UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

Topics in the Theory of Learning

Permalink

https://escholarship.org/uc/item/8xx63870

Author

Shafer, Jonathan

Publication Date

2023

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

Topics in the Theory of Learning

By

Jonathan Shafer

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Shafi Goldwasser, Chair
Professor Peter L. Bartlett
Assistant Professor Shay Moran
Assistant Professor Avishay Tal

Fall 2023

Topics in the Theory of Learning

Copyright 2023
by
Jonathan Shafer

Abstract

Topics in the Theory of Learning

By

Jonathan Shafer

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Shafi Goldwasser, Chair

AI applications have seen great advances in recent years, so much so that the people who design and build them often have difficulty understanding, predicting and controlling what they do. To make progress on these fundamental challenges, I believe that developing a solid mathematical foundation for AI is both beneficial and possible. The research presented in this dissertation is an attempt to chip away at a few small aspects of that endeavor.

The dissertation is divided into three parts.

Part I addresses a question at the core of learning theory: "how much data is necessary for supervised learning?" Concretely, Chapter 2 considers statistical settings, in which training data is drawn from an unknown distribution. Here, we answer a question posed by Antos and Lugosi (1996) concerning the shape of learning curves. To do so, we define a new combinatorial quantity, which we call the *Vapnik–Chervonenkis–Littlestone dimension*, and show that it characterizes the rate at which the learner's error decays. Our result has a number of additional benefits: it refines the trichotomy theorem of Bousquet, Hanneke, Moran, van Handel, and Yehudayoff (2021); qualitatively strengthens classic 'no free lunch' lower bounds; and establishes that, in the distribution-dependent setting, semi-supervised learning is no easier than supervised learning. Chapter 3 considers adversarial settings, in which few or no assumptions are made regarding the source of the training data. Specifically, we chart the landscape of transductive online learning, showing how it compares to the standard setting in online learning, and how it relates to combinatorial quantities such as the VC and Littlestone dimensions.

Part II investigates whether it is possible to verify the optimality of a machine learning outcome offered by an untrusted party, such that verification would be significantly cheaper (in terms of compute, or the quantity or quality of training data) compared to the cost of running a trusted machine learning system. This question has many parallels in the theory of computation, and it also has tangible implications to the economics of selling machine

learning as a service. Our first contribution is to introduce a notion of *interactive proofs for verifying machine learning*. Here, the entity running the learning algorithm proves to the verifier that a proposed hypothesis is competitive with some benchmark, for instance, is sufficiently close to the best hypothesis in a reference class of hypotheses. Our primary focus is verifying agnostic supervised machine learning. Within this framework, we show a host of verification protocols and lower bounds, establishing that in some cases, verification can be significantly cheaper than learning, while in other cases it cannot. In particular, our results include: (1) for supervised learning, the sample complexity gap between learning and verifying is quadratic in some (natural) cases, and furthermore it can never be greater than quadratic; (2) whereas learning the class of Fourier-sparse boolean functions using i.i.d. samples is LPN-hard, we show that there exists an efficient protocol for verifying this class, wherein the verifier only uses i.i.d. samples.

Part III studies notions of *stability* in machine learning. We offer a taxonomy that can help make sense of an assortment of seemingly-unrelated stability definitions that have appeared in the learning theory literature. Our starting point is an observation that many of these definitions actually follow a similar abstract formulation. We call this the *Bayesian* formulation of stability, and we ask, to what extent are the various Bayesian definitions in the literature actually different from one another. To answer this question, we distinguish between two variants: distribution-*dependent* Bayesian stability, and distribution-*independent* Bayesian stability. Putting together results from a number of recent publications shows that many distribution-dependent Bayesian definitions, including approximate differential privacy, are in fact weakly equivalent to each other. To complete the picture, we investigate the family of distribution-independent Bayesian definitions. We show that here too, many definitions, including pure differential privacy, are weakly equivalent to each other. Our proof involves developing a boosting algorithm that simultaneously improves the accuracy and the stability of a learning algorithm.

