(\hat{h}) \leq er(h_u) \leq \nu(\mathbb{C}, \mathcal{D}) + \epsilon$.

Thus, we can take $\bar{\lambda}(\epsilon, \mathcal{D}) = \max\{24^3 \ln^3 \frac{4}{\epsilon}, 3\lceil \bar{\Lambda}_u(\nu(\mathbb{C}, \mathcal{D}) + \epsilon/2, \mathcal{D}) \rceil\} \in \mathcal{P}olylog(1/\epsilon)$. and $\lambda(\epsilon, \delta, \mathcal{D}) = \max\{24^3 \ln^3 \frac{4}{\delta}, 3\lceil \Lambda_u(\nu(\mathbb{C}, \mathcal{D}) + \epsilon, \delta/2, \mathcal{D}) \rceil\} \in \mathcal{P}olylog(1/\epsilon)$. $\qquad \square$

Because of Lemma 4.7, it suffices to focus our discussion purely on the $\mathcal{B}enign\mathcal{N}oise(\mathbb{C})$ case, since any label complexity results for $\mathcal{B}enign\mathcal{N}oise(\mathbb{C})$ immediately imply almost equally strong label complexity results for $\mathcal{A}gnostic$, losing only an additive polylogarithmic term. With this in mind, we state the following active learning algorithm, designed for the $\mathcal{B}enign\mathcal{N}oise(\mathbb{C})$ setting.

Meta-Algorithm 6: *BenignActivizer*($\mathcal{A}_p$, n)
Input: passive algorithm $\mathcal{A}_p$, label budget n
Output: classifier $\hat{h}$

---

0. Request the first $\lfloor n/3 \rfloor$ labels and let Q denote these $\lfloor n/3 \rfloor$ labeled examples
1. Let $V = \{h \in \mathbb{C} : er_Q(h) - \min_{h' \in \mathbb{C}} er_Q(h') \leq \tau\}$
2. Let $\mathcal{U}_2$ be the next m_n unlabeled examples
3. For $k = 1, 2, \ldots, d$
4. $Q_k \leftarrow \{\}$
5.   For $t = 1, 2, \ldots, \lfloor 2n/(3 \cdot 2^k) \rfloor$
6.   Let x' be the next unlabeled example for which $\min_{j \leq k} \hat{\Delta}^{(j)}(x, \mathcal{U}_2) \geq 1 - \gamma$
7.   Request the label y' of x' and let $Q_k \leftarrow Q_k \cup \{(x', y')\}$
8.   Construct the classifier $\hat{h}_k$, for $k \in \{1, 2, \ldots, d+1\}$ (see description below)
9. Return $\hat{h}_{\hat{k}}$, for $\hat{k} = \max \left\{ k : \max_{j < k} er_{Q_j}(\hat{h}_k) - er_{Q_j}(\hat{h}_j) \leq T_{kj} \right\}$.

The definition of $\hat{h}_k$ in Step 8 of Meta-Algorithm 6 is as follows.

Let $h_k = \mathcal{A}_p(Q_k)$, $k'(x) = \min\{k' : \hat{\Delta}^{(k')}(x, \mathcal{U}_2) < 1 - \gamma\}$, and

$$
\hat{h}_k(x) = \begin{cases} \arg \max_{y \in \{-1, +1\}} \hat{\Gamma}^{(k'(x))}(x, y, \mathcal{U}_2), & \text{if } k'(x) \leq k \\ h_k(x), & \text{otherwise} \end{cases}.
$$

For the threshold T_{kj} in Step 9 of Meta-Algorithm 6, for our purposes, we can take the following definition.

$$
T_{kj} = 5\sqrt{\frac{2048d \ln(1024d) + \ln(32(d+1)/\delta)}{|Q_k|}}.
$$

It is interesting to note that this algorithm requires only two batches of label requests, which is clearly the minimum number for any algorithm that takes advantage of the sequential aspects of active learning. However, even with this, we have the following general results.

Theorem 4.8. *Let $\tau = \frac{15}{n} + 7\sqrt{\frac{\ln(4n) + d \ln \frac{2n}{d}}{n}}$, $\delta \in (0, 1)$, and let $\hat{\Delta}^{(k)}$ and $\hat{\Gamma}^{(k)}$ be defined as in (4.1), (4.3), and (4.2). For any VC class $\mathbb{C}$, by applying Meta-Algorithm 6 with $\mathcal{A}_p$ being any algorithm outputting a minimizer of the empirical error rate from $\mathbb{C}$, the combined active algorithm achieves a label complexity Λ_a such that $\forall \mathcal{D} \in \mathcal{BenignNoise}(\mathbb{C})$,*

$$
\Lambda_a(\epsilon + \nu(\mathbb{C}, \mathcal{D}), \delta, \mathcal{D}) = o(1/\epsilon^2).
$$

The proof of Theorem 4.8 is included in Section 4.4.1. Theorem 4.8, combined with Lemma 4.7, immediately implies the following quite general corollary.

> **Corollary 4.9.** *For any VC class $\mathbb{C}$, and $\delta \in (0, 1)$, there exists an active learning algorithm achieving a label complexity Λ_a such that, $\forall \mathcal{D} \in \mathcal{A}gnostic$,*
>
> $$\Lambda_a(\epsilon + \nu(\mathbb{C}, \mathcal{D}), \delta, \mathcal{D}) = o(1/\epsilon^2).$$

Note that this result shows strict improvements over the known worst-case (minimax) label complexities for passive learning.

4.4   Proofs

4.4.1   Proof of Theorems 4.3, 4.4, and 4.8

Throughout this subsection, we will assume $\mathbb{C}$ is a VC class, $0 \le \tau = o(1)$, $m_n \ge n$, $\gamma \in (0, 1)$, and $\hat{\Delta}^{(k)}$ and $\hat{\Gamma}^{(k)}$ are defined as in (4.1), (4.3) and (4.2), as stated in the conditions of the theorems. Furthermore, we will define $V = \{h \in \mathbb{C} : er_{\lfloor n/3 \rfloor}(h) - \min_{h' \in \mathbb{C}} er_{\lfloor n/3 \rfloor}(h') \le \tau\}$, and unless otherwise specified, $\mathcal{D}_{XY} \in \mathcal{A}gnostic$ and we will simply discuss the behavior for this fixed, but arbitrary, distribution.

Also, recall that we are using the convention that $\mathcal{X}^0 = \{\{\}\}$ and we say a set of classifiers V shatters $\{\}$ iff $V \ne \{\}$.

> **Lemma 4.10.** *For any $N \in \mathbb{N}$, and N classifiers $\{h_1, h_2, \ldots, h_N\}$,*
> *$ActiveSelect(\{h_1, h_2, \ldots, h_N\}, m)$ makes at most m label requests, and if $h_{\hat{k}}$ is the classifier*
> *output by $ActiveSelect(\{h_1, h_2, \ldots, h_N\}, m)$, then with probability*
> $\ge 1 - 2(N-1)exp\{-(m/\binom{N}{2}))/72\}$, $er(h_{\hat{k}}) \le 2\min_k er(h_k)$.

Proof. This proof is essentially identical to the proof of Theorem 3.7 from Chapter 3.

First note that the total number of label requests used by $ActiveSelect$ is at most m, since each pair of classifiers uses at most $m/\binom{N}{2}$ requests.

Let $k^{**} = \operatorname{argmin}_k er(h_k)$. Now for any $j \in \{1, 2, \ldots, N\}$ with $\mathbb{P}(h_j(X) \neq h_{k^{**}}(X)) > 0$, the law of large numbers implies that with probability 1 we will find at least $m/\binom{N}{2}$ examples remaining in the sequence for which $h_j(x) \neq h_{k^{**}}(x)$, and furthermore since $er(h_{k^{**}}|\{x : h_j(x) \neq h_{k^{**}}(x)\}) \leq 1/2$, Hoeffding's inequality implies that $\mathbb{P}(m_{k^{**}j} > (7/12)m/\binom{N}{2}) \leq exp\{-(m/\binom{N}{2})/72\}$. A union bound implies

$$\mathbb{P}\left(\max_j m_{k^{**}j} > (7/12)m/\binom{N}{2}\right) \leq (N-1)exp\left\{-\left(m/\binom{N}{2}\right)/72\right\}.$$

Now suppose $k \in \{1, 2, \ldots, N\}$ has $er(h_k) > 2er(h_{k^{**}})$. In particular, this implies $\mathbb{P}(h_k(X) \neq h_{k^{**}}(X)) > 0$ and $er(h_k|\{x : h_{k^{**}}(x) \neq h_k(x)\}) > 2/3$. By Hoeffding's inequality, we have that $\mathbb{P}(m_{kk^{**}} \leq (7/12)m/\binom{N}{2}) \leq exp\{-(m/\binom{N}{2})/72\}$. By a union bound, we have that $\mathbb{P}(\exists k : er(h_k) > 2er(h_{k^{**}}) \text{ and } \max_j m_{kj} \leq (7/12)m/\binom{N}{2}) \leq (N-1)exp\{-(m/\binom{N}{2})/72\}$.

So, by a union bound, with probability $\geq 1 - 2(N-1)exp\{-(m/\binom{N}{2})/72\}$, for the $\hat{k}$ chosen by $ActiveSelect$,

$$\max_j m_{\hat{k}j} \leq \max_j m_{h_{k^{**}}j} \leq (7/12)m/\binom{N}{2} < \min_{k:er(h_k)>2er(h_{k^{**}})} \max_j m_{kj},$$

and thus $er(h_{\hat{k}}) \leq 2er(h_{k^{**}})$ as claimed. $\qquad \square$

Lemma 4.11. *There is an event H_n, holding with probability $\geq 1 - exp\{-\sqrt{n}\}$, such that for some $\mathbb{C}$-dependent function $\phi(n) = o(1)$, $V \subseteq \mathbb{C}(\phi(n); \mathcal{D}_{XY})$.*

Proof. By the uniform convergence bounds proven by Vapnik [1982], for a $\mathbb{C}$-dependent finite constant c, with probability $\geq 1 - exp\{-n^{1/2}\}$, $V \subseteq \mathbb{C}\left(cn^{-1/4} + \tau; \mathcal{D}_{XY}\right)$. Thus, the result holds for $\phi(n) = cn^{-1/4} + \tau = o(1)$. $\qquad \square$

Lemma 4.12. *If $\tau \geq \frac{15}{n} + 7\sqrt{\frac{\ln(4n) + d\ln\frac{2n}{d}}{n}}$, then there is a strictly positive function $\phi'(n) = o(1)$ such that, with probability $\geq 1 - 1/n$, $\mathbb{C}(\phi'(n); \mathcal{D}_{XY}) \subseteq V$.*

Proof. By the uniform convergence bounds proven by Vapnik [1982], with probability $1 - 1/n$, every $h \in \mathbb{C}$ has $|er(h) - er_{\lfloor n/3 \rfloor}(h)| \leq \tau/3$. Therefore, on this event, $V \supseteq \mathbb{C}(\tau/3; \mathcal{D}_{XY})$. Thus, we can let $\phi'(n) = \tau/3$, which satisfies the desired conditions. $\qquad \square$

Lemma 4.13. *For any $n \in \mathbb{N}$, there is an event H'_n for the data sequence $\mathcal{Z}_{\lfloor n/3 \rfloor}$ with*

$$
\mathbb{P}(H'_n) \geq
\begin{cases}
1, & \text{if } \mathcal{D}_{XY} \in \mathcal{R}ealizable(\mathbb{C}) \\[2mm]
1 - 1/n, & \text{if } \mathcal{D}_{XY} \notin \mathcal{R}ealizable(\mathbb{C}) \text{ but } \tau \geq \frac{15}{n} + 7\sqrt{\frac{\ln(4n) + d\ln\frac{2n}{d}}{n}}
\end{cases},
$$

s.t. on H'_n, for any $k \in \{1, 2, \ldots, d+1\}$ with $\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1) > 0$,

$$
\mathbb{P}(S \in \mathcal{X}^{k-1} : V \text{ shatters } S | \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1)
$$

$$
= \mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[V(r) \text{ shatters } S] = 1 | \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1) = 1.
$$

Proof. For the case of $\mathcal{D}_{XY} \notin \mathcal{R}ealizable(\mathbb{C})$ and $\tau \geq \frac{15}{n} + 7\sqrt{\frac{\ln(4n) + d\ln\frac{2n}{d}}{n}}$, the result immediately follows from Lemma 4.12, which implies that on an event of probability $\geq 1 - 1/n$, for any set S, $\mathbb{1}[V \text{ shatters } S] \geq \lim_{r \searrow 0} \mathbb{1}[V(r) \text{ shatters } S] = \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S]$.

Next we examine the case where $\mathcal{D}_{XY} \in \mathcal{R}ealizable(\mathbb{C})$. We will show this is true for any fixed k, and the existence of H'_n then holds by the union bound. Fix any set $S \in \mathcal{X}^{k-1}$ s.t. $\lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1$. Suppose $V(r)$ does not shatter S for some $r > 0$. Then there is an infinite sequence of sets $\{\{h_1^{(i)}, h_2^{(i)}, \ldots, h_{2^{k-1}}^{(i)}\}\}_i$ with $\forall j \leq 2^{k-1}$, $\mathbb{P}(x : h_j^{(i)}(x) \neq h^*(x)) \searrow 0$, such that each $\{h_1^{(i)}, \ldots, h_{2^{k-1}}^{(i)}\} \subseteq \mathbb{C}(r)$ and shatters S. Since $V(r)$ does not shatter S, $1 = \inf_i \mathbb{1}[\exists j : h_j^{(i)} \notin V(r)] = \inf_i \mathbb{1}[\exists j : h_j^{(i)}(\mathcal{Z}_{\lfloor n/3 \rfloor}) \neq h^*(\mathcal{Z}_{\lfloor n/3 \rfloor})]$. But

$$
\mathbb{E}[\inf_i \mathbb{1}[\exists j : h_j^{(i)}(\mathcal{Z}_{\lfloor n/3 \rfloor}) \neq h^*(\mathcal{Z}_{\lfloor n/3 \rfloor})]] \leq \inf_i \mathbb{E}[\mathbb{1}[\exists j : h_j^{(i)}(\mathcal{Z}_{\lfloor n/3 \rfloor}) \neq h^*(\mathcal{Z}_{\lfloor n/3 \rfloor})]]
$$

$$
\leq \lim_{i \to \infty} \sum_{j \leq 2^{k-1}} \lfloor n/3 \rfloor \mathbb{P}(x : h_j^{(i)}(x) \neq h^*(x)) = 0,
$$

where the second inequality follows from the union bound. Therefore, $\forall r > 0$, $\mathbb{P}(\mathcal{Z}_{\lfloor n/3 \rfloor} \in \mathcal{X}^{\lfloor n/3 \rfloor} : V(r) \text{ does not shatter } S) = 0$ by Markov's inequality. Furthermore, since $\mathbb{1}[V(r) \text{ does not shatter } S]$ is monotonic in r, Markov's inequality and the monotone convergence

theorem give us that

$$\mathbb{P}(\mathcal{Z}_{\lfloor n/3 \rfloor} \in \mathcal{X}^{\lfloor n/3 \rfloor} : \lim_{r \searrow 0} \mathbb{1}[V(r) \text{ does not shatter } S] = 1)$$

$$\leq \mathbb{E}[\lim_{r \searrow 0} \mathbb{1}[V(r) \text{ does not shatter } S]] = \lim_{r \searrow 0} \mathbb{P}(\mathcal{Z}_{\lfloor n/3 \rfloor} \in \mathcal{X}^{\lfloor n/3 \rfloor} : V(r) \text{ does not shatter } S) = 0.$$

This implies that

$$\mathbb{P}(\mathcal{Z}_{\lfloor n/3 \rfloor} \in \mathcal{X}^{\lfloor n/3 \rfloor} : \mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[V(r) \text{ shatters } S] = 0 | \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1) > 0)$$

$$= \lim_{\xi \searrow 0} \mathbb{P}(\mathcal{Z}_{\lfloor n/3 \rfloor} \in \mathcal{X}^{\lfloor n/3 \rfloor} : \mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[V(r) \text{ shatters } S] = 0 | \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1) > \xi)$$

$$\leq \lim_{\xi \searrow 0} \mathbb{P}(\mathcal{Z}_{\lfloor n/3 \rfloor} \in \mathcal{X}^{\lfloor n/3 \rfloor} : \mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1 \neq \lim_{r \searrow 0} \mathbb{1}[V(r) \text{ shatters } S]) > \xi)$$

$$\leq \lim_{\xi \searrow 0} \frac{1}{\xi} \mathbb{E}[\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1 \neq \lim_{r \searrow 0} \mathbb{1}[V(r) \text{ shatters } S])] \text{ (by Markov's ineq)}$$

$$= \lim_{\xi \searrow 0} \frac{1}{\xi} \mathbb{E}[\mathbb{1}[\lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } \mathbf{S}] = 1] \mathbb{P}(\mathcal{Z}_{\lfloor n/3 \rfloor} : \lim_{r \searrow 0} \mathbb{1}[V(r) \text{ shatters } \mathbf{S}] = 0)] \text{ (by Fubini's thm)}$$

$$= \lim_{\xi \searrow 0} 0 = 0.$$

$\square$

Lemma 4.14. *Suppose $k \in \mathbb{N}$ satisfies $\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1) > 0$. There is a function $q(n) = o(1)$ such that, for any $n \in \mathbb{N}$, on event $H_n \cap H'_n$ (defined above),*

$$\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 0 | V \text{ shatters } S) \leq q(n).$$

Proof. By Lemmas 4.11 and 4.13, we know that on event $H_n \cap H'_n$,

$$\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 0 | V \text{ shatters } S)$$

$$= \frac{\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 0 \text{ and } V \text{ shatters } S)}{\mathbb{P}(S \in \mathcal{X}^{k-1} : V \text{ shatters } S)}$$

$$\leq \frac{\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 0 \text{ and } V \text{ shatters } S)}{\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1)}$$

$$\leq \frac{\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 0 \text{ and } \mathbb{C}(\phi(n)) \text{ shatters } S)}{\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1)}.$$

106

Define $q(n)$ as this latter quantity. Since

$$\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 0 \text{ and } \mathbb{C}(r') \text{ shatters } S) \text{ is monotonic in } r',$$

$$\lim_{n \to \infty} q(n) = \lim_{r' \searrow 0} \frac{\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 0 \text{ and } \mathbb{C}(r') \text{ shatters } S)}{\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1)}$$

$$= \frac{\mathbb{E}[\mathbb{1}[\lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 0] \lim_{r' \searrow 0} \mathbb{1}[\mathbb{C}(r') \text{ shatters } S]]}{\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1)} = 0,$$

where the second equality holds by the monotone convergence theorem. This proves

$q(n) = o(1)$, as claimed. $\qquad\square$

Lemma 4.15. *Let $k^* \in \mathbb{N}$ be the smallest index k for which*

$$\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1) > 0 \text{ and}$$

$$\mathbb{P}(S \in \mathcal{X}^{k-1} : \mathbb{P}(x : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S \cup \{x\}] = 1) = 0 | \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1) > \gamma.$$

Such a $k^* \le d+1$ exists, and $\forall \zeta \in (0,1), \exists n_\zeta$ s.t. $\forall n > n_\zeta$, if $\mathcal{D}_{XY} \in \mathcal{R}ealizable(\mathbb{C})$ or $\tau \ge \frac{15}{n} + 7\sqrt{\frac{\ln(4n) + d \ln \frac{2n}{d}}{n}}$ and $\mathcal{D}_{XY} \in \mathcal{B}enignNoise(\mathbb{C})$, on event $H_n \cap H'_n$ (defined above), $\forall k \le k^*$,

$$\mathbb{P}(x : \eta(x) \ne 1/2 \text{ and } \mathbb{P}(S \in \mathcal{X}^{k-1} : V_{(x,h^*(x))} \text{ does not shatter } S | V \text{ shatters } S) > \zeta) =$$

$$\mathbb{P}(x : \eta(x) \ne 1/2 \text{ and } \mathbb{P}(S \in \mathcal{X}^{k-1} : V_{(x,h^*(x))} \text{ does not shatter } S | \lim_{r \searrow 0} \mathbb{1}[V(r) \text{ shatters } S] = 1) > \zeta)$$

$$= 0.$$

Proof. First we prove that such a k^* is guaranteed to exist. As mentioned, by convention any set of classifiers shatters $\{\}$, and $\{\} \in \mathcal{X}^0$, so there exist values of k for which $\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1) > 0$. Furthermore, we will see that for any $k \in \{1, \ldots, d+1\}$, if this condition is satisfied for k, but

$$\mathbb{P}(S \in \mathcal{X}^{k-1} : \mathbb{P}(x : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S \cup \{x\}] = 1) = 0 | \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1) \le \gamma,$$

then $\mathbb{P}(S \in \mathcal{X}^k : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1) > 0$. We prove this by contradiction. Suppose the implication is not true for some k. Then

$$0 < 1 - \gamma$$

$$\leq \quad \mathbb{P}(S \in \mathcal{X}^{k-1} : \mathbb{P}(x : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S \cup \{x\}] = 1) > 0 | \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1)$$

$$\leq \quad \lim_{\xi \searrow 0} \frac{\mathbb{P}(S \in \mathcal{X}^{k-1} : \mathbb{P}(x : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S \cup \{x\}] = 1) > \xi)}{\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1)}$$

$$\leq \quad \lim_{\xi \searrow 0} \frac{\mathbb{E}[\mathbb{P}(x : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } \mathbf{S} \cup \{x\}] = 1)]}{\xi \mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1)} \quad \text{(by Markov's inequality)}$$

$$= \quad \lim_{\xi \searrow 0} \frac{\mathbb{P}(S \in \mathcal{X}^{k} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1)}{\xi \mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1)} = \lim_{\xi \searrow 0} 0 = 0.$$

This is a contradiction, so it must be true that the implication holds for all k. This establishes the existence of k^*, since we definitely have

$$\mathbb{P}(S \in \mathcal{X}^d : \lim_{r \searrow 0} \mathbb{P}(x : \mathbb{C}(r) \text{ shatters } S \cup \{x\}) = 0 | \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1) = 1 > \gamma,$$

so that *some* k satisfies both conditions.

Next we prove the second claim. Take $k \leq k^*$. Let n_ζ be s.t. $\sup_{n > n_\zeta} q(n) < \zeta$; it must exist since $q(n) = o(1)$. By Lemma 4.14, for $n > n_\zeta$, on $H_n \cap H'_n$,

$$\mathbb{P}(x : \eta(x) \neq 1/2 \text{ and } \mathbb{P}(S \in \mathcal{X}^{k-1} : V_{(x,h^*(x))} \text{ does not shatter } S | V \text{ shatters } S) > \zeta)$$

$$\leq \quad \mathbb{P}(x : \eta(x) \neq 1/2 \text{ and }$$

$$\mathbb{P}(S \in \mathcal{X}^{k-1} : V_{(x,h^*(x))} \text{ does not shatter } S | \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1) + q(n) > \zeta)$$

$$\leq \quad \frac{1}{\zeta - q(n)} \mathbb{E}[\mathbb{1}[\eta(x) \neq 1/2] \mathbb{P}(S \in \mathcal{X}^{k-1} : V_{(\mathbf{X}, h^*(\mathbf{X}))} \text{ does not shatter } S | \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1)]$$

$$\text{(by Markov's inequality)}$$

$$\leq \quad \frac{\mathbb{E}[\mathbb{1}[\lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } \mathbf{S}] = 1] \mathbb{P}(x : \eta(x) \neq 1/2 \text{ and } V_{(x,h^*(x))} \text{ does not shatter } \mathbf{S})]}{(\zeta - q(n)) \mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1)} \quad \text{(by Fubini's theorem)}$$

$$\leq \quad \frac{\mathbb{E}[\mathbb{1}[\lim_{r \searrow 0} \mathbb{1}[V(r) \text{ shatters } \mathbf{S}] = 1] \mathbb{P}(x : \eta(x) \neq 1/2 \text{ and } V_{(x,h^*(x))} \text{ does not shatter } \mathbf{S})]}{(\zeta - q(n)) \mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1)} \quad \text{(by Lemma 4.13).} \qquad (4.4)$$

For any set $S \in \mathcal{X}^{k-1}$ for which $\lim_{r \searrow 0} \mathbb{1}[V(r) \text{ shatters } S] = 1$, there is an infinite sequence of sets $\{\{h_1^{(i)}, h_2^{(i)}, \ldots, h_{2^{k-1}}^{(i)}\}\}_i$ with $\forall j \leq 2^{k-1}$, $\mathbb{P}(x : \eta(x) \neq 1/2 \text{ and } h_j^{(i)}(x) \neq h^*(x)) \searrow 0$, such that

each $\{h_1^{(i)}, \ldots, h_{2^{k-1}}^{(i)}\} \subseteq V$ and shatters S. If $V_{(x,h^*(x))}$ does not shatter S, then

$$1 = \inf_i \mathbb{1}[\exists j : h_j^{(i)} \notin V_{(x,h^*(x))}] = \inf_i \mathbb{1}[\exists j : h_j^{(i)}(x) \neq h^*(x)].$$

In particular, by Markov's inequality,

$\mathbb{P}(x : \eta(x) \neq 1/2$ and $V_{(x,h^*(x))}$ does not shatter $S)$

$$\leq \qquad \mathbb{P}(x : \eta(x) \neq 1/2 \text{ and } \inf_i \mathbb{1}[\exists j : h_j^{(i)}(x) \neq h^*(x)] = 1)$$

$$\leq \qquad \mathbb{E}[\mathbb{1}[\eta(X) \neq 1/2] \inf_i \mathbb{1}[\exists j : h_j^{(i)}(X) \neq h^*(X)]]$$

$$\leq \qquad \inf_i \mathbb{P}(x : \eta(x) \neq 1/2 \text{ and } \exists j \text{ s.t. } h_j^{(i)}(x) \neq h^*(x))$$

$$\leq \qquad \sum_{j \leq 2^{k-1}} \lim_{i \to \infty} \mathbb{P}(x : \eta(x) \neq 1/2 \text{ and } h_j^{(i)}(x) \neq h^*(x)) = 0.$$

This means (4.4) equals 0. $\qquad\qquad\square$

Lemma 4.16. *Suppose $k \in \{1, 2, \ldots, d+1\}$ satisfies*

$\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1) > 0$ *and*

$\alpha_k = \mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{P}(x : \mathbb{C}(r) \text{ shatters } S \cup \{x\}) = 0 | \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1) > \gamma.$

Then there is a function $\Delta_n^{(k)} = o(1)$ such that, on event $H_n \cap H_n'$ (defined above),

$\mathbb{P}(x : \mathbb{P}(S \in \mathcal{X}^{k-1} : V \text{ shatters } S \cup \{x\} | V \text{ shatters } S) \geq 1 - (\gamma + \alpha_k)/2) \leq \Delta_n^{(k)}.$

Proof. Let

$$\mathcal{A} = \{S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1 \text{ and } \lim_{r \searrow 0} \mathbb{P}(x : \mathbb{C}(r) \text{ shatters } S \cup \{x\}) = 0\}.$$

Then, letting $\phi(n)$ be as in Lemma 4.11, on event $H_n \cap H_n'$,

$\mathbb{P}(x : \mathbb{P}(S \in \mathcal{X}^{k-1} : V \text{ shatters } S \cup \{x\} | V \text{ shatters } S) \geq 1 - (\gamma + \alpha_k)/2)$

$$\leq \mathbb{P}(x : \mathbb{P}(S \in \mathcal{X}^{k-1} : \mathbb{C}(\phi(n)) \text{ shatters } S \cup \{x\} | \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1)$$

$$+ \mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 0 | V \text{ shatters } S) \geq 1 - (\gamma + \alpha_k)/2) \quad (4.5)$$

By Lemma 4.13, we know there is some finite $\tilde{n}_1$ s.t. any $n > \tilde{n}_1$ has (on event $H_n \cap H_n'$)

$$\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 0 | V \text{ shatters } S) \leq (\alpha_k - \gamma)/3.$$

We therefore have that, for $n > \tilde{n}_1$, on event $H_n \cap H_n'$, (4.5) is at most

$$\mathbb{P}(x:\mathbb{P}(S\in\mathcal{X}^{k-1}:\mathbb{C}(\phi(n)) \text{ shatters } S\cup\{x\}|\lim_{r\searrow0}\mathbb{1}[\mathbb{C}(r) \text{ shatters } S]=1)+(\alpha_k-\gamma)/3\geq 1-(\gamma+\alpha_k)/2)$$

$$\leq \mathbb{P}(x:\mathbb{P}(S\in\mathcal{X}^{k-1}:\mathbb{C}(\phi(n)) \text{ shatters } S\cup\{x\}|S\in\mathcal{A})\alpha_k+(1-\alpha_k)+(\alpha_k-\gamma)/3\geq 1-(\gamma+\alpha_k)/2)$$

$$= \mathbb{P}(x : \mathbb{P}(S \in \mathcal{X}^{k-1} : \mathbb{C}(\phi(n)) \text{ shatters } S \cup \{x\}|S \in \mathcal{A}) \geq (\alpha_k - \gamma)/(6\alpha_k))$$

$$\leq \frac{6\alpha_k}{\alpha_k-\gamma}\mathbb{E}[\mathbb{P}(S \in \mathcal{X}^{k-1} : \mathbb{C}(\phi(n)) \text{ shatters } S \cup \{X\}|S \in \mathcal{A})] \text{ (by Markov's inequality)}$$

$$\leq \frac{6\alpha_k}{\alpha_k-\gamma}\mathbb{E}[\mathbb{P}(x : \mathbb{C}(\phi(n)) \text{ shatters } \mathbf{S} \cup \{x\})|\mathbf{S} \in \mathcal{A}] \text{ (by Fubini's theorem).}$$

We will define $\Delta_n^{(k)}$ equal to this last quantity for any $n > \tilde{n}_1$ (we can take $\Delta_n^{(k)} = 1$ for $n \leq \tilde{n}_1$). It remains only to show this quantity is $o(1)$. Since $\frac{6\alpha_k}{\alpha_k-\gamma}\mathbb{E}[\mathbb{P}(x : \mathbb{C}(r) \text{ shatters } \mathbf{S} \cup \{x\})|\mathbf{S} \in \mathcal{A}]$ is monotonic in r,

$$\lim_{n\to\infty} \Delta_n^{(k)} = \lim_{r\searrow0} \frac{6\alpha_k}{\alpha_k - \gamma}\mathbb{E}[\mathbb{P}(x : \mathbb{C}(r) \text{ shatters } \mathbf{S} \cup \{x\})|\mathbf{S} \in \mathcal{A}].$$

Since for any $S \in \mathcal{X}^{k-1}$, $\mathbb{P}(x : \mathbb{C}(r) \text{ shatters } S \cup \{x\})$ is monotonic in r, the monotone convergence theorem implies

$$\lim_{r\searrow0} \frac{6\alpha_k}{\alpha_k - \gamma}\mathbb{E}[\mathbb{P}(x : \mathbb{C}(r) \text{ shatters } S \cup \{x\})|S \in \mathcal{A}]$$

$$= \frac{6\alpha_k}{\alpha_k - \gamma}\mathbb{E}[\lim_{r\searrow0}\mathbb{P}(x : \mathbb{C}(r) \text{ shatters } S \cup \{x\})|S \in \mathcal{A}] = 0.$$

$\square$

Lemma 4.17. $\forall n \in \mathbb{N}$, *there is an event* $\tilde{H}_n \subseteq H_n \cap H'_n$ *on* $\mathcal{Z}$ *that, if*

$\mathcal{D}_{XY} \in \mathcal{B}enignNoise(\mathbb{C})$, *has*

$\mathbb{P}(\tilde{H}_n) \geq 1 - cn^{4/3} \cdot exp\{-c'n^{1/3}\} - \mathbb{1}[\mathcal{D}_{XY} \notin \mathcal{R}ealizable(\mathbb{C})]n^{-1}$, *for* $\mathcal{D}_{XY}$- *and*

$\mathbb{C}$-*dependent constants* $c, c' \in (0, \infty)$, *such that*

$$\forall n \in \mathbb{N}, \text{ on } \tilde{H}_n, |\{x \in \mathcal{L}_{k^*} : \hat{\Delta}^{(k^*)}(x, \mathcal{U}_2) \geq 1 - \gamma\}| \leq \lfloor n/(3 \cdot 2^{k^*}) \rfloor, \tag{4.6}$$

$\exists \breve{\Delta}_n^{(k^*)} = o(1)$ *and* $\tilde{\Delta}_n^{(k^*)} = o(1)$ *s.t.* $\forall n \in \mathbb{N}$, *on* $\tilde{H}_n$,

$$\bar{\Delta}^{(k^*)}(\mathcal{U}_2) \leq \breve{\Delta}_n^{(k^*)} \text{ and } \hat{\Delta}^{(k^*)}(\mathcal{U}_1, \mathcal{U}_2) \leq \tilde{\Delta}_n^{(k^*)}, \tag{4.7}$$

where $\forall k, \bar{\Delta}^{(k)}(\mathcal{U}_2) = \mathbb{P}(x : \hat{\Delta}^{(k)}(x, \mathcal{U}_2) \geq 1 - \gamma)$; *also* $\exists n^* \in \mathbb{N}$ *s.t.* $\forall n > n^*$, *if*

$\mathcal{D}_{XY} \in \mathcal{R}ealizable(\mathbb{C})$, *on* $\tilde{H}_n, \forall x \in \mathcal{L}_{k^*}$,

$$\hat{\Delta}^{(k^*)}(x, \mathcal{U}_2) < 1 - \gamma \Rightarrow \hat{\Gamma}^{(k^*)}(x, -h^*(x), \mathcal{U}_2) < \hat{\Gamma}^{(k^*)}(x, h^*(x), \mathcal{U}_2), \tag{4.8}$$

where $\mathcal{L}_{k^*}$ *is as in Meta-Algorithm 5; also,* $\forall n > n^*$, *if* $\mathcal{D}_{XY} \in \mathcal{B}enignNoise(\mathbb{C})$ *and*

$\tau \geq \frac{15}{n} + 7\sqrt{\frac{\ln(4n) + d \ln \frac{2n}{d}}{n}}$, *then on* $\tilde{H}_n$,

$\mathbb{P}(x : \eta(x) \neq 1/2 \text{ and } \exists k \leq k^* \text{ s.t. } \hat{\Delta}^{(k)}(x, \mathcal{U}_2) < 1 - \gamma \text{ and}$

$$\hat{\Gamma}^{(k)}(x, h^*(x), \mathcal{U}_2) \leq \hat{\Gamma}^{(k)}(x, -h^*(x), \mathcal{U}_2)) \leq (d+1)e^{-c''n^{1/3}}, \tag{4.9}$$

for a $\mathbb{C}$- *and* $\mathcal{D}_{XY}$-*dependent finite constant* $c'' > 0$.