The dissertation consists of five chapters, each of which is self-contained and can be read independently. However, a small number of basic notions crop up repeatedly in different guises. Taken together, the dissertation showcases the richness, versatility and unity of learning theory.

To the caring and kind people in my life.

Contents

List of Algorithms

List of Protocols

List of Figures

List of Tables

Acknowledgments

I am deeply grateful to many people who have shown me true kindness and wisdom during my Ph.D. years.

Having Shafi Goldwasser as my Ph.D. advisor is a story of breathtaking and benevolent luck; it has been an astounding privilege. Her scientific creativity remains unmatched, except by her good will. I have also benefitted tremendously from the mentorship of Shay Moran, Amir Yehudayoff, Guy Rothblum and Ido Nachum. Learning from Shafi and them has been at the heart of my Ph.D.

I have been further blessed with ingenious co-authors and collaborators. Their minds shine brightly throughout the pages of this dissertation, yet their amity has shown brighter still. Beyond the above, they include Olivier Bousquet, Nataly Brukhim, Michael Gastpar, Jesse Goodman, Steve Hanneke, Michael Kim, Robert Kleinberg, Saachi Mutreja, Ido Nachum, Hilla Schefler, Abhishek Shetty, Ilya Tolstikhin, Neekon Vafa, Vinod Vaikuntanathan, Thomas Weinberger, and others.

I am indebted to Christos Papadimitriou and Alessandro Chiesa for their warm welcome upon my arrival at Berkeley; to Peter Bartlett, Shay Moran, and Avishay Tal for serving on my Ph.D. committee; and to Robert Kleinberg, Guy Rothblum, and Shay Moran, for hosting me at Cornell, Weizmann and the Technion, respectively.

A special thank-you to Jean Nguyen, who has been an unfailingly trustworthy and sympathetic guide to all administrative issues at Berkeley; to Josephine Williamson who looked out for a graduate student instructor who was being under-paid; and to the other friendly staff members at Berkeley.

Finally, to my close friends, my family, and to my partner — you mean the world to me.

Sources

This dissertation is a compilation of the following joint works:

- Chapter 2: Bousquet, Hanneke, Moran, Shafer, and Tolstikhin (2023);
- Chapter 3: Hanneke, Moran, and Shafer (2023);
- Chapter 4: Goldwasser, Rothblum, Shafer, and Yehudayoff (2021);
- Chapter 5: Mutreja and Shafer (2023);
- Chapter 6: Moran, Schefler, and Shafer (2023).

Preface

Cryptography used to be a very risky business. Mary, Queen of Scots, was beheaded after a cipher she trusted was broken, revealing her involvement in a conspiracy against the English monarch. Generally, the pattern has been an endless game of cat-and-mouse between the makers and breakers of codes: new ciphers were devised, deemed secure (or even "unbreakable"), and subsequently cracked — often with disastrous consequences. The fate of any particular cipher was highly uncertain.[1]

But modern cryptography offers a greater degree of certainty. The advent of computer science in the twentieth century provided a precise mathematical language for analyzing the security of ciphers. Rather than relying solely on intuition and practical experience, the security of modern ciphers rests to a large extent on solid *mathematical proof*. Just as the proofs in Euclid's *Elements* are no less valid today than they were in Alexandria circa 300 B.C.E., the proofs underlying modern cryptography will remain valid into the future. Every day, ordinary people transfer billions of web pages, messages and dollars electronically — secured by cryptography. The ubiquity of cryptography in modern life bears witness to the success of this new, proof-based paradigm.