Proof. Since most of this lemma discusses only $k = k^*$, in the proof I will simplify the notation by dropping (k^*) superscripts, so that $\hat{\Delta}(\mathcal{U}_1, \mathcal{U}_2)$ abbreviates $\hat{\Delta}^{(k^*)}(\mathcal{U}_1, \mathcal{U}_2)$, $\hat{\Gamma}(x, y, \mathcal{U}_2)$ abbreviates $\hat{\Gamma}^{(k^*)}(x, y, \mathcal{U}_2)$, and so on. I do this only for k^*, and will include the superscripts for any other value of k so that there is no ambiguity.

We begin with (4.6). Recall that $\mathcal{L}_{k^*}$ is initially an independent sample of size $\lfloor n/(6 \cdot 2^{k^*}\hat{\Delta}(\mathcal{U}_1, \mathcal{U}_2)) \rfloor$ sampled from $\mathcal{D}_{XY}[\mathcal{X}]$ (i.e., before we add labels to the examples). Let $\bar{\Delta}(\mathcal{U}_2) = \mathbb{P}(x : \hat{\Delta}(x, \mathcal{U}_2) \geq 1 - \gamma)$.

By Hoeffding's inequality, on an event $H_n^{(1)}(\mathcal{U}_2)$ on $\mathcal{U}_1$ with $\mathbb{P}(\mathcal{U}_1 : H_n^{(1)}(\mathcal{U}_2)) \geq 1 - 2 \cdot$
$exp\{-2m_n^{1/3}\} \geq 1 - 2 \cdot exp\{-2n^{1/3}\}$,

$$|\bar{\Delta}(\mathcal{U}_2) - \frac{1}{m_n} \sum_{z \in \mathcal{U}_1} \mathbb{1}[\hat{\Delta}(z, \mathcal{U}_2) \geq 1 - \gamma]| \leq \frac{1}{m_n^{1/3}},$$

and therefore

$$\bar{\Delta}(\mathcal{U}_2) \leq \hat{\Delta}(\mathcal{U}_1, \mathcal{U}_2).$$

By a Chernoff bound, there is an event $H_n^{(2)}(\mathcal{U}_2)$ on $\mathcal{L}_{k^*}$ and $\mathcal{U}_1$ with

$$\mathbb{P}(\mathcal{L}_{k^*}, \mathcal{U}_1 : H_n^{(2)}(\mathcal{U}_2)) \geq 1 - exp\{-\lfloor n/(6 \cdot 2^{k^*} \bar{\Delta}(\mathcal{U}_2)) \rfloor \bar{\Delta}(\mathcal{U}_2)/3\} \geq 1 - exp\{-(n - 6 \cdot 2^{k^*})/(18 \cdot 2^{k^*})\}$$

such that, on an event $H_n^{(1)}(\mathcal{U}_2) \cap H_n^{(2)}(\mathcal{U}_2)$,

$$|\{x \in \mathcal{L}_{k^*} : \hat{\Delta}(x, \mathcal{U}_2) \geq 1 - \gamma\}| \leq 2\lfloor n/(6 \cdot 2^{k^*} \bar{\Delta}(\mathcal{U}_2)) \rfloor \bar{\Delta}(\mathcal{U}_2) \leq n/(3 \cdot 2^{k^*}).$$

Since the left side of (4.6) is an integer, (4.6) is established.

Next we prove (4.7). If $k^* = 1$, the result clearly holds. In particular, we have $\bar{\Delta}^{(1)}(\mathcal{U}_2) = \mathbb{P}(DIS(V))$, and Hoeffding's inequality implies that on an event with probability $1 - exp\{-2m_n^{1/3}\}$, $\hat{\Delta}^{(1)}(\mathcal{U}_1, \mathcal{U}_2) \leq \mathbb{P}(DIS(V)) + 2m_n^{-1/3}$. Combined with Lemma 4.16, we have bounds of $\Delta_n^{(1)} + 2m_n^{-1/3} = o(1)$.

Otherwise, we have $k^* \geq 2$. In this case, by Hoeffding's inequality and a union bound (over k values), for an event H_n'' over $\mathcal{U}_2$, with $\mathbb{P}(H_n'') \geq 1 - (d+1)exp\{-2\lfloor m_n/(k^* - 1)\rfloor^{1/3}\}$, on $H_n'' \cap H_n'$, for all $k \in \{2, \ldots, k^*\}$ (by Lemma 4.13)

$$M_k \geq \mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1)\lfloor m_n/(k-1)\rfloor - \lfloor m_n/(k-1)\rfloor^{2/3}.$$

Let us name the right side of this inequality $m(n)$. Recall that for $k \leq k^*$,

$$\mathbb{P}(S \in \mathcal{X}^{k-1} : \lim_{r \searrow 0} \mathbb{1}[\mathbb{C}(r) \text{ shatters } S] = 1) > 0$$

by definition of k^*, so $m(n)$ diverges. On event $H_n^{(1)}(\mathcal{U}_2)$,

$$\hat{\Delta}(\mathcal{U}_1, \mathcal{U}_2) \leq \bar{\Delta}(\mathcal{U}_2) + \frac{2}{m_n^{1/3}} \leq \bar{\Delta}(\mathcal{U}_2) + \frac{2}{n^{1/3}}. \tag{4.10}$$

112

Thus, it suffices to bound $\bar{\Delta}(\mathcal{U}_2)$ by a $o(1)$ function. In fact, since we have M_{k^*} lower bounded by a diverging function on $H_n'' \cap H_n'$, so for sufficiently large n, on $H_n' \cap H_n''$,

$$\bar{\Delta}(\mathcal{U}_2) \leq \mathbb{P}(x : \hat{\Delta}(x, \mathcal{U}_2) - M_{k^*}^{-1/3} \geq 1 - (2\gamma + \alpha)/3).$$

Thus, it suffices to bound $\mathbb{P}(x : \hat{\Delta}(x, \mathcal{U}_2) - M_{k^*}^{-1/3} \geq 1 - (2\gamma + \alpha)/3)$ by a $o(1)$ function. On event $H_n \cap H_n' \cap H_n''$, we have that

$\mathbb{P}(x : \hat{\Delta}(x, \mathcal{U}_2) - M_{k^*}^{-1/3} \geq 1 - (2\gamma + \alpha)/3)$

$\leq \mathbb{P}(x : \mathbb{P}(S \in \mathcal{X}^{k^*-1} : V \text{ shatters } S \cup \{x\} | V \text{ shatters } S) \geq 1 - (\gamma + \alpha)/2) +$

$\mathbb{P}(x : |\mathbb{P}(S \in \mathcal{X}^{k^*-1} : V \text{ shatters } S \cup \{x\} | V \text{ shatters } S) - \frac{1}{M_{k^*}} \sum_{i=1}^{\lfloor m/(k^*-1) \rfloor} \mathbb{1}[V \text{ shatters } S_i \cup \{x\}]| > (\alpha - \gamma)/6)$

By Lemma 4.16, on event $H_n \cap H_n'$,

$$\mathbb{P}(x : \mathbb{P}(S \in \mathcal{X}^{k^*-1} : V \text{ shatters } S \cup \{x\} | V \text{ shatters } S) \geq 1 - (\gamma + \alpha)/2) \leq \Delta_n^{(k^*)} = o(1).$$

Thus, it suffices to prove the existence of a $o(1)$ bound on

$\mathbb{P}(x : |\mathbb{P}(S \in \mathcal{X}^{k^*-1} : V \text{ shatters } S \cup \{x\} | V \text{ shatters } S) - \frac{1}{M_{k^*}} \sum_{i=1}^{\lfloor m/(k^*-1) \rfloor} \mathbb{1}[V \text{ shatters } S_i \cup \{x\}]| > (\alpha - \gamma)/6)$

For this, we proceed as follows. Define $\hat{p}_x = \frac{1}{M_{k^*}} \sum_{i=1}^{\lfloor m/(k^*-1) \rfloor} \mathbb{1}[V \text{ shatters } S_i \cup \{x\}]$, a random variable depending on $\mathcal{U}_2$, and $p_x = \mathbb{P}(S \in \mathcal{X}^{k^*-1} : V \text{ shatters } S \cup \{x\} | V \text{ shatters } S)$.

$\mathbb{P}(\mathcal{U}_2 : M_{k^*} \geq m(n) \text{ and } \mathbb{P}(x : |p_x - \hat{p}_x| > (\alpha - \gamma)/6) > M_{k^*}^{-1/3})$

$\leq \mathbb{P}\left(\mathcal{U}_2 : M_{k^*} \geq m(n) \text{ and } \frac{6}{\alpha - \gamma} \mathbb{E}[|p_X - \hat{p}_X|] > M_{k^*}^{-1/3}\right)$ (by Markov's inequality)

$= \sum_{m=m(n)}^{\lfloor m_n/(k^*-1) \rfloor} \mathbb{P}(\mathcal{U}_2 : M_{k^*} = m) \mathbb{P}\left(\mathcal{U}_2 : \mathbb{E}[|p_X - \hat{p}_X|] > m^{-1/3}(\alpha - \gamma)/6 | M_{k^*} = m\right)$

$\leq \sup_{m \geq m(n)} \mathbb{P}\left(\mathcal{U}_2 : exp\{t_m m \mathbb{E}[|p_X - \hat{p}_X|]\} > exp\{t_m m^{2/3}(\alpha - \gamma)/6\} | M_{k^*} = m\right),$

for any values $t_m > 0$. We now proceed as in Chernoff's bounding technique. By Markov's

113

inequality, this last quantity is at most

$$\sup_{m\geq m(n)} \mathbb{E}[e^{t_m m\mathbb{E}[|p_X-\hat{p}_X|]}|M_{k^*}=m]exp\{-t_m m^{2/3}(\alpha-\gamma)/6\}$$

$$\leq \sup_{m\geq m(n)} \mathbb{E}[\mathbb{E}[e^{t_m m|p_X-\hat{p}_X|}]|M_{k^*}=m]exp\{-t_m m^{2/3}(\alpha-\gamma)/6\} \text{ (by Jensen and Fubini)}$$

$$\leq \sup_{m\geq m(n)} (\sup_{p\in[0,1]} \mathbb{E}[e^{t_m\mathbf{B}_{m,p}-t_m mp}] + \sup_{p\in[0,1]} \mathbb{E}[e^{t_m mp-t_m\mathbf{B}_{m,p}}])exp\{-t_m m^{2/3}(\alpha-\gamma)/6\}$$

where $\mathbf{B}_{m,p} \sim$ Binomial(m,p), and the expectation is now over $\mathbf{B}_{m,p}$. By symmetry, if p is the maximizer of the first expectation, then $1-p$ maximizes the second expectation, and the maximizing values are identical, so this is at most

$$2 \sup_{m\geq m(n)} \sup_{p\in[0,1]} \mathbb{E}[exp\{t_m\mathbf{B}_{m,p} - t_m mp\}]exp\{-t_m m^{2/3}(\alpha-\gamma)/6)\}.$$

Following the usual proof for Hoeffding's inequality [see e.g., Devroye et al., 1996], this is at most

$$2 \sup_{m\geq m(n)} exp\{t_m^2 m/8\}exp\{-t_m m^{2/3}(\alpha-\gamma)/6)\}.$$

Taking $t_m = m^{-1/3}2(\alpha-\gamma)/3$, this is

$$2 \sup_{m\geq m(n)} exp\{m^{1/3}(\alpha-\gamma)^2/18 - m^{1/3}2(\alpha-\gamma)^2/18\}$$

$$= 2 \sup_{m\geq m(n)} exp\{-m^{1/3}(\alpha-\gamma)^2/18\} = 2exp\{-m(n)^{1/3}(\alpha-\gamma)^2/18\}.$$

Therefore, there is an event H_n''' on $\mathcal{U}_2$ with

$$\mathbb{P}(H_n''') \geq 1 - 2exp\{-m(n)^{1/3}(\alpha-\gamma)^2/18\} \geq 1-$$

$$2exp\{-(\mathbb{P}(S\in\mathcal{X}^{k^*-1}:\lim_{r\searrow 0}\mathbb{1}[\mathbb{C}(r)\text{ shatters }S]=1)\lfloor n/(k^*-1)\rfloor-\lfloor n/(k^*-1)\rfloor^{2/3})^{1/3}(\alpha-\gamma)^2/18\},$$

such that on $H_n'''\cap H_n''\cap H_n'$,

$$\mathbb{P}(x:|\mathbb{P}(S\in\mathcal{X}^{k^*-1}:V\text{ shatters }S\cup\{x\}|V\text{ shatters }S) - \frac{1}{M_{k^*}}\sum_{i=1}^{\lfloor m/(k^*-1)\rfloor}\mathbb{1}[V\text{ shatters }S_i\cup\{x\}]|>(\alpha-\gamma)/6)$$

$$\leq M_{K^*}^{-1/3} \leq m(n)^{-1/3} = o(1).$$

Finally, we turn to (4.8) and (4.9). If $k=1$, then for $\mathcal{D}_{XY}\in\mathcal{R}ealizable(\mathbb{C})$, we clearly have $h^*\in V$; otherwise, if $\mathcal{D}_{XY}\in\mathcal{B}enignNoise(\mathbb{C})$ and $\tau \geq \frac{15}{n}+7\sqrt{\frac{\ln(4n)+d\ln\frac{2n}{d}}{n}}$, then Lemma 4.12

implies that, on an event over $\mathcal{Z}_{\lfloor n/3 \rfloor}$ of probability $1 - 1/n$, with probability 1 over x such that $\eta(x) \neq 1/2$, if $\hat{\Gamma}^{(1)}(x, y, \mathcal{U}_2) > \hat{\Gamma}^{(1)}(x, -y, \mathcal{U}_2)$, then $y = h^*(x)$. This implies (4.8) for $k^* = 1$ and it covers the $k = 1$ case for (4.9).

Let us now focus on $k \geq 2$ for (4.9), and in particular $k^* \geq 2$ for both (4.9) and (4.8). By Lemma 4.15, for any x in a set of probability 1, Hoeffding's inequality and a union bound (over k values) implies there is an event $H_n^{iv}(x)$ with $\mathbb{P}(\mathcal{U}_2 : H_n^{iv}(x)) \geq 1 - (d+1)exp\{-2m(n)^{1/3}\}$ such that, for $n > n_{\gamma/4}$, on the additional event $H_n^{iv}(x) \cap H_n \cap H_n' \cap H_n''$, if $\eta(x) \neq 1/2$, $\forall k \in \{2, \ldots, k^*\}$,

$$\frac{1}{M_k} \sum_{i=1}^{\lfloor m_n/(k-1) \rfloor} \mathbb{1}[V_{(x,h^*(x))} \text{ does not shatter } S_i^{(k)} \text{ and } V \text{ shatters } S_i^{(k)}]$$

$$\leq \mathbb{P}(S \in \mathcal{X}^{k-1} : V_{(x,h^*(x))} \text{ does not shatter } S | V \text{ shatters } S) + M_k^{-1/3}$$

$$\leq \gamma/4 + M_k^{-1/3} \leq \gamma/4 + m(n)^{-1/3}.$$

For sufficiently large n, $m(n)^{-1/3} < \gamma/4$. If $k \in \{2, \ldots, k^*\}$ and $\hat{\Delta}^{(k)}(x, \mathcal{U}_2) < 1 - \gamma$, then

$$\frac{1}{M_k} \sum_{i=1}^{\lfloor m_n/(k-1) \rfloor} \mathbb{1}[V \text{ does not shatter } S_i^{(k)} \cup \{x\} \text{ and } V \text{ shatters } S_i^{(k)}] > \gamma,$$

and thus, if this happens for sufficiently large n on the event $H_n^{iv}(x) \cap H_n \cap H_n' \cap H_n''$, we must have

115

$$\frac{1}{M_k}\hat{\Gamma}^{(k)}(x, -h^*(x), \mathcal{U}_2) =$$

$$\leq \frac{1}{M_k} \sum_{i=1}^{\lfloor m_n/(k-1) \rfloor} \mathbb{1}\left[V_{(x,h^*(x))} \text{ does not shatter } S_i^{(k)} \text{ and } V \text{ shatters } S_i^{(k)}\right]$$

$$< \gamma/2 = -\gamma/2 + \gamma$$

$$< -\gamma/2 + \frac{1}{M_k} \sum_{i=1}^{\lfloor m_n/(k-1) \rfloor} \mathbb{1}\left[V \text{ does not shatter } S_i^{(k)} \cup \{x\} \text{ and } V \text{ shatters } S_i^{(k)}\right]$$

$$= -\gamma/2 + \frac{1}{M_k} \sum_{i=1}^{\lfloor m_n/(k-1) \rfloor} \mathbb{1}\left[V_{(x,h^*(x))} \text{ does not shatter } S_i^{(k)} \text{ and } V \text{ shatters } S_i^{(k)}\right]$$

$$+ \frac{1}{M_k} \sum_{i=1}^{\lfloor m_n/(k-1) \rfloor} \mathbb{1}\left[V_{(x,h^*(x))} \text{ shatters } S_i^{(k)} \text{ and } V_{(x,-h^*(x))} \text{ does not}\right]$$

$$\leq \frac{1}{M_k} \sum_{i=1}^{\lfloor m_n/(k-1) \rfloor} \mathbb{1}\left[V_{(x,-h^*(x))} \text{ does not shatter } S_i^{(k)} \text{ and } V \text{ shatters } S_i^{(k)}\right]$$

$$= \frac{1}{M_k}\hat{\Gamma}^{(k)}(x, h^*(x), \mathcal{U}_2).$$

By a union bound over the elements of $\mathcal{L}_{k^*}$,

$$\mathbb{P}(\mathcal{U}_2 : \bigcap_{x \in \mathcal{L}_{k^*}} H_n^{iv}(x)) \geq 1 - nm_n^{1/3}(d+1)exp\{-2m(n)^{1/3}\},$$

which suffices to prove (4.8).

Also, we have the following.

$$\mathbb{P}(\mathcal{U}_2 : \mathbb{P}(x : H_n^{iv}(x) \text{ does not occur}) > exp\{-m(n)^{1/3}\})$$

$$\leq exp\{m(n)^{1/3}\}\mathbb{E}[\mathbb{P}(x : H_n^{iv}(x) \text{ does not occur})] \text{ (by Markov's inequality)}$$

$$= exp\{m(n)^{1/3}\}\mathbb{E}[\mathbb{P}(\mathcal{U}_2 : H_n^{iv}(X) \text{ does not occur})] \text{ (by Fubini's theorem)}$$

$$\leq exp\{m(n)^{1/3}\}\mathbb{E}[(d+1)exp\{-2m(n)^{1/3}\}] = (d+1)exp\{-m(n)^{1/3}\}.$$

This suffices to prove (4.9). $\qquad\square$

Proof of Theorem 4.3. The result now follows directly from Lemmas 4.17 and 4.10. (4.7) implies $|\mathcal{L}_{k^*}| \geq L(n)$ for some function $L(n) = \omega(n)$, while (4.6)implies we will infer the labels

for all but at most $\lfloor n/(3 \cdot 2^{k^*}) \rfloor$ of them, and (4.8) implies that, for sufficiently large n, the inferred labels are correct. Lemma 4.10 implies that $er(\hat{h})$ is at most twice the error of any of the $d+1$ classifiers. These things happen on an event that only fails with probability at most $exp\{-c \cdot n^{1/\chi}\}$ for some $\mathcal{D}_{XY}$-dependent constant $c > 0$, and a universal constant $\chi > 0$.

Defining $L^{-1}(m) = \min\{n : L(n) \geq m\}$, we get that, for some distribution over $\ell \in \{L(n), L(n)+1, \ldots\}$ (independent of the data),

$$\mathbb{E}[er(\hat{h})] \leq \mathbb{E}_{\mathcal{Z}}[\mathbb{E}_{\ell}[2er(A_p(\mathcal{Z}_\ell))]] + exp\{-c \cdot n^{1/\chi}\} \leq \sup_{\ell \geq L(n)} \mathbb{E}_{\mathcal{Z}}[2er(A_p(\mathcal{Z}_\ell))] + exp\{-c \cdot n^{1/\chi}\}.$$

Therefore,

$$\bar{\Lambda}_a(3\epsilon, \mathcal{D}_{XY}) \leq L^{-1}(\bar{\Lambda}_p(\epsilon, \mathcal{D}_{XY})) + c^{-\chi} \ln^\chi \frac{1}{\epsilon}.$$

If $\bar{\Lambda}_p(\epsilon, \mathcal{D}_{XY}) \gg 1$, $L^{-1}(\bar{\Lambda}_p(\epsilon, \mathcal{D}_{XY})) = o(\bar{\Lambda}_p(\epsilon, \mathcal{D}_{XY}))$, so $\bar{\Lambda}_p(\epsilon, \mathcal{D}_{XY}) \notin Polylog(1/\epsilon)$ implies the improvements claim, and otherwise $\bar{\Lambda}_a(\epsilon, \mathcal{D}_{XY}) \in Polylog(1/\epsilon)$. $\square$

Proof of Theorem 4.4. This follows identical reasoning to the proof of Theorem 4.3, except that instead of adding $exp\{-c \cdot n^{1/\chi}\}$ to the expected error, we simply take $\Lambda_a(2\epsilon, 2\delta, \mathcal{D}_{XY}) = \max\{L^{-1}(\Lambda_p(\epsilon, \delta, \mathcal{D}_{XY})), c^{-\chi} \ln^\chi(1/\delta)\}$ to ensure the failure probability for the aforementioned events is at most δ. For $\Lambda_p(\epsilon, \delta, \mathcal{D}_{XY}) \gg 1$ this is effectively not a restriction at all for small ϵ, and otherwise we still have $\Lambda_a(\epsilon, 2\delta, \mathcal{D}_{XY}) = O(1)$. $\square$

Lemma 4.18. *Let $\hat{h}$ be the classifier returned by Meta-Algorithm 6, when*
$\tau \geq \frac{15}{n} + 7\sqrt{\frac{\ln(4n) + d \ln \frac{2n}{d}}{n}}$, *and $\mathcal{D}_{XY} \in \mathcal{B}enignNoise(\mathbb{C})$. Then for any $n \in \mathbb{N}$, there is some*
$\mathcal{E}_n = o(n^{-1/2})$ *such that, on an event $\tilde{H}'_n \subseteq \tilde{H}_n$ with $\mathbb{P}(\tilde{H}'_n) \geq \mathbb{P}(\tilde{H}_n) - \delta/2$,*

$$er(\hat{h}) - \nu \leq \mathcal{E}_n.$$

Proof. For brevity, we introduce the notation $\mathbb{Q}_k = \{x : k'(x) > k\}$, where as before $k'(x) = \min\{k' : \hat{\Delta}^{(k')}(x, \mathcal{U}_2) < 1 - \gamma\}$.

First note that, by Alexander's results on uniform convergence [Alexander, 1984, Devroye et al.,

1996], combined with a union bound, on an event $\tilde{H}_n''$ of probability $1 - \delta/2$, every $h \in \mathbb{C}$ has

$$\forall k, |er(h|\mathbb{Q}_k) - er_{Q_k}(h)| \leq \sqrt{\frac{2048d \ln(1024d) + \ln(32(d+1)/\delta)}{|Q_k|}}.$$

Define $\tilde{H}_n' = \tilde{H}_n \cap \tilde{H}_n''$, and for the remainder of the proof we assume this event holds. In particular, this implies every $\hat{h}_k$ has

$$er(\hat{h}_k|\mathbb{Q}_k) \leq \inf_{h \in \mathbb{C}} er(h|\mathbb{Q}_k) + 2\sqrt{\frac{2048d \ln(1024d) + \ln(32(d+1)/\delta)}{|Q_k|}}.$$

Consider any $k \leq k^*$. We have (by Lemma 4.17)

$$
\begin{aligned}
er(\hat{h}_k) = \quad & \mathbb{P}(\mathbb{Q}_k)er(\hat{h}_k|\mathbb{Q}_k) \\
& + \mathbb{P}((x,y) : x \notin \mathbb{Q}_k \text{ and } \eta(x) = 1/2 \text{ and } \hat{h}_k(x) \neq y) \\
& + \mathbb{P}((x,y) : x \notin \mathbb{Q}_k \text{ and } \eta(x) \neq 1/2 \text{ and } \hat{h}_k(x) = h^*(x) \neq y) \\
& + \mathbb{P}((x,y) : x \notin \mathbb{Q}_k \text{ and } \eta(x) \neq 1/2 \text{ and } \hat{h}_k(x) \neq h^*(x) = y) \\
\leq \quad & \mathbb{P}(\mathbb{Q}_k)\left(er(h^*|\mathbb{Q}_k) + 2\sqrt{\frac{2048d \ln(1024d) + \ln(32(d+1)/\delta)}{|Q_k|}}\right) \\
& + (1/2)\mathbb{P}(x : x \notin \mathbb{Q}_k \text{ and } \eta(x) = 1/2) + \\
& \mathbb{P}((x,y) : x \notin \mathbb{Q}_k \text{ and } \eta(x) \neq 1/2 \text{ and } h^*(x) \neq y) + (d+1)e^{-c''n^{1/3}} \\
\leq \quad & \mathbb{P}(\mathbb{Q}_k)\left(er(h^*|\mathbb{Q}_k) + 2\sqrt{\frac{2048d \ln(1024d) + \ln(32(d+1)/\delta)}{|Q_k|}}\right) \\
& + er(h^*|\mathcal{X} \setminus \mathbb{Q}_k)\mathbb{P}(\mathcal{X} \setminus \mathbb{Q}_k) + (d+1)e^{-c''n^{1/3}} \\
\leq \quad & \nu + \mathbb{P}(\mathbb{Q}_k)2\sqrt{\frac{2048d \ln(1024d) + \ln(32(d+1)/\delta)}{\lfloor 2n/(3 \cdot 2^k) \rfloor}} + (d+1)e^{-c''n^{1/3}}.
\end{aligned}
$$

Now there are two cases to consider. In the first case, $k^* \leq \hat{k}$. In this case, we have

$$er(\hat{h}_{\hat{k}}) - er(\hat{h}_{k^*})$$

$$
\begin{aligned}
= \quad & \mathbb{P}(\mathbb{Q}_{k^*})\left(er(\hat{h}_{\hat{k}}|\mathbb{Q}_{k^*}) - er(\hat{h}_{k^*}|\mathbb{Q}_{k^*})\right) \\
\leq \quad & \mathbb{P}(\mathbb{Q}_{k^*})\left(er_{Q_{k^*}}(\hat{h}_{\hat{k}}) - er_{Q_{k^*}}(\hat{h}_{k^*}) + 2\sqrt{\frac{2048d\ln(1024d) + \ln(32(d+1)/\delta)}{|Q_{k^*}|}}\right) \\
\leq \quad & \mathbb{P}(\mathbb{Q}_{k^*})7\sqrt{\frac{2048d\ln(1024d) + \ln(32(d+1)/\delta)}{|Q_{\hat{k}}|}}
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
er(\hat{h}_{\hat{k}}) - \nu \leq \quad & er(\hat{h}_{k^*}) - \nu + \mathbb{P}(\mathbb{Q}_{k^*})7\sqrt{\frac{2048d\ln(1024d) + \ln(32(d+1)/\delta)}{\lfloor 2n/(3\cdot 2^{\hat{k}})\rfloor}} \\
\leq \quad & \mathbb{P}(\mathbb{Q}_{k^*})9\sqrt{\frac{2048d\ln(1024d) + \ln(32(d+1)/\delta)}{\lfloor 2n/(3\cdot 2^{\hat{k}})\rfloor}} + (d+1)e^{-c''n^{1/3}} \\
\leq \quad & \bar{\Delta}_n^{(k^*)}(\mathcal{U}_2)9\sqrt{\frac{2048d\ln(1024d) + \ln(32(d+1)/\delta)}{\lfloor 2n/(3\cdot 2^{\hat{k}})\rfloor}} + (d+1)e^{-c''n^{1/3}} \\
\leq \quad & \breve{\Delta}_n^{(k^*)}9\sqrt{\frac{2048d\ln(1024d) + \ln(32(d+1)/\delta)}{\lfloor 2n/(3\cdot 2^{d+1})\rfloor}} + (d+1)e^{-c''n^{1/3}}.
\end{aligned}
$$

Since $\breve{\Delta}_n^{(k^*)} = o(1)$ (by definition in Lemma 4.17), this last quantity is $o(n^{-1/2})$.

On the other hand, suppose $\hat{k} < k^*$. If $\mathbb{P}(\mathbb{Q}_{\hat{k}}) = 0$, then the aforementioned bound on excess error implies the result. Otherwise, for $k = \hat{k} + 1$, $\exists j \leq \hat{k}$ such that

$$5\sqrt{\frac{2048d\ln(1024d) + \ln(32(d+1)/\delta)}{\lfloor 2n/(3 \cdot 2^k)\rfloor}}$$

$$< \quad er_{Q_j}(\hat{h}_k) - er_{Q_j}(\hat{h}_j)$$

$$\leq \quad er(\hat{h}_k|\mathbb{Q}_j) - er(\hat{h}_j|\mathbb{Q}_j) + 2\sqrt{\frac{2048d\ln(1024d) + \ln(32(d+1)/\delta)}{|Q_j|}}$$

$$= \quad \mathbb{P}((x,y): \hat{h}_k(x) \neq y \text{ and } \eta(x) \neq 1/2|\mathbb{Q}_k)\mathbb{P}(\mathbb{Q}_k|\mathbb{Q}_j)$$

$$+ \mathbb{P}((x,y): \hat{h}_k(x) \neq y \text{ and } \eta(x) \neq 1/2 \text{ and } x \notin \mathbb{Q}_k|x \in \mathbb{Q}_j)$$

$$- \mathbb{P}((x,y): \hat{h}_j(x) \neq y \text{ and } \eta(x) \neq 1/2|x \in \mathbb{Q}_j) + 2\sqrt{\frac{2048d\ln(1024d) + \ln(32(d+1)/\delta)}{|Q_j|}}$$

$$\leq \quad \mathbb{P}(\mathbb{Q}_k|\mathbb{Q}_j)\mathbb{P}((x,y): \hat{h}_k(x) \neq y \text{ and } \eta(x) \neq 1/2|\mathbb{Q}_k)$$

$$+ \mathbb{P}((x,y): \hat{h}_k(x) \neq y \text{ and } \eta(x) \neq 1/2 \text{ and } x \notin \mathbb{Q}_k|x \in \mathbb{Q}_j)$$

$$- \mathbb{P}((x,y): h^*(x) \neq y \text{ and } \eta(x) \neq 1/2|x \in \mathbb{Q}_j) + 2\sqrt{\frac{2048d\ln(1024d) + \ln(32(d+1)/\delta)}{|Q_j|}}$$

$$= \quad \mathbb{P}(\mathbb{Q}_k|\mathbb{Q}_j)(er(\hat{h}_k|\mathbb{Q}_k) - er(h^*|\mathbb{Q}_k))$$

$$+ \mathbb{P}((x,y): \hat{h}_k(x) \neq y \text{ and } \eta(x) \neq 1/2 \text{ and } x \notin \mathbb{Q}_k|x \in \mathbb{Q}_j)$$

$$- \mathbb{P}((x,y): h^*(x) \neq y \text{ and } \eta(x) \neq 1/2 \text{ and } x \notin \mathbb{Q}_k|x \in \mathbb{Q}_j)$$

$$+ 2\sqrt{\frac{2048d\ln(1024d) + \ln(32(d+1)/\delta)}{\lfloor 2n/(3 \cdot 2^j)\rfloor}}$$

$$\leq \quad \mathbb{P}(\mathbb{Q}_k|\mathbb{Q}_j)2\sqrt{\frac{2048d\ln(1024d) + \ln(32(d+1)/\delta)}{\lfloor 2n/(3 \cdot 2^k)\rfloor}}$$

$$+ \mathbb{P}(x: \hat{h}_k(x) \neq h^*(x) \text{ and } \eta(x) \neq 1/2 \text{ and } x \notin \mathbb{Q}_k)/\mathbb{P}(\mathbb{Q}_j)$$

$$+ 2\sqrt{\frac{2048d\ln(1024d) + \ln(32(d+1)/\delta)}{\lfloor 2n/(3 \cdot 2^j)\rfloor}}$$

$$\leq \quad 4\sqrt{\frac{2048d\ln(1024d) + \ln(32(d+1)/\delta)}{\lfloor 2n/(3 \cdot 2^k)\rfloor}} + (d+1)e^{-c''n^{1/3}}/\mathbb{P}(\mathbb{Q}_{\hat{k}})$$

In particular, this implies

$$\mathbb{P}(\mathbb{Q}_{\hat{k}}) \le (d+1)e^{-c''n^{1/3}}\sqrt{\frac{\lfloor 2n/(3\cdot 2^{\hat{k}+1})\rfloor}{2048d\ln(1024d)+\ln(32(d+1)/\delta)}}.$$

Therefore,

$$er(\hat{h}_{\hat{k}}) - \nu \le \mathbb{P}(\mathbb{Q}_{\hat{k}})2\sqrt{\frac{2048d\ln(1024d)+\ln(32(d+1)/\delta)}{\lfloor 2n/(3\cdot 2^{\hat{k}})\rfloor}} + (d+1)e^{-c''n^{1/3}}$$

$$\le (1+\sqrt{2})(d+1)e^{-c''n^{1/3}} = o(n^{-1/2}).$$

□

Proof of Theorem 4.8. This result now follows directly from Lemma 4.18. That is, for sufficiently large n (say $n > s$, for some $s \in \mathbb{N}$), $\mathbb{P}(\tilde{H}_n) \le \delta/2$, so with probability $1 - \delta$, $er(\hat{h}) - \nu \le \mathcal{E}_n$. We can define $\mathcal{E}'_n = 1$ for $n \le s$, and $\mathcal{E}_n$ for $n > s$. Then we have for all n, with probability $1 - \delta$, $er(\hat{h}) - \nu \le \mathcal{E}'_n = o(n^{-1/2})$. Thus, the algorithm obtains a label complexity

$$\Lambda_a(\epsilon + \nu, \delta, \mathcal{D}_{XY}) \le 1 + \sup_{n\in\mathbb{N}} n\mathbb{1}[\mathcal{E}'_n \ge \epsilon].$$