Around the same time that cryptography was finally starting to find a solid foundation in mathematics, a new mercurial field was born: artificial intelligence. It started off with great optimism: "It is expected by IBM [...] that within a few years there will be a number of 'brains' translating all languages with equal aplomb and dispatch" (news report, 1954[2]); "Machines will be capable, within twenty years, of doing any work a man can do" (future Nobel and Turing laureate Herbert Simon, writing in 1960[3]). But by the mid 1970s, the mood was changing: "Most workers in AI [...] confess to a pronounced feeling of disappointment. [...] In no part of the field have the discoveries made so far produced the major impact that was then promised" (Lighthill, 1973, p. 8). This led to the first 'AI winter', where funding dried up and research slowed. Since then, the field has gone through a series of boom-and-bust

[1] Some examples: the Vigenère cipher was touted in the early modern period as *le chiffre indéchiffrable*, but was occasionally broken by contemporaries (and was decisively broken by Kasiski, 1863); the German diplomatic codes in World War I are infamous for the Zimmermann Telegram, compromised by the British Admiralty; and, of course, the cracking of the Enigma influenced the outcome of World War II. See Dooley (2018); Kahn (1996); Singh (1999) for accounts of this tumultuous history.

[2] See Hutchins (2004), p. 5.

[3] Simon (1960), p. 38; reproduced on p. 96 of Simon (1965), which also espouses a similar position in the introduction, under the name "technological radicalism".

cycles, with expectations oscillating wildly.

AI has come a long way since the early years, but its unpredictable nature remains. In June 2023, two Manhattan lawyers were fined $5,000 for submitting a legal brief replete with “bogus judicial decisions, with bogus quotes and bogus internal citations” (Weiser, 2023a; 2023b). The lawyers used ChatGPT for their legal research, unaware of its uncanny penchant for fabrication. In August 2023, San Francisco “filed motions asking the utilities commission to halt the expansion [of autonomous taxis] altogether,” following “dozens of incidents in which a driverless car interfered with emergency vehicles,” including an autonomous vehicle that “collided with a fire truck in the city, injuring a passenger” (Lu, 2023). Disturbingly, the possibility that AI poses an existential risk to humanity at large is widely discussed (Bostrom, 2014; Russell, 2019). In March 2023, leading academics and practitioners called for a moratorium on developing larger AI models, stating that AI labs are “locked in an out-of-control race to develop and deploy ever more powerful digital minds that no one — not even their creators — can understand, predict, or reliably control” (Future of Life Institute, 2023).

There are many reasons why AI systems can fail, or be hard to predict or control. One major reason is that AI, like early cryptography, lacks a solid mathematical foundation. Progress is made by tinkering, in a process of trial-and-error. Even in hindsight, practitioners cannot fully explain why certain designs work and others fail — let alone derive those outcomes mathematically. There is a fundamental paucity of understanding.

I believe that developing a solid mathematical theory of AI is both beneficial and possible. The research presented in this dissertation is an attempt to chip away at a few small aspects of this problem.

The 2012 Turing Award was presented to Goldwasser and Micali for “pioneer[ing] the field of provable security, which laid the mathematical foundations that made modern cryptography possible” (Association for Computing Machinery, 2013). The transformation from a practice-driven field to a proof-based one has made cryptography safer and more useful. I am hopeful that in the future, scientific understanding of the fundamental information-processing phenomena of learning will undergo a similar transformation.

Chapter 1

A Brief Overview

This chapter offers a synopsis of our main contributions. For a more gentle and comprehensive introduction to each topic, see the introductory section in each of the subsequent chapters.

1.1 Learning

Fine-Grained Distribution-Dependent Learning Curves

The starting point for Chapter 2 is an observation by Antos and Lugosi (1996), who noted that classic VC lower bounds are somewhat unsatisfactory: people often face a specific unknown population distribution, and want to know how many i.i.d. samples they would need to collect from this distribution to guarantee an error of at most ε, for various choices of ε. But VC lower bounds do not fully answer this question, because the 'hard' distribution that witnesses the lower bound might be different for each value of ε.