Now define $\mathcal{E}''_n = \mathcal{E}'_n + 2^{-n} = o(n^{-1/2})$. Then

$$\lim_{\epsilon\searrow 0}\epsilon^2\Lambda_a(\epsilon+\nu,\delta,\mathcal{D}_{XY}) \le \lim_{\epsilon\searrow 0}\epsilon^2(1+\sup_{n\in\mathbb{N}} n\mathbb{1}[\mathcal{E}''_n \ge \epsilon])$$

$$= \lim_{\epsilon\searrow 0}\epsilon^2 \sup_{n\in\mathbb{N}, n\ge\lfloor\log_2(1/\epsilon)\rfloor} n\mathbb{1}[\mathcal{E}''_n \ge \epsilon]$$

$$\le \lim_{\epsilon\searrow 0}\epsilon^2 \sup_{n\in\mathbb{N}, n\ge\lfloor\log_2(1/\epsilon)\rfloor} n\frac{(\mathcal{E}''_n)^2}{\epsilon^2}$$

$$= \lim_{\epsilon\searrow 0}\sup_{n\in\mathbb{N}, n\ge\lfloor\log_2(1/\epsilon)\rfloor} n(\mathcal{E}''_n)^2$$

$$= \lim_{n\to\infty}\sup n(\mathcal{E}''_n)^2 = \left(\lim_{n\to\infty}\sup \sqrt{n}\mathcal{E}''_n\right)^2 = 0.$$

Therefore, $\Lambda_a(\epsilon+\nu,\delta,\mathcal{D}_{XY}) = o(1/\epsilon^2)$, as claimed. □

Chapter 5

Beyond Label Requests: A General Framework for Interactive Statistical Learning

In this chapter, I describe a general framework in which a learning algorithm is tasked with learning some concept from a known class by interacting with a teacher via questions. Each question has an arbitrary known cost associated with it, which the learner is required to pay in order to have the question answered. Exploring the information-theoretic limits of this framework, I define a notion called the *cost complexity* of learning, analogous to traditional notions of sample complexity. I discuss this topic for the Exact Learning setting as well as PAC Learning with a pool of unlabeled examples. In the former case, the learner is allowed to ask *any* question, while in the latter case, all questions must concern the target concept's behavior on a set of unlabeled examples. In both settings, I derive upper and lower bounds on the cost complexity of learning, based on a combinatorial quantity I call the *General Identification Cost*.

5.1 Introduction

The ability to ask questions to a knowledgeable teacher can make learning easier. This fact is no secret to any elementary school student. But how much easier? Some questions are more difficult for the teacher to answer than others. How much inconvenience must even the most conscientious learner cause to a teacher in order to learn a concept? This chapter explores these and related questions about the fundamental advantages and limitations of learning by interaction.

In machine learning research, it is becoming increasingly apparent that well-designed interactive learning algorithms can provide valuable improvements in learning performance while reducing the amount of effort required of a human annotator. This research has mainly focused on two formal settings of learning: Exact Learning by queries and pool-based Active PAC Learning. Informally, the objective in the setting of Exact Learning by queries is to perfectly identify a target concept (classifier) by asking questions. In contrast, the pool-based Active PAC setting is concerned only with approximating the concept with high probability with respect to an unknown distribution on the set of possible instances. In this latter setting, the learning algorithm is restricted to asking only questions that relate to the concept's behavior on a particular set of unannotated instances drawn independently from the unknown distribution.

In this chapter, I study both of these active learning settings under a broad definition. Specifically, I consider a learning protocol in which the learner can ask *any* question, but each possible question has an associated *cost*. For example, a query of the form "what is the label of example x" might cost \$1, while a query of the form "show me a positive example" might cost \$10. The objective is to learn the concept while minimizing the total *cost* of queries made. One would like to know how much cost even the most clever learner might be required to pay to learn a concept from a particular concept space in the worst case. This can be viewed as a generalization of notions of *sample complexity* or *query complexity* found in the learning theory literature. I refer to this best worst case cost as the *cost complexity* of learning. This quantity is defined without reference to computational feasibility, focusing instead on the information-theoretic boundaries

of this setting (in the limit of unbounded computation). Below, I derive bounds on the cost complexity of learning, as a function of the concept space and cost function, for both Exact Learning from queries and pool-based Active PAC Learning.

Section 5.2 formally introduces the setting of Exact Learning from queries, describes some related work, and defines cost complexity for that setting. It also serves to introduce the notation and fundamental definitions used throughout this chapter. The section closely parallels the work of Balcázar et al. [Balcázar et al., 2001]. The primary contribution of Section 5.2 is a derivation of upper and lower bounds on the cost complexity of Exact Learning from queries. This is followed, in Section 5.3, by a formal definition of pool-base Active PAC Learning and extension of the notion of cost complexity to that setting. The primary contributions of Section 5.3 include a derivation of upper and lower bounds on the cost complexity of learning in that general setting, as well as an interesting corollary for intersection-closed concept spaces. I know of no previous work giving general results of this type.

5.2   Active Exact Learning

In this setting, there is an *instance space* $\mathcal{X}$ and *concept space* $\mathbb{C}$ on $\mathcal{X}$ such that any $h \in \mathbb{C}$ is a distinct function $h : \mathcal{X} \to \{0, 1\}$.[1] Additionally, define $\mathbb{C}^* = \{h : \mathcal{X} \to \{0, 1\}\}$. That is, $\mathbb{C}^*$ is the *most general* concept space, containing all possible labelings of $\mathcal{X}$. In particular, any concept space $\mathbb{C}$ is a subset of $\mathbb{C}^*$. For a particular learning problem, there is an unknown *target concept* $f \in \mathbb{C}$, and the task is to identify f using a teacher's answers to queries made by the learning algorithm. Formally, an *actual query* is any function in $\tilde{Q} = \{\tilde{q} : \mathbb{C}^* \to 2^{\mathcal{A}^*} \setminus \{\varnothing\}\}$,[2] for some *answer set* $\mathcal{A}^*$. By a learning algorithm "making an actual query", I mean that it selects

---

[1] All of the main results easily generalize to multiclass as well.

[2] The restriction that $\tilde{q}(f) \neq \{\}$ is a bit like an assumption that every valid question has at least one answer for any target concept. However, we can always define some particular answer to mean "there is no answer," so this restriction is really more of a notational convenience than an assumption.

a function $\tilde{q} \in \tilde{Q}$, passes it to the teacher, and the teacher returns a single *answer* $\tilde{a} \in \tilde{q}(f)$ where f is the target concept. A concept $h \in \mathbb{C}^*$ is *consistent* with an answer $\tilde{a}$ to an actual query $\tilde{q}$ if $\tilde{a} \in \tilde{q}(h)$. Thus, I assume the teacher always returns an answer that the target concept is consistent with; however, when there are multiple such answers, the teacher may arbitrarily select from amongst them.

Traditionally, the subject of active learning has been studied with respect to specific restricted query types, such as membership queries, and the learning algorithm's objective has been to minimize the *number* of queries used to learn. However, it is often the case that learning with these simple types of queries is difficult, but if the learning algorithm is allowed just a few *special* queries, learning becomes significantly easier. The reason we are initially reluctant to allow the learner to ask certain types of queries is that these queries are difficult, expensive, or sometimes impossible to answer. However, we can incorporate this difficulty level into the framework by assigning each query type a specific *cost*, and then allowing the learning algorithm to explicitly optimize the *cost* needed to learn, rather than the *number* of queries. In addition to allowing the algorithm to trade off between different types of queries, this also gives us the added flexibility to specify different costs within the same family (e.g., perhaps some membership queries are more expensive than others).

Formally, in this framework there is a *cost function*. Let $\alpha > 0$ be a constant. A cost function is any $c : \tilde{Q} \to (\alpha, \infty]$. In practice, c would typically be defined by the user responsible for answering the queries, and could be based on the time, resources, or operating expenses necessary to obtain the answer. Note that if a particular type of query is unanswerable for a particular application, or if the user wishes to work with a reduced set of possible queries, one can always define the costs of those undesirable query types to be ∞, so that any reasonable learning algorithm ignores them if possible.

While the notion of *actual query* closely corresponds to the actual mechanism of querying in practice, it will be more convenient to work with the information-theoretic implications of these

queries. Define the set of *effective queries* $\mathcal{Q} = \{q : \mathbb{C}^* \to 2^{2^{\mathbb{C}^*}} \setminus \{\varnothing\} | \forall f \in \mathbb{C}^*, a \in q(f) \Rightarrow [f \in a \wedge \forall h \in a, a \in q(h)]\}$. Each effective query corresponds to an equivalence class of actual queries, defined by mapping any answer to the set of concepts consistent with it. We can thus define the mapping

$$\mathcal{E}(q) = \{\tilde{q} | \tilde{q} \in \tilde{Q}, \forall f \in \mathbb{C}^*, [\exists \tilde{a} \in \tilde{q}(f) \text{ with } a = \{h | h \in \mathbb{C}^*, \tilde{a} \in \tilde{q}(h)\}] \Leftrightarrow a \in q(f)\}.$$

By an algorithm "making an effective query q," I mean that it makes an actual query in $\mathcal{E}(q)$,[3] (a good algorithm will pick a cheaper actual query). For the purpose of this best-worst-case analysis, the following definition is appropriate. For a cost function c, define a corresponding *effective cost function* (overloading notation) $c : \mathcal{Q} \to [\alpha, \infty]$, such that $\forall q \in \mathcal{Q}, c(q) = \inf_{\tilde{q} \in \mathcal{E}(q)} c(\tilde{q})$. The following definitions illustrate how query types can be defined using effective queries.

A *positive example query* is any $\tilde{q} \in \mathcal{E}(q_S)$ for some $S \subseteq \mathcal{X}$, such that $q_S \in \mathcal{Q}$ is defined by $\forall f \in \mathbb{C}^*$ s.t. $[\exists x \in S : f(x) = 1], q_S(f) = \{\{h | h \in \mathbb{C}^*, h(x) = 1\} | x \in S : f(x) = 1\}$, and $\forall f \in \mathbb{C}^*$ s.t. $[\forall x \in S, f(x) = 0], q_S(f) = \{\{h | h \in \mathbb{C}^* : \forall x \in S, h(x) = 0\}\}$.

A *membership query* is any $\tilde{q} \in \mathcal{E}(q_{\{x\}})$ for some $x \in \mathcal{X}$. This special case of a positive example query can equivalently be defined by $\forall f \in \mathbb{C}^*, q_{\{x\}}(f) = \{\{h | h \in \mathbb{C}^*, h(x) = f(x)\}\}$. These effectively correspond to asking for any example labeled 1 in S or an indication that there are none (positive example query), and asking for the label of a particular example in $\mathcal{X}$ (membership query). I will refer to these two query types in subsequent examples, but the reader should keep in mind that the theorems below apply to *all* types of queries.

Additionally, it will be useful to have a notion of an *effective oracle*, which is an unknown function defining how the teacher will answer the various queries. Formally, an effective oracle T is any function in $\mathcal{T} = \{T : \mathcal{Q} \to 2^{\mathbb{C}^*} | \forall q \in \mathcal{Q}, T(q) \in \cup_{f \in \mathbb{C}^*} q(f)\}$.[4] For convenience, I also

---

[3]I assume $\mathcal{A}^*$ is sufficiently expressive so that $\forall q \in \mathcal{Q}, \mathcal{E}(q) \neq \varnothing$; alternatively, we could define $\mathcal{E}(q) = \varnothing \Rightarrow c(q) = \infty$ without sacrificing the main theorems. Additionally, I will assume that it is possible to find an actual query in $\mathcal{E}(q)$ with cost arbitrarily close to $\inf_{\tilde{q} \in \mathcal{E}(q)} c(\tilde{q})$ for any $q \in \mathcal{Q}$ using finite computation.

[4]An effective oracle corresponds to a deterministic stateless teacher, which gives up as little information as

overload this notation, defining for a set of queries $R \subseteq Q$, $T(R) = \cap_{q \in R} T(q)$.

Definition 5.1. *A* learning algorithm $\mathcal{A}$ for $\mathbb{C}$ using cost function c *is any algorithm which, for any (unknown) target concept $f \in \mathbb{C}$, by a finite number of finite cost actual queries, is guaranteed to reduce the set of concepts in $\mathbb{C}$ consistent with the answers to precisely $\{f\}$. A concept space $\mathbb{C}$ is* learnable *with cost function c using total cost t if there exists a learning algorithm for $\mathbb{C}$ using c guaranteed to have the sum of costs of the queries it makes at most t.*

Definition 5.2. *For any instance space $\mathcal{X}$, concept space $\mathbb{C}$ on $\mathcal{X}$, and cost function c, define the* cost complexity*, denoted CostComplexity$(\mathbb{C}, c)$, as the infimum $t \geq 0$ such that $\mathbb{C}$ is learnable with cost function c using total cost no greater than t.*

---

[5]Equivalently, we can define cost complexity using the following recurrence. If $|\mathbb{C}| = 1$, $CostComplexity(\mathbb{C}, c) = 0$. Otherwise,

$$CostComplexity(\mathbb{C}, c) = \inf_{\tilde{q} \in \tilde{Q}} c(\tilde{q}) + \max_{f \in \mathbb{C}, \tilde{a} \in \tilde{q}(f)} CostComplexity(\{h | h \in \mathbb{C}, \tilde{a} \in \tilde{q}(h)\}, c)$$

Since

$$\inf_{\tilde{q} \in \tilde{Q}} c(\tilde{q}) + \max_{f \in \mathbb{C}, \tilde{a} \in \tilde{q}(f)} CostComplexity(\{h | h \in \mathbb{C}, \tilde{a} \in \tilde{q}(h)\}, c)$$

$$= \inf_{q \in Q} \inf_{\tilde{q} \in \mathcal{E}(q)} c(\tilde{q}) + \max_{f \in \mathbb{C}, \tilde{a} \in \tilde{q}(f)} CostComplexity(\mathbb{C} \cap \{h | h \in \mathbb{C}^*, \tilde{a} \in \tilde{q}(h)\}, c)$$

$$= \inf_{q \in Q} c(q) + \max_{f \in \mathbb{C}, a \in q(f)} CostComplexity(\mathbb{C} \cap a, c),$$

we can equivalently define cost complexity in terms of *effective queries* and *effective cost*. That is, *CostComplexity*$(\mathbb{C}, c)$ is the infimum $t \geq 0$ such that there is an algorithm guaranteed to identify any $f \in \mathbb{C}$ using *effective* queries with total of *effective* costs no greater than t.

possible. It is also possible to analyze a setting in which asking two queries from the same equivalence class, or asking the same question twice, can possibly lead to two different answers. However, the worst case in both settings is identical, so the worst case results obtained for this setting also apply to the more general case.

[5]I have made the dependence of $\mathcal{A}$ on the teacher implicit. To be formally correct, $\mathcal{A}$ should have the teacher's effective oracle T as input, and is guaranteed to output f for any $T \in \mathcal{T}$ s.t. $\forall q \in Q$, $T(q) \in q(f)$. Cost is then a book-keeping device recording how $\mathcal{A}$ uses T during execution.

5.2.1 Related Work

There have been a relatively large number of contributions to the study of Exact Learning from queries. In particular, much interest has been given to settings in which the learning algorithm is restricted to a few specific types of queries (e.g. membership queries and equivalence queries). However, these contributions focus entirely on the *number* of queries needed, rather than *cost*. The most relevant work in this area is by Balcázar, Castro, and Guijarro [Balcázar et al., 2001]. Prior to publication of [Balcázar et al., 2002], there were a variety of publications in which the learning algorithm could use some specific set of queries, and which derived bounds on the number of queries any algorithm might be required to make in the worst case in order to learn. For example, [Hellerstein et al., 1996] analyzed the combination of membership and proper equivalence queries, [Hegedüs, 1995] additionally analyzed learning from membership queries alone, while [Balcázar et al., 1999] considered learning from just proper equivalence queries. Amidst these various special case analyses, somewhat surprisingly, Balcázar et al. [Balcázar et al., 2002] discovered that the query complexity bounds derived in these works were all special cases of a single general theorem, applying to the broad class of *sample-based queries*. They further generalized this result in [Balcázar et al., 2001], giving results that apply to any combination of *any* query types. That work defines an abstract combinatorial quantity, which they call the *General Dimension*, which provides a lower bound on the query complexity, and is within a log factor of it. Furthermore, the General Dimension can actually be computed for a variety of interesting combinations of query types. Until now there has not been any analysis I know of that considers learning with *all* query types, but giving each query a cost, and bounding the worst-case *cost* that a learning algorithm might be required to incur. In particular, the analysis of the next subsection can be viewed as a generalization of [Balcázar et al., 2001] to add this notion of cost, such that [Balcázar et al., 2001] represents the special case of cost that is uniformly 1 on a particular set of queries and ∞ on all other queries.

5.2.2 Cost Complexity Bounds

I now turn to the subject of exploring the fundamental limits of interactive learning in terms of cost. This discussion closely parallels that of Balcázar, Castro, and Guijarro [Balcázar et al., 2001].

Definition 5.3. *For any instance space $\mathcal{X}$, concept space $\mathbb{C}$ on $\mathcal{X}$, and cost function c, define the* General Identification Cost*, denoted $GIC(\mathbb{C}, c)$, as follows.*

$$GIC(\mathbb{C}, c) = \inf\{t | t \geq 0, \forall T \in \mathcal{T}, \exists R \subseteq \mathcal{Q}, s.t. [\textstyle\sum_{q \in R} c(q) \leq t] \wedge [|\mathbb{C} \cap T(R)| \leq 1]\}$$

We can also express this as $GIC(\mathbb{C}, c) = \sup_{T \in \mathcal{T}} \inf_{R \subseteq \mathcal{Q}: |\mathbb{C} \cap T(R)| \leq 1} \sum_{q \in R} c(q)$. Note that calculating this corresponds to a much simpler optimization problem than calculating the cost complexity. The General Identification Cost is a direct generalization of the General Dimension of [Balcázar et al., 2001], which itself generalizes quantities such as Extended Teaching Dimension [Hegedüs, 1995], Strong Consistency Dimension [Balcázar et al., 1999], and the Certificate Sizes of [Hellerstein et al., 1996]. It can be interpreted as a sort of game. This game is similar to the usual setting, except that the teacher's answers are not restricted to be consistent with a concept. Imagine there is a helpful spy who knows precisely how the teacher will respond to every query. The spy is able to suggest queries to the learner, and wishes to cause the learner to pay as little as possible. If the spy is sufficiently clever at suggesting queries, and the learner follows every suggestion by the spy, then after asking some minimal cost set of queries the learner can narrow the set of concepts in $\mathbb{C}$ consistent with the answers down to at most one. The General Identification Cost is precisely the worst case limiting cost the learner might be forced to pay during this process, no matter how clever the spy is at suggesting queries.

Lemma 5.4. *For any instance space $\mathcal{X}$, concept space $\mathbb{C}$ on $\mathcal{X}$, and cost function c, if $V \subseteq \mathbb{C}$, then $GIC(V, c) \leq GIC(\mathbb{C}, c)$.*

Proof. It clearly holds if $GIC(\mathbb{C}, c) = \infty$. If $GIC(\mathbb{C}, c) < k$, then $\forall T \in \mathcal{T}, \exists R \subseteq \mathcal{Q}$ s.t. $\sum_{q \in R} c(q) < k$ and $1 \geq |\mathbb{C} \cap T(R)| \geq |V \cap T(R)|$, and therefore $GIC(V, c) < k$. The limit as $k \to GIC(\mathbb{C}, c)$ gives the result. $\square$

Lemma 5.5. *For any $\gamma > 0$, instance space $\mathcal{X}$, finite concept space $\mathbb{C}$ on $\mathcal{X}$ with $|\mathbb{C}| > 1$, and cost function c such that $GIC(\mathbb{C}, c) < \infty$, $\exists q \in \mathcal{Q}$ such that $\forall T \in \mathcal{T}$,*

$$|\mathbb{C} \setminus T(q)| \geq c(q)\frac{|\mathbb{C}| - 1}{GIC(\mathbb{C}, c) + \gamma}.$$

That is, regardless of which answer the teacher picks, there are at least $c(q)\frac{|\mathbb{C}|-1}{GIC(\mathbb{C},c)+\gamma}$ concepts in $\mathbb{C}$ inconsistent with the answer.

Proof. Suppose $\forall q \in \mathcal{Q}, \exists T_q \in \mathcal{T}$ such that $|\mathbb{C} \setminus T_q(q)| < c(q)\frac{|\mathbb{C}|-1}{GIC(\mathbb{C},c)+\gamma}$. Then define an effective oracle T with the property that $\forall q \in \mathcal{Q}, T(q) = T_q(q)$. We have thus defined an oracle such that $\forall R \subseteq \mathcal{Q}, \sum_{q \in R} c(q) \leq GIC(\mathbb{C}, c) + \gamma \Rightarrow$

$$|\mathbb{C} \cap T(R)| = |\mathbb{C}| - |\mathbb{C} \setminus T(R)| \geq |\mathbb{C}| - \sum_{q \in R} |\mathbb{C} \setminus T_q(q)|$$

$$> |\mathbb{C}| - \sum_{q \in R} c(q)\frac{|\mathbb{C}| - 1}{GIC(\mathbb{C}, c) + \gamma} \geq |\mathbb{C}| - (GIC(\mathbb{C}, c) + \gamma)\frac{|\mathbb{C}| - 1}{GIC(\mathbb{C}, c) + \gamma} = 1.$$

In particular, this contradicts the definition of $GIC(\mathbb{C}, c)$. $\qquad\square$

This brings us to the main theorem of this section.

Theorem 5.6. *For any instance space $\mathcal{X}$, concept space $\mathbb{C}$ on $\mathcal{X}$, and cost function c,*

$$GIC(\mathbb{C}, c) \leq \textit{CostComplexity}(\mathbb{C}, c) \leq GIC(\mathbb{C}, c)\log_2 |\mathbb{C}|$$

Proof. I begin with the lower bound. Let $k < GIC(\mathbb{C}, c)$. By definition of GIC, $\exists T \in \mathcal{T}$, such that $\forall R \subseteq \mathcal{Q}, \sum_{q \in R} c(q) \leq k \Rightarrow |\mathbb{C} \cap T(R)| > 1$. In particular, this implies that an adversarial teacher can answer any sequence of queries with cost no greater than k in a way that leaves at least 2 concepts in $\mathbb{C}$ consistent with the answers, either of which could be the target concept f. This implies *CostComplexity*$(\mathbb{C}, c) > k$. The limit as $k \to GIC(\mathbb{C}, c)$ gives the bound.

Next I prove the upper bound. If $GIC(\mathbb{C}, c) = \infty$ or $|\mathbb{C}| = \infty$, the bound holds vacuously, so let us assume these are finite. Say the teacher's answers correspond to some effective oracle

$T \in \mathcal{T}$. Consider a recursive algorithm $\mathcal{A}_\gamma$ that makes effective queries from $\mathcal{Q}$.[6] If $|\mathbb{C}| = 1$, then $\mathcal{A}_\gamma$ halts and outputs the single remaining concept. Otherwise, let q be an effective query having the property guaranteed by Lemma 5.5. That is, $|\mathbb{C} \setminus T(q)| \geq c(q) \frac{|\mathbb{C}|-1}{GIC(\mathbb{C},c)+\gamma}$. Defining $V = \mathbb{C} \cap T(q)$ (a generalized notion of *version space*), this implies that $c(q) \leq (GIC(\mathbb{C}, c) + \gamma) \frac{|\mathbb{C}|-|V|}{|\mathbb{C}|-1}$ and $|V| < |\mathbb{C}|$. Say $\mathcal{A}_\gamma$ makes effective query q, and then recurses on V. In particular, we can immediately see that this algorithm identifies f using no more than $|\mathbb{C}| - 1$ queries.

I now prove by induction on $|\mathbb{C}|$ that *CostComplexity*$(\mathbb{C}, c) \leq (GIC(\mathbb{C}, c) + \gamma) H_{|\mathbb{C}|-1}$, where $H_n = \sum_{i=1}^{n} \frac{1}{i}$ is the nth harmonic number. If $|\mathbb{C}| = 1$, then the cost complexity is 0. For $|\mathbb{C}| > 1$,

CostComplexity$(\mathbb{C}, c)$

$$
\begin{aligned}
&\leq c(q) + \textit{CostComplexity}(V, c) \\
&\leq (GIC(\mathbb{C}, c) + \gamma) \frac{|\mathbb{C}| - |V|}{|\mathbb{C}| - 1} + (GIC(V, c) + \gamma) H_{|V|-1} \\
&\leq (GIC(\mathbb{C}, c) + \gamma) \left(\frac{|\mathbb{C}| - |V|}{|\mathbb{C}| - 1} + H_{|V|-1} \right) \\
&\leq (GIC(\mathbb{C}, c) + \gamma) H_{|\mathbb{C}|-1}
\end{aligned}
$$

where the second inequality uses the inductive hypothesis along with the properties of q guaranteed by Lemma 5.5, and the third inequality uses Lemma 5.4. Finally, noting that $H_{|\mathbb{C}|-1} \leq \log_2 |\mathbb{C}|$ and taking the limit as $\gamma \to 0$ proves the theorem. $\qquad \square$

One interesting implication of this proof is that the greedy algorithm that chooses q to maximize $\min_{T \in \mathcal{T}} \frac{|\mathbb{C} \setminus T(q)|}{c(q)}$ has a cost complexity within a $\log_2 |\mathbb{C}|$ factor of optimal.

---

[6]I use the definition of cost complexity in terms of effective cost, so that we need not concern ourselves with how $\mathcal{A}_\gamma$ chooses its *actual queries*. However, we could define $\mathcal{A}_\gamma$ to make actual queries with cost within γ of the effective query cost, so that the result still holds as $\gamma \to 0$.

5.2.3   An Example: Discrete Intervals

As a simple example of cost complexity, consider $\mathcal{X} = \{1, 2, \ldots, N\}$, for $N \geq 4$, $\mathbb{C} = \{h_{a,b} : \mathcal{X} \to \{0, 1\} | a, b \in \mathcal{X}, a \leq b, \forall x \in \mathcal{X}, [a \leq x \leq b \Leftrightarrow h_{a,b}(x) = 1]\}$, and define an effective cost function c that is 1 for membership queries $q_{\{x\}}$ for any $x \in \mathcal{X}$, k for the positive example query $q_{\mathcal{X}}$ where $3 \leq k \leq N - 1$, and ∞ for any other queries. In this case, $GIC(\mathbb{C}, c) = k + 1$. In the spy game, say the teacher answers effective queries with an effective oracle T. Let $\mathcal{X}_+ = \{x | x \in \mathcal{X}, T(q_{\{x\}}) = \{h | h \in \mathbb{C}^*, h(x) = 1\}\}$. If $\mathcal{X}_+ \neq \varnothing$, then let $a = \min \mathcal{X}_+$ and $b = \max \mathcal{X}_+$. The spy tells the learner to make queries $q_{\{a\}}, q_{\{b\}}, q_{\{a-1\}}$ (if $a > 1$), and $q_{\{b+1\}}$ (if $b < N$). This narrows the version space to $\{h_{a,b}\}$, at a worst-case effective cost of 4. If $\mathcal{X}_+ = \varnothing$, then the spy suggests query $q_{\mathcal{X}}$. If $T(q_{\mathcal{X}}) = \{f_-\}$, the "all 0" concept, then no concepts in $\mathbb{C}$ are consistent. Otherwise, $T(q_{\mathcal{X}}) = \{h | h \in \mathbb{C}^*, h(x) = 1\}$ for some $x \in \mathcal{X}$, and the spy suggests membership query $q_{\{x\}}$. In this case, $T(q_{\{x\}}) \cap T(q_{\mathcal{X}}) = \varnothing$, so the worst-case cost is $k + 1$ (without $q_{\mathcal{X}}$, it would cost $N - 1$). These are the only cases to consider, so $GIC(\mathbb{C}, c) = k + 1$. By Theorem 5.6, this implies

$k + 1 \leq CostComplexity(\mathbb{C}, c) \leq 2(k + 1) \log_2 N$.

We can slightly improve this by noting that we only use $q_{\mathcal{X}}$ once. Specifically, if a learning algorithm begins (in the regular setting) by asking $q_{\mathcal{X}}$, revealing that $f(x) = 1$ for some $x \in \mathcal{X}$, then we can reduce to two disjoint learning problems, with concept spaces $\mathbb{C}'_1 = \{h_{x,b} | b \in \{x, \ldots, N\}\}$, and $\mathbb{C}'_2 = \{h_{a,x} | a \in \{1, 2, \ldots, x\}\}$, with cost functions $c_1(q) = c(q)$ for $q \in \{q_{\{x\}}, q_{\{x+1\}}, \ldots, q_{\{N\}}\}$ and ∞ otherwise, and $c_2(q) = c(q)$ for $q \in \{q_{\{1\}}, q_{\{2\}}, \ldots, q_{\{x\}}\}$ and ∞ otherwise, and corresponding $GIC(\mathbb{C}'_1, c) \leq 2$, $GIC(\mathbb{C}'_2, c) \leq 2$. So we can say that

$CostComplexity(\mathbb{C}, c) \leq k + CostComplexity(\mathbb{C}'_1, c_1) + CostComplexity(\mathbb{C}'_2, c_2) \leq k + 4 \log_2 N$.

One algorithm that achieves this begins by making the positive example query, and then performs binary search above and below the indicated positive example to find the boundaries.

5.3 Pool-Based Active PAC Learning

In many scenarios, a more realistic definition of learning is that supplied by the Probably Approximately Correct (PAC) model. In this case, unlike the previous section, we are interested only in discovering with high probability a function with behavior very *similar* to the target concept on examples sampled from some distribution. Formally, as above there is an instance space $\mathcal{X}$, and a concept space $\mathbb{C} \subseteq \mathbb{C}^*$ on $\mathcal{X}$; unlike above, there is also a distribution $\mathcal{D}$ over $\mathcal{X}$. As with Exact Learning, the learning algorithm interacts with a teacher by making queries. However, in this setting the learning algorithm is given as input a finite sequence[7] of unlabeled examples $\mathcal{U}$, each drawn independently according to $\mathcal{D}$, and *all queries* made by the algorithm must concern only the behavior of the target concept on examples in $\mathcal{U}$. Formally, a *data-dependent cost function* is any function $c : \tilde{Q} \times 2^{\mathcal{X}} \to (\alpha, \infty]$. For a given set of unlabeled examples $\mathcal{U}$, and data-dependent cost function c, define $c_{\mathcal{U}}(\cdot) = c(\cdot, \mathcal{U})$. Thus, $c_{\mathcal{U}}$ is a cost function in the sense of the previous section. For a given $c_{\mathcal{U}}$, the corresponding effective cost function $c_{\mathcal{U}} : \mathcal{Q} \to [\alpha, \infty]$ is defined as in the previous section.

Definition 5.7. *Let $\mathcal{X}$ be an instance space, $\mathbb{C}$ a concept space on $\mathcal{X}$, and $\mathcal{U} = (x_1, x_2, \ldots, x_{|\mathcal{U}|})$ a finite sequence of unlabeled examples. Define $\forall h \in \mathbb{C}, h(\mathcal{U}) = (h(x_1), h(x_2), \ldots, h(x_{|\mathcal{U}|}))$. Define $\mathbb{C}[\mathcal{U}] \subseteq \mathbb{C}$ as any concept space such that $\forall h \in \mathbb{C}, |\{h'|h' \in \mathbb{C}[\mathcal{U}], h'(\mathcal{U}) = h(\mathcal{U})\}| = 1$.*

---

[7]I will implicitly overload all notation for sets and sequences, so that if a set is used where a sequence is required, then an arbitrary ordering of the set is implied (though this ordering should be used consistently), and if a sequence is used where a set is required, then the set of distinct elements of the sequence is implied.

Definition 5.8. *A* sample-based cost function *is any data-dependent cost function c such that for all finite $\mathcal{U} \subseteq \mathcal{X}, \forall q \in \mathcal{Q}$,*

$$c_{\mathcal{U}}(q) < \infty \Rightarrow \forall f \in \mathbb{C}^*, \forall a \in q(f), \forall h \in \mathbb{C}^*, [h(\mathcal{U}) = f(\mathcal{U}) \Rightarrow h \in a].$$

This corresponds to queries that are about *the target concept's labels on some subset of $\mathcal{U}$. Additionally, $\forall \mathcal{U} \subseteq \mathcal{X}, x \in \mathcal{X}$, and $q \in \mathcal{Q}$, $c(q, \mathcal{U} \cup \{x\}) \le c(q, \mathcal{U})$. That is, in addition to the above property, adding extra examples to which q's answers do not refer does not increase its cost.*

For example, membership queries on $x \in \mathcal{U}$ and positive examples queries on $S \subseteq \mathcal{U}$ could have finite costs under a sample-based cost function. As in the previous section, there is a target concept $f \in \mathbb{C}$, but unlike that section, we do not try to *identify* f, but instead attempt to *approximate* it with high probability.