This lead Antos and Lugosi to pose the question of 'strong minimax lower bounds': for a given VC class, what is the 'worst' rate function $f(n)$ such that for every learning algorithm there exists a realizable distribution for which the expected 0-1 loss after observing n i.i.d. samples is at least $f(n)$ for infinitely many values of n?[1]

We answer this question, showing in Theorem 2.3.1 that for any hypothesis class $\mathcal{H}$, the optimal learning rate satisfies

$$\frac{\Omega(d)}{n} \leq \mathsf{Opt} \leq \frac{O(d)}{n} + C \cdot e^{-c \cdot n},$$

where $d = \mathsf{VCL}(\mathcal{H})$, the Vapnik–Chervonenkis–Littlestone dimension, is a combinatorial quantity we define.[2]

[1] See Question 2.1.1 for a more careful formulation.

[2] The lower bound holds for infinitely many n, the upper bound holds for all n, d depends only on $\mathcal{H}$, while C and c may depend also on the population distribution.

This characterization of the optimal learning rate answers the question of strong minimax lower bounds (up to universal multiplicative constants), qualitatively strengthening classic 'no free lunch' lower bounds. It offers some further benefits as well, including:[3]

- Refining the upper bound of Bousquet et al. (2021). They showed a bound of c/n, where c depends on the hypothesis class and on the distribution. In contrast, our expression offers a *decomposition* of the error rate into a linear component that depends only on the class, and an exponential component that depends also on the distribution.
- Quantitatively strengthening the lower bound of Bousquet et al. (2021). We show a lower bound of $\Omega(d/n)$, distinguishing between different linear rates with different values of d, whereas they show a lower bound of $\Omega(1/n)$ for all linear-rate classes.
- Showing that, in the setting of distribution-dependent learning curves, semi-supervised learning is no easier than supervised learning.

The proofs use tools from the theory of infinite (Gale–Stewart) games, a simple theorem from Ramsey theory, and a careful application of Fatou's lemma.

A Trichotomy for Transductive Online Learning

Chapter 3 studies the transductive online learning setting of Ben-David, Kushilevitz, and Mansour (1997). This setting is similar to the standard online learning setting (Littlestone, 1988), except that the adversary reveals to the learner the full sequence of instances to be labeled at the start of the game. Prediction tasks of this type arise in many real-world situations where an agent has a schedule or a to-do list, known in advance, consisting of specific decisions to be made in order.

We show four main results, elucidating how the optimal number of learner mistakes is governed by well-known combinatorial quantities like the VC and Littlestone dimensions:

1. A *trichotomy* for the realizable case (Theorem 3.4.1): for any hypothesis class, the optimal number of mistakes on a sequence of length n is either n, $\Theta(\log(n))$, or $\Theta(1)$; this is determined by the finiteness of the VC and Littlestone dimensions.
2. For classes with a finite Littlestone dimensions d_L, the optimal number of mistakes in the realizable setting is $\Omega(\log(d_L))$ (Theorem 3.3.1). This improves upon a lower bound of $\Omega\left(\sqrt{\log(d_L)}\right)$ due to Ben-David et al. (1997).
3. A trichotomy for the realizable multiclass setting with a finite number of labels (Theorem 3.5.1), which is analogous to the binary-label trichotomy in Item 1. Here, the Natarajan dimension takes the place of the VC dimension. However, we also show that in the case of an infinite number of labels, the analogous trichotomy (employing the DS dimension) does *not* hold.

[3] See Section 2.1 (Benefits of the New Characterization) for further discussion.

4. For the agnostic setting, we show a regret bound of $\tilde{\Theta}\left(\sqrt{d_{VC} \cdot n}\right)$, where d_{VC} is the VC dimension (Theorem 3.6.1).

The proofs use a connection to the threshold dimension, a novel multi-class generalization of the threshold dimension, a finite variant of the result from Ramsey theory mentioned above, as well as a potential-based argument of a type that, to our knowledge, hasn't been used before in online learning.

1.2 PAC Verification

Chapters 4 and 5 explores the possibilities and limitations of using interactive proofs to verify machine learning. The main idea is that while machine learning systems are often expensive to run, they might in some cases be amenable to cheap verification of their outputs. In these cases, a party with fewer computational resources (e.g., a small business, or an individual citizen) can delegate a machine learning task to a computationally more powerful party (e.g., a large corporation, or a government), and then easily verify that the purported result is valid. This would enable weaker parties to hold stronger parties to account, and would facilitate smooth interactions in conditions of imperfect trust.