Definition 5.9. *For instance space $\mathcal{X}$, concept space $\mathbb{C}$ on $\mathcal{X}$, distribution $\mathcal{D}$ on $\mathcal{X}$, target concept $f \in \mathbb{C}$, and concept $h \in \mathbb{C}$, define the* error rate *of h, denoted $error_{\mathcal{D}}(h, f)$, as*

$$error_{\mathcal{D}}(h, f) = \mathcal{P}r_{X \sim \mathcal{D}} \{h(X) \neq f(X)\}$$

Definition 5.10. *For $(\epsilon, \delta) \in (0, 1)^2$, an (ϵ, δ)-learning algorithm for $\mathbb{C}$ using sample-based cost function c is any algorithm $\mathcal{A}$ taking as input a finite sequence of unlabeled examples, such that for any target concept $f \in \mathbb{C}$ and finite sequence $\mathcal{U}$, $\mathcal{A}(\mathcal{U})$ outputs a concept in $\mathbb{C}$ after making a finite number of actual queries with finite costs under $c_{\mathcal{U}}$. Additionally, any (ϵ, δ)-learning algorithm $\mathcal{A}$ has the property that $\exists m \in [0, \infty)$ such that, for any target concept $f \in \mathbb{C}$ and distribution $\mathcal{D}$ on $\mathcal{X}$,*

$$\mathcal{P}r_{\mathcal{U} \sim \mathcal{D}^m} \{error_{\mathcal{D}}(\mathcal{A}(\mathcal{U}), f) > \epsilon\} \le \delta.$$

A concept space $\mathbb{C}$ is (ϵ, δ)-learnable given sample-based cost function c using total cost t if there exists an (ϵ, δ)-learning algorithm $\mathcal{A}$ for $\mathbb{C}$ using c such that for all finite example sequences $\mathcal{U}$, $\mathcal{A}(\mathcal{U})$ is guaranteed to have the sum of costs of the queries it makes at most t under $c_{\mathcal{U}}$.

> **Definition 5.11.** *For any instance space $\mathcal{X}$, concept space $\mathbb{C}$ on $\mathcal{X}$, sample-based cost function c, and $(\epsilon, \delta) \in (0, 1)^2$, define the (ϵ, δ)-cost complexity, denoted $CostComplexity(\mathbb{C}, c, \epsilon, \delta)$, as the infimum $t \geq 0$ such that $\mathbb{C}$ is (ϵ, δ)-learnable given c using total cost no greater than t.*

As in the previous section, because it is the *limiting* case, we can equivalently define the (ϵ, δ)-cost complexity as the infimum $t \geq 0$ such that there is an (ϵ, δ)-learning algorithm guaranteed to have the sum of *effective* costs of the *effective* queries it makes at most t. The main results from this section include a new combinatorial quantity $GPIC(\mathbb{C}, c, m, \tau)$ such that if d is the VC-dimension of $\mathbb{C}$, then

$$GPIC(\mathbb{C}, c, \Theta(\tfrac{1}{\epsilon}), \delta) \leq CostComplexity(\mathbb{C}, c, \epsilon, \delta) \leq GPIC(\mathbb{C}, c, \tilde{\Theta}\left(\tfrac{d}{\epsilon}\right), 0)\tilde{\Theta}(d).$$

5.3.1 Related Work

Previous work on pool-based active learning in the PAC model has been restricted almost exclusively to uniform-cost membership queries on examples in the unlabeled set $\mathcal{U}$. There has been some recent progress on query complexity bounds for that restricted setting. Specifically, Dasgupta [Dasgupta, 2004] analyzes a greedy active learning scheme and derives bounds for the number of membership queries in $\mathcal{U}$ it uses under an *average case* setting, in which the target concept is selected randomly from a known distribution. A similar type of analysis was previously given by Freund et al. [Freund et al., 1997] to prove positive results for the Query by Committee algorithm. In a subsequent paper, Dasgupta [Dasgupta, 2005] derives upper and lower bounds on the number of membership queries in $\mathcal{U}$ required for active learning for any particular distribution $\mathcal{D}$, under the assumption that $\mathcal{D}$ is known. The results I derive in this section imply *worst-case* results (over both $\mathcal{D}$ and f) for this as a special case of more general bounds applying to *any* sample-based cost function.

5.3.2 Cost Complexity Upper Bounds

I now derive bounds on the cost complexity of pool-based Active PAC Learning.

Definition 5.12. *For an instance space $\mathcal{X}$, concept space $\mathbb{C}$ on $\mathcal{X}$, sample-based cost function c, and nonnegative integer m, define the* General Identification Cost Growth Function, *denoted $GIC(\mathbb{C}, c, m)$, as follows.*

$$GIC(\mathbb{C}, c, m) = \sup_{\mathcal{U} \in \mathcal{X}^m} GIC(\mathbb{C}[\mathcal{U}], c_{\mathcal{U}})$$

Definition 5.13. *For any instance space $\mathcal{X}$, concept space $\mathbb{C}$ on $\mathcal{X}$, and $(\epsilon, \delta) \in (0, 1)^2$, let $M(\mathbb{C}, \epsilon, \delta)$ denote the* sample complexity *of $\mathbb{C}$ (in the classic passive learning sense), or the smallest m such that there is an algorithm $\mathcal{A}$ taking as input a set of examples $\mathcal{L}$ and labels, and outputting a classifier (without making any queries), such that for any $\mathcal{D}$ and $f \in \mathbb{C}$,*

$$\mathcal{P}r_{\mathcal{L} \sim \mathcal{D}^m} \{error_{\mathcal{D}}(\mathcal{A}(\mathcal{L}, f(\mathcal{L})), f) > \epsilon\} \leq \delta.$$

It is known (e.g., [Anthony and Bartlett, 1999]) that

$$\max\{\tfrac{d-1}{32\epsilon}, \tfrac{1}{2\epsilon} \ln \tfrac{1}{\delta}\} \leq M(\mathbb{C}, \epsilon, \delta) \leq \tfrac{4d}{\epsilon} \ln \tfrac{12}{\epsilon} + \tfrac{4}{\epsilon} \ln \tfrac{2}{\delta}$$

for $0 < \epsilon < 1/8$, $0 < \delta < .01$, and $d \geq 2$, where d is the VC-dimension of $\mathbb{C}$. Furthermore, Warmuth has conjectured [Warmuth, 2004] that $M(\mathbb{C}, \epsilon, \delta) = \Theta(\tfrac{1}{\epsilon}(d + \log \tfrac{1}{\delta}))$.

With these definitions in mind, we have the following novel theorem.

Theorem 5.14. *For any instance space $\mathcal{X}$, concept space $\mathbb{C}$ on $\mathcal{X}$ with VC-dimension $d \in (0, \infty)$, sample-based cost function c, $\epsilon \in (0, 1)$, and $\delta \in (0, \tfrac{1}{2})$, if $m = M(\mathbb{C}, \epsilon, \delta)$, then*

$$CostComplexity(\mathbb{C}, c, \epsilon, \delta) \leq GIC(\mathbb{C}, c, m) d \log_2 \tfrac{em}{d}$$

Proof. For the unlabeled sequence, sample $\mathcal{U} \sim \mathcal{D}^m$. If $GIC(\mathbb{C}, c, m) = \infty$, then the upper bound holds vacuously, so let us assume this is finite. Also, $d \in (0, \infty)$ implies $|\mathcal{U}| \in (0, \infty)$ [Anthony and Bartlett, 1999]. By definition of $M(\mathbb{C}, \epsilon, \delta)$, there exists a (passive learning) algorithm $\mathcal{A}$ such that $\forall f \in \mathbb{C}, \forall \mathcal{D}, \mathcal{P}r_{\mathcal{U} \sim \mathcal{D}^m}\{error_{\mathcal{D}}(\mathcal{A}(\mathcal{U}, f(\mathcal{U})), f) > \epsilon\} \leq \delta$. Therefore any algorithm that, by a finite sequence of effective queries with finite cost under $c_{\mathcal{U}}$, identifies $f(\mathcal{U})$ and then outputs $\mathcal{A}(\mathcal{U}, f(\mathcal{U}))$, is an (ϵ, δ)-learning algorithm for $\mathbb{C}$ using c.

Suppose now that there is a *ghost teacher*, who knows the teacher's target concept $f \in \mathbb{C}$. The ghost teacher uses the $h \in \mathbb{C}[\mathcal{U}]$ with $h(\mathcal{U}) = f(\mathcal{U})$ as its target concept. In order to answer any

actual queries $\tilde{q} \in \tilde{Q}$ with $c_{\mathcal{U}}(\tilde{q}) < \infty$, the ghost teacher simply passes the query to the real teacher and then answers the query using the real teacher's answer. This answer is guaranteed to be valid because $c_{\mathcal{U}}$ is a sample-based cost function. Thus, identifying $f(\mathcal{U})$ can be accomplished by identifying $h(\mathcal{U})$, which can be accomplished by identifying h. The task of identifying h can be reduced to an *Exact Learning* task with concept space $\mathbb{C}[\mathcal{U}]$ and cost function $c_{\mathcal{U}}$, where the teacher for the Exact Learning task is the ghost teacher. Therefore, by Theorem 5.6, the total cost required to identify $f(\mathcal{U})$ with a finite sequence of queries is no greater than

$$CostComplexity(\mathbb{C}[\mathcal{U}], c_{\mathcal{U}}) \le GIC(\mathbb{C}[\mathcal{U}], c_{\mathcal{U}}) \log_2 |\mathbb{C}[\mathcal{U}]| \le GIC(\mathbb{C}[\mathcal{U}], c_{\mathcal{U}}) d \log_2 \frac{|\mathcal{U}|e}{d}, \quad (5.1)$$

where the last inequality is due to Sauer's Lemma (e.g., [Anthony and Bartlett, 1999]). Finally, taking the worst case (supremum) over all $\mathcal{U} \in \mathcal{X}^m$ completes the proof. $\qquad\square$

Note that (5.1) also implies a data-dependent bound, which could potentially be useful for practical applications in which the unlabeled examples are available when bounding the cost. It can also be used to state a distribution-dependent bound.

5.3.3 An Example: Intersection-Closed Concept Spaces

As an example application, we can use the above theorem to prove new results for any intersection-closed concept space[8] as follows.

---

[8]An intersection-closed concept space $\mathbb{C}$ has the property that for any $h_1, h_2 \in \mathbb{C}$, there is a concept $h_3 \in \mathbb{C}$ such that $\forall x \in \mathcal{X}, [h_1(x) = h_2(x) = 1 \Leftrightarrow h_3(x) = 1]$. For example, conjunctions and axis-aligned rectangles are intersection-closed.

> **Lemma 5.15.** *For any instance space $\mathcal{X}$, intersection-closed concept space $\mathbb{C}$ with VC-dimension $d \geq 1$, sample-based cost function c such that membership queries in $\mathcal{U}$ have cost $\leq \mu$ (i.e., $\forall \mathcal{U} \subseteq \mathcal{X}, x \in \mathcal{U}, c_{\mathcal{U}}(q_{\{x\}}) \leq \mu$) and positive example queries in $\mathcal{U}$ have cost $\leq \kappa$ (i.e., $\forall \mathcal{U} \subseteq \mathcal{X}, S \subseteq \mathcal{U}, c_{\mathcal{U}}(q_S) \leq \kappa$), and integer $m \geq 0$,*
>
> $$GIC(\mathbb{C}, c, m) \leq \kappa + \mu d$$

Proof. Say we have some set of unlabeled examples $\mathcal{U}$, and consider bounding the value of $GIC(\mathbb{C}[\mathcal{U}], c_{\mathcal{U}})$. In the spy game, suppose the teacher is answering with effective oracle $T \in \mathcal{T}$. Let $\mathcal{U}_+ = \{x | x \in \mathcal{U}, T(q_{\{x\}}) = \{h | h \in \mathbb{C}^*, h(x) = 1\}\}$. The spy first tells the learner to make the $q_{\mathcal{U} \setminus \mathcal{U}_+}$ query (if $\mathcal{U} \setminus \mathcal{U}_+ \neq \varnothing$). If $\exists x \in \mathcal{U} \setminus \mathcal{U}_+$ s.t. $T(q_{\mathcal{U} \setminus \mathcal{U}_+}) = \{h | h \in \mathbb{C}^*, h(x) = 1\}$, then the spy tells the learner to make effective query $q_{\{x\}}$ for this x, and there are no concepts in $\mathbb{C}[\mathcal{U}]$ consistent with the answers to these two queries; the total effective cost for this case is $\kappa + \mu$. If this is not the case, but $|\mathcal{U}_+| = 0$, then there is at most one concept in $\mathbb{C}[\mathcal{U}]$ consistent with the answer to $q_{\mathcal{U} \setminus \mathcal{U}_+}$: namely, the $h \in \mathbb{C}[\mathcal{U}]$ with $h(x) = 0$ for all $x \in \mathcal{U}$, if there is such an h. In this case, the cost is just κ.

Otherwise, let $\bar{S}$ be a largest subset of $\mathcal{U}_+$ such that $\exists h \in \mathbb{C}$ with $\forall x \in \bar{S}, h(x) = 1$. If $\bar{S} = \varnothing$, then making any membership query in $\mathcal{U}_+$ leaves all concepts in $\mathbb{C}[\mathcal{U}]$ inconsistent (at cost μ), so let us assume $\bar{S} \neq \varnothing$. For any $S \subseteq \mathcal{X}$, define

$$CLOS(S) = \{x | x \in \mathcal{X}, \forall h \in \mathbb{C}, [\forall y \in S, h(y) = 1] \Rightarrow h(x) = 1\}$$

the *closure* of S. Let $\bar{S}'$ be a smallest subset of $\bar{S}$ such that $CLOS(\bar{S}') = CLOS(\bar{S})$, known as a *minimal spanning set* of $\bar{S}$ [Helmbold et al., 1990]. The spy now tells the learner to make queries $q_{\{x\}}$ for all $x \in \bar{S}'$.

Any concept in $\mathbb{C}$ consistent with the answer to $q_{\mathcal{U} \setminus \mathcal{U}_+}$ must label every $x \in \mathcal{U} \setminus \mathcal{U}_+$ as 0. Any concept in $\mathbb{C}$ consistent with the answers to the membership queries on $\bar{S}'$ must label every $x \in CLOS(\bar{S}') = CLOS(\bar{S}) \supseteq \bar{S}$ as 1. Additionally, every concept in $\mathbb{C}$ that labels every $x \in \bar{S}$ as 1 must label every $x \in \mathcal{U}_+ \setminus \bar{S}$ as 0, since $\bar{S}$ is defined to be maximal. This labeling of

these three sets completely defines a labeling of $\mathcal{U}$, and as such there is at most one $h \in \mathbb{C}[\mathcal{U}]$ consistent with the answers to all queries made by the learner. Helmbold, Sloan, and Warmuth [Helmbold et al., 1990] proved that, for an intersection-closed concept space with VC-dimension d, for any set $\bar{S}$, all minimal spanning sets of $\bar{S}$ have size at most d. This implies the learner makes at most d membership queries in $\mathcal{U}$, and thus has a total cost of at most $\kappa + \mu d$. $\qquad\square$

Corollary 5.16. *Under the conditions of Lemma 5.15, if $d \geq 10$, then for $0 < \epsilon < 1$, and $0 < \delta < \frac{1}{2}$,*

$$CostComplexity(\mathbb{C}, c, \epsilon, \delta) \leq (\kappa + \mu d) d \log_2 \left(\frac{e}{d} \max \left\{ \frac{16d}{\epsilon} \ln d, \frac{6}{\epsilon} \ln \frac{28}{\delta} \right\} \right)$$

Proof. This follows from Theorem 5.14, Lemma 5.15, and Auer & Ortner's result [Auer and Ortner, 2004] that for intersection-closed concept spaces with $d \geq 10$, $M(\mathbb{C}, \epsilon, \delta) \leq \max \left\{ \frac{16d}{\epsilon} \ln d, \frac{6}{\epsilon} \ln \frac{28}{\delta} \right\}$. $\qquad\square$

For example, consider the concept space of axis-parallel hyper-rectangles in $\mathcal{X} = \mathbb{R}^n$, $\mathbb{C} = \{h : \mathcal{X} \to \{0, 1\} | \exists ((a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)) : \forall x \in \mathbb{R}^n, h(x) = 1 \Leftrightarrow \forall i \in \{1, 2, \ldots, n\}, a_i \leq x_i \leq b_i\}$. One can show that this is an intersection-closed concept space with VC-dimension $2n$. For a sample-based cost function c of the form stated in Lemma 5.15, we have that $CostComplexity(\mathbb{C}, c, \epsilon, \delta) \leq \tilde{O}((\kappa + n\mu)n)$. Unlike the example in the previous section, if all other query types have infinite cost, then for $n \geq 2$ there are distributions that force any algorithm achieving this bound for small ϵ and δ to use multiple positive example queries q_S with $|S| > 1$. In particular, for finite constant κ, this is an exponential improvement over the cost complexity of PAC active learning with only uniform cost membership queries on $\mathcal{U}$.

5.3.4 A Cost Complexity Lower Bound

At first glance, it might seem that $GIC(\mathbb{C}, c, \lceil \frac{1-\epsilon}{\epsilon} \rceil)$ could be a lower bound on

CostComplexity$(\mathbb{C}, c, \epsilon, \delta)$. In fact, one can show this is true for $\delta < (\frac{\epsilon d}{e})^d$. However, there are

simple examples for which this is not a lower bound for general ϵ and δ.[9] We therefore require a

slight modification of GIC to introduce dependence on δ.

Definition 5.17. *For an instance space $\mathcal{X}$, finite concept space $\mathbb{C}$ on $\mathcal{X}$, cost function c, and*

$\delta \in [0, 1)$, *define the* General Partial Identification Cost, *denoted $GPIC(\mathbb{C}, c, \delta)$ as follows.*

$GPIC(\mathbb{C}, c, \delta) = \inf\{t | t \geq 0, \forall T \in \mathcal{T}, \exists R \subseteq \mathcal{Q}, \text{ s.t. } [\sum_{q \in R} c(q) \leq t] \wedge [|\mathbb{C} \cap T(R)| \leq \delta|\mathbb{C}|+1]\}$

Definition 5.18. *For an instance space $\mathcal{X}$, concept space $\mathbb{C}$ on $\mathcal{X}$, sample-based cost function*

c, non-negative integer m, and $\delta \in [0, 1)$, define the General Partial Identification Cost Growth

Function, *denoted $GPIC(\mathbb{C}, c, m, \delta)$, as follows.*

$$GPIC(\mathbb{C}, c, m, \delta) = \sup_{\mathcal{U} \in \mathcal{X}^m} GPIC(\mathbb{C}[\mathcal{U}], c_\mathcal{U}, \delta)$$

It is easy to see that $GIC(\mathbb{C}, c) = GPIC(\mathbb{C}, c, 0)$ and $GIC(\mathbb{C}, c, m) = GPIC(\mathbb{C}, c, m, 0)$, so

that all of the above results could be stated in terms of $GPIC$.

Theorem 5.19. *For any instance space $\mathcal{X}$, concept space $\mathbb{C}$ on $\mathcal{X}$, sample-based cost function*

c, $(\epsilon, \delta) \in (0, 1)^2$, and any $V \subseteq \mathbb{C}$,

$$GPIC(V, c, \lceil \tfrac{1-\epsilon}{\epsilon} \rceil, \delta) \leq \textit{CostComplexity}(\mathbb{C}, c, \epsilon, \delta)$$

Proof. Let $S \subseteq \mathcal{X}$ be a set with $1 \leq |S| \leq \lceil \frac{1-\epsilon}{\epsilon} \rceil$, and let $\mathcal{D}_S$ be the uniform distribution on S.

Thus, $error_{\mathcal{D}_S}(h, f) \leq \epsilon \Leftrightarrow h(S) = f(S)$. I will show that any algorithm $\mathcal{A}$ guaranteeing

$Pr_{\mathcal{U} \sim \mathcal{D}_S^m}\{error_{\mathcal{D}_S}(\mathcal{A}(\mathcal{U}), f) > \epsilon\} \leq \delta$ cannot also guarantee cost strictly less than

$GPIC(V[S], c_S, \delta)$. If $\delta|V[S]| \geq |V[S]| - 1$, the result is clear since no algorithm guarantees

cost less than 0, so assume $\delta|V[S]| < |V[S]| - 1$. Suppose $\mathcal{A}$ is an algorithm that guarantees,

---

[9]The infamous "Monty Hall" problem is an interesting example of this. For another example, consider $\mathcal{X} = \{1, 2, \ldots, N\}$, $\mathbb{C} = \{h_x | x \in \mathcal{X}, \forall y \in \mathcal{X}, h_x(y) = I[x = y]\}$, and cost that is 1 for membership queries in $\mathcal{U}$ and infinite for other queries. Although $GIC(\mathbb{C}, c, N) = N - 1$, it is possible to achieve better than $\epsilon = \frac{1}{N+1}$ with probability close to $\frac{N-2}{N-1}$ using cost no greater than $N - 2$.

for every finite sequence $\mathcal{U}$ of elements from S, $\mathcal{A}(\mathcal{U})$ incurs total cost strictly less than $GPIC(V[S], c_S, \delta)$ under $c_{\mathcal{U}}$ (and therefore also under c_S). By definition of $GPIC$, $\exists \hat{T} \in \mathcal{T}$ such that for any set of queries R that $\mathcal{A}(\mathcal{U})$ makes, $|V[S] \cap \hat{T}(R)| > \delta|V[S]| + 1$. I now proceed by the probabilistic method. Say the teacher draws the target concept f uniformly at random from $V[S]$, and $\forall q \in \mathcal{Q}$ s.t. $f \in \hat{T}(q)$, answers with $\hat{T}(q)$. Any $q \in \mathcal{Q}$ such that $f \notin \hat{T}(q)$ can be answered with an arbitrary $a \in q(f)$. Let $h_{\mathcal{U}} = \mathcal{A}(\mathcal{U})$; let $R_{\mathcal{U}}$ denote the set of queries $\mathcal{A}(\mathcal{U})$ would make if *all* queries were answered with $\hat{T}$.

$\mathbb{E}_f[\mathcal{P}r_{\mathcal{U} \sim \mathcal{D}_S^m}\{error_{\mathcal{D}_S}(\mathcal{A}(\mathcal{U}), f) > \epsilon\}]$

$$= \mathbb{E}_{\mathcal{U} \sim \mathcal{D}_S^m}[\mathcal{P}r_f\{h_{\mathcal{U}}(S) \neq f(S)\}]$$

$$\geq \mathbb{E}_{\mathcal{U} \sim \mathcal{D}_S^m}[\mathcal{P}r_f\{h_{\mathcal{U}}(S) \neq f(S) \wedge f \in \hat{T}(R_{\mathcal{U}})\}]$$

$$\geq \min_{\mathcal{U} \in S^m} \frac{|V[S] \cap \hat{T}(R_{\mathcal{U}})| - 1}{|V[S]|} > \delta.$$

Therefore, there exists a deterministic method for selecting f and answering queries such that $\mathcal{P}r_{\mathcal{U} \sim \mathcal{D}_S^m}\{error_{\mathcal{D}_S}(\mathcal{A}(\mathcal{U}), f) > \epsilon\} > \delta$. In particular, this proves that there are no (ϵ, δ)-learning algorithms that guarantee cost strictly less than $GPIC(V[S], c_S, \delta)$. Taking the supremum over sets S completes the proof. $\qquad\square$

> **Corollary 5.20.** *Under the conditions of Theorem 5.19,*
> $$GPIC(\mathbb{C}, c, \lceil \tfrac{1-\epsilon}{\epsilon} \rceil, \delta) \leq \textit{CostComplexity}(\mathbb{C}, c, \epsilon, \delta).$$

Equipped with Theorem 5.19, it is straightforward to prove the claim made in Section 5.3.3 that there are distributions forcing any (ϵ, δ)-learning algorithm for Axis-parallel rectangles using only membership queries (at cost μ) to pay $\Omega(\frac{\mu(1-\delta)}{\epsilon})$. The details are left as an exercise.

5.4 Discussion and Open Problems

Note that the usual "query counting" analysis done for Active Learning is a special case of cost complexity (uniform cost 1 on the allowed queries, infinite cost on the others). In particular, Theorem 5.14 can easily be specialized to give a worst-case bound on the query complexity for

the widely studied setting in which the learner can make any *membership queries* on examples in $\mathcal{U}$ [Dasgupta, 2005]. However, for this special case, one can derive a slightly tighter bound. Following the proof technique of Hegedüs [Hegedüs, 1995], one can show that for any sample-based cost function c such that $\forall \mathcal{U} \subseteq \mathcal{X}, q \in \mathcal{Q}$,

$c_{\mathcal{U}}(q) < \infty \Rightarrow [c_{\mathcal{U}}(q) = 1 \wedge \forall f \in \mathbb{C}^*, |q(f)| = 1]$, *CostComplexity*$(\mathbb{C}, c_{\mathcal{X}}) \leq 2\frac{GIC(\mathbb{C},c_{\mathcal{X}}) \log_2 |\mathbb{C}|}{\log_2 GIC(\mathbb{C},c_{\mathcal{X}})}$.

This implies for the PAC setting that *CostComplexity*$(\mathbb{C}, c, \epsilon, \delta) \leq 2\frac{GIC(\mathbb{C},c,m)d \log_2 m}{\log_2 GIC(\mathbb{C},c,m)}$, for VC-dimension $d \geq 3$ and $m = M(\mathbb{C}, \epsilon, \delta)$. This includes the cost function assigning 1 to membership queries on $\mathcal{U}$ and ∞ to all others.

Active Learning in the PAC model is closely related to the topic of *Semi-Supervised Learning*. Balcan & Blum [Balcan and Blum, 2005] have recently derived a variety of sample complexity bounds for Semi-Supervised Learning. Many of the techniques can be transfered to the pool-based Active Learning setting in a fairly natural way. Specifically, suppose there is a quantitative notion of "compatibility" between a concept and a distribution, which can be estimated from a finite unlabeled sample. If we know the target concept is highly compatible with the data distribution, we can draw enough unlabeled examples to estimate compatibility, then identify and discard those concepts that are probably highly incompatible. The set of highly compatible concepts may be significantly less expressive, therefore reducing *both* the number of examples for which an algorithm must learn the labels to guarantee generalization *and* the number of labelings of those examples the algorithm must distinguish between, thereby also reducing the cost complexity.

There are a variety of interesting extensions of this framework worth pursuing. Perhaps the most natural direction is to move into the agnostic PAC framework, which has thus far been quite elusive for active learning except for a few results [Balcan et al., 2006, Kääriäinen, 2005]. Another possibility is to derive cost complexity bounds when the cost c is a function of not only the query, but also the target concept. Then every time the learning algorithm makes a query q, it is charged $c(q, f)$, but does not necessarily know what this value is. However, it can always

upper bound the total cost so far by the worst case over concepts in the version space. Can anything interesting be said about this setting (or variants), perhaps under some benign smoothness constraints on $c(q, \cdot)$? This is of some practical importance since, for example, it is often more difficult to label examples that occur near a decision boundary.

Bibliography

K. Alexander. Probability inequalities for empirical processes and a law of the iterated logarithm. *Annals of Probability*, 4:1041–1067, 1984. 4.4.1

M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999. 5.3.2, 5.3.2, 5.3.2

A. Antos and G. Lugosi. Strong minimax lower bounds for learning. *Machine Learning*, 30: 31–56, 1998. 3.2.2, 3.3

P. Auer and R. Ortner. A new PAC bound for intersection-closed concept classes. In *17th Annual Conference on Learning Theory (COLT)*, 2004. 2.9.2, 5.3.3

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. 3.6

M.-F. Balcan and A. Blum. A PAC-style model for learning from labeled and unlabeled data. In *Conference on Learning Theory*, 2005. 5.4

M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *Proc. of the 23rd International Conference on Machine Learning*, 2006. 2.1, 2.1.2, 2.2.1, 2.3, 3.1, 3.5.2, 5.4

M.-F. Balcan, A. Broder, and T. Zhang. Margin based active learning. In *Proc. of the 20th Conference on Learning Theory*, 2007. 3.1, 3.5.2

M.-F. Balcan, S. Hanneke, and J. Wortman. The true sample complexity of active learning. In

Proceedings of the 21st Conference on Learning Theory, 2008. 2.1.2, 3.1

J. L. Balcázar, J. Castro, D. Guijarro, and H.-U. Simon. The consistency dimension and distribution-dependent learning from queries. In *Algorithmic Learning Theory*, 1999. 5.2.1, 5.2.2

J. L. Balcázar, J. Castro, and D. Guijarro. A general dimension for exact learning. In *14th Annual Conference on Learning Theory*, 2001. 5.1, 5.2.1, 5.2.2, 5.2.2

J. L. Balcázar, J. Castro, and D. Guijarro. A new abstract combinatorial dimension for exact learning via queries. *Journal of Computer and System Sciences*, 64:2–21, 2002. 5.2.1

P. L. Bartlett, O. Bousquet, and S. Mendelson. Local rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005. 2.8

G. Benedek and A. Itai. Learnability by fixed distributions. In *Proc. of the First Workshop on Computational Learning Theory*, pages 80–90, 1988. 1.1

A. Beygelzimer, S. Dasgupta, and J. Langford. Importance weighted active learning, 2009. 1.7, 2.1.2, 2.1.2

A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the Association for Computing Machinery*, 36(4): 929–965, 1989. 1.1, 2.3.1, 3.11, 3.11

R. Castro and R. Nowak. Upper and lower error bounds for active learning. In *The 44th Annual Allerton Conference on Communication, Control and Computing*, 2006. 2.1, 2.1.1, 2.3.3, 2.3.4

R. Castro and R. Nowak. Minimax bounds for active learning. In *Proceedings of the 20th Conference on Learning Theory*, 2007. 2.3.4, 3.1, 2

D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994. 1.2, 1.4, 2.1.2, 3.1

S. Dasgupta. Analysis of a greedy active learning strategy. In *Advances in Neural Information*

Processing Systems 17, 2004. 3.1, 3.2.2, 5.3.1

S. Dasgupta. Coarse sample complexity bounds for active learning. In *Advances in Neural Information Processing Systems 18*, 2005. 1.6, 1.6, 3.1, 3.2, 3.2.1, 3.2.2, 3.4, 3.5.2, 3.6, 3.10, 5.3.1, 5.4

S. Dasgupta, A. Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. In *Proc. of the 18th Conference on Learning Theory*, 2005. 1.6, 3.1, 3.5.2

S. Dasgupta, D. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. Technical Report CS2007-0898, Department of Computer Science and Engineering, University of California, San Diego, 2007. 2.1, 2.1.2, 2.1.2, 2.2.2, 2.3, 2.3.2, 2.9, 3.1

L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag New York, Inc., 1996. 2.2.2, 2.4, 3.5.2, 3.6, 3.11, 3.11, 4.4.1, 4.4.1

Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28:133–168, 1997. 3.2.2, 5.3.1

S. Hanneke. Teaching dimension and the complexity of active learning. In *Proceedings of the 20th Conference on Learning Theory*, 2007a. 3.1, 3.2.2

S. Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th International Conference on Machine Learning*, 2007b. 2.1, 2.1.2, 2.1.2, 2.2, 2.3, 2.1.2, 2.3.1, 2.3.2, 2.8, 2.9.1, 3.1, 3.2.1, 3.5.2, 3.5.2

D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100:78–150, 1992. 1.5

D. Haussler, N. Littlestone, and M. Warmuth. Predicting $\{0, 1\}$-functions on randomly drawn points. *Information and Computation*, 115:248–292, 1994. 2.9, 3.3, 3.3, 4.2

T. Hegedüs. Generalized teaching dimension and the query complexity of learning. In *Proc. of the 8th Annual Conference on Computational Learning Theory*, 1995. 5.2.1, 5.2.2, 5.4

L. Hellerstein, K. Pillaipakkamnatt, V. Raghavan, and D. Wilkins. How many queries are

needed to learn? *Journal of the Association for Computing Machinery*, 43(5):840–862, 1996. 5.2.1, 5.2.2

D. Helmbold, R. Sloan, and M. Warmuth. Learning nested differences of intersection-closed concept classes. *Machine Learning*, 5:165–196, 1990. 5.3.3

M. Kääriäinen. On active learning in the non-realizable case. In *NIPS Workshop on Foundations of Active Learning*, 2005. 5.4

M. Kääriäinen. Active learning in the non-realizable case. In *Proc. of the 17th International Conference on Algorithmic Learning Theory*, 2006. 1.7, 2.3.3

A. T. Kalai, A. R. Klivans, Y. Mansour, and R. A. Servedio. Agnostically learning halfspaces. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, 2005. 2.8

V. Koltchinskii. Local rademacher complexities and oracle inequalities in risk minimization. *The Annals of Statistics*, 34(6):2593–2656, 2006. 1.1, 1.3, 2.1.1, 2.3.4, 2.3.4, 2.4, 2.6, 2.16, 2.7, 2.7.2

S. R. Kulkarni. On metric entropy, vapnik-chervonenkis dimension, and learnability for a class of distributions. Technical Report CICS-P-160, Center for Intelligent Control Systems, 1989. 1.1

S. R. Kulkarni, S. K. Mitter, and J. N. Tsitsiklis. Active learning using arbitrary binary valued queries. *Machine Learning*, 11:23–35, 1993. 1.5

Y. Li and P. M. Long. Learnability and the doubling dimension. In *Advances in Neural Information Processing*, 2007. 2.1.2, 2.9.1

P. M. Long. On the sample complexity of PAC learning halfspaces against the uniform distribution. *IEEE Transactions on Neural Networks*, 6(6):1556–1559, 1995. 1.1

E. Mammen and A. Tsybakov. Smooth discrimination analysis. *Annals of Statistics*, 27: 1808–1829, 1999. 1.3, 2.1, 2.1.1, 2.3.4

P. Massart and Élodie Nédélec. Risk bounds for statistical learning. *The Annals of Statistics*, 34 (5):2326–2366, 2006. 2.1.1, 2.7.2

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. 3.6

A. B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004. 1.3, 2.1, 2.1.1, 2.3.3, 2.3.4, 2.3.4, 2.4

L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, Nov. 1984. 1.3

A. W. van der Vaart and J. A. Wellner. *Weak Convergence and Empirical Processes*. Springer, 1996. 1.3, 1.3, 2.3.4

V. Vapnik. *Estimation of Dependencies Based on Empirical Data*. Springer-Verlag, New York, 1982. 2.1.2, 2.2.1, 2.3.1, 2.4, 2.9, 2.9, 3.11, 3.11, 4.4.1, 4.4.1

V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc., 1998. 1.1, 1.3, 2.2.1, 3.6, 4.2

M. Warmuth. The optimal pac algorithm. In *Conference on Learning Theory*, 2004. 5.3.2