We introduce a mathematical formulation of this idea. We focus on verification of supervised *probably approximately correct* (PAC) learning, but also consider verification of other types of statistical computations. In this setting, we show multiple upper and lower bounds for supervised learning, including:

1. The number of i.i.d. samples used by the verifier can never be less than the square root of the number of samples required for learning (Theorem 5.2.1). Furthermore, there exist natural hypothesis classes where such a quadratic gap is attained (Theorem 5.2.2).

2. For some hypothesis classes, there is no significant sample complexity gap between *proper* learning and *proper* verification (Theorem 4.4.1).

3. Whereas learning the class of Fourier-sparse boolean functions using i.i.d. samples is computationally hard under the *learning parities with noise* (LPN) assumption, we show that there exists an efficient protocol for verifying this class in which the verifier only uses i.i.d. samples (Theorem 4.2.6).

The proofs feature the probabilistic method, reductions to and from distribution testing, and an interactive version of the Goldreich–Levin algorithm, among other tools.

1.3 Stability

Stability is a central notion in learning theory, having strong connections to generalization. Furthermore, recent research has uncovered connections between stability and other aspects of learning including privacy, fairness, and replicability. Consequently, quit a number of definitions of stability have been proposed in recent years, with different motivations and contexts. To help make sense of this wealth of definitions, we study equivalences between them. By dividing the definitions into equivalence classes, we can simplify the picture, and create a clean *taxonomy* of definitions. Equivalences also enables cross-pollination, transferring (some) results from one subfield to another.

Our taxonomy considers *Bayesian* notions of stability, under which an algorithm is considered stable if its prior and posterior distributions are close to each other. We distinguish between two such types of definitions, named distribution-*dependent* Bayesian stability, and distribution-*independent* Bayesian stability.

A survey of exiting literature reveals that the family of distribution-dependent Bayesian definitions contains many important definitions, including approximate differential privacy — and these definitions are weakly equivalent to each other!

To complete the picture, we ask: are there also interesting distribution-*independent* definitions that are weakly equivalent to each other? And what equivalences does *pure* differential privacy satisfy?

We answer these questions by showing equivalences between a number of distribution-independent Bayesian definitions of interest, including pure differential privacy. Along the way, we prove a boosting result, stating that there exists a boosting algorithm that simultaneously improves both the accuracy and the stability of a learning algorithm. Our proofs use recent results on the *fractional clique dimension* (Alon, Moran, Schefler, and Yehudayoff, 2023), among other tools.

Part I

Learning

Chapter 2

Fine-Grained Distribution-Dependent Learning Curves

2.1 Introduction

The most fundamental question in learning theory is arguably "*what can be learned, and what quantities of resources (such as data and computation) are necessary for learning when learning is possible?*" The classic and definitive mathematical treatment of this question for supervised learning has traditionally been provided by the PAC framework, due to Vapnik and Chervonenkis (1968, 1971) and Valiant (1984). However, it has become increasingly clear that the PAC model does not accurately capture the reality of learning; VC bounds are overly pessimistic, and modern machine learning algorithms routinely outperform them. This is partially because the PAC model constitutes a worst-case analysis over all distributions. In contrast, machine learning practitioners are typically faced with one (or a few) target distributions, they are interested in optimizing performance only with respect to these specific distributions, and therefore they can vastly outdo the worst-case analysis.

Indeed, Antos and Lugosi (1996, 1998) observed that while the classic PAC bounds decay like $\Omega(d/n)$ for a class of VC dimension d, there exist hypothesis classes with arbitrarily large VC dimension that are learnable such that for every realizable distribution the expected loss decays exponentially fast.

They wrote:

> "[I]n some sense, these [VC] lower bounds are not satisfactory. They do not tell us anything about the way the error decreases as the sample size is increased for a given classification problem. These bounds, for each n, give information about the maximal error within the class, but not about the behavior of the error for a single fixed [distribution] as the sample size n increases. In other words, the 'bad' [distribution], causing the largest error for a learning rule, may be different for each n."[1]

[1] From Antos and Lugosi (1996), edited for clarity.

This lead them to study the following question:

Question 2.1.1 (Strong Minimax Lower Bound)**.** *For a VC class, what is the largest $d' \geq 0$ such that for every learning algorithm there exists a realizable distribution for which the expected 0-1 loss after observing n i.i.d. samples is at least d'/n infinitely often?*

They were able to answer this question for a number of specific hypothesis classes. Furthermore, they showed that "it is neither the VC dimension, nor the rate of increase of the shatter coefficients of the class" that determine the answer. The general case, however, has remained open.

In this chapter we solve Question 2.1.1. We do so in a principled manner, by contributing to the nascent study of distribution-dependent learning curves. We build upon the recent results of Bousquet et al. (2021), who offered a characterization of these curves.

For each instance, consisting of a hypothesis class and a target distribution, the distribution-dependent learning curve is the expected 0-1 loss of a learning algorithm as a function of the number of i.i.d. samples from the distribution (see Section 2.2 for formal definitions).

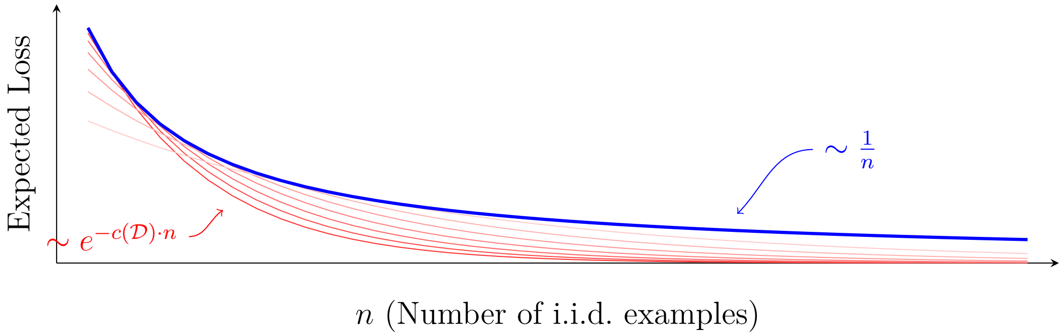


Figure 2.1: Illustration of the difference between distribution-dependent and PAC rates. Each red curve shows exponential decay of the error $\mathsf{Opt} = \mathbb{E}_{S\sim\mathcal{D}^n}[L^{0\text{-}1}_{\mathcal{D}}(\widehat{h}_S)]$ for a different data distribution $\mathcal{D}$; but the PAC rate only captures the pointwise supremum of these curves (the blue curve) which decays linearly at best.

Source: Bousquet et al. (2021), adapted with permission.

The lay of the land when viewed from the perspective of distribution-dependent learning curves is remarkably structured, and remarkably different from that of the PAC model, as captured by the following crisp trichotomy.

Theorem 2.1.2 (Bousquet et al. (2021), Theorem 1.6)**.** *For every concept class $\mathcal{H}$ with $|\mathcal{H}| \geq 3$, exactly one of the following holds:*

- *$\mathcal{H}$ is learnable with optimal rate e^{-n}.*

- *$\mathcal{H}$ is learnable with optimal rate $\frac{1}{n}$.*
- *$\mathcal{H}$ requires arbitrarily slow rates.*

This differs markedly from PAC learning bounds because, for example, it is possible for a class $\mathcal{H}$ to have infinite VC dimension but still be learnable with an exponential rate; and ERM algorithms, which are optimal in the PAC setting, can perform arbitrarily worse than the best learning algorithm in the distribution-dependent setting (see Example 2.3 and Example 2.6 respectively in Bousquet et al., 2021). Bousquet et al. (2021) also provide a combinatorial characterization (via infinite trees) that determines for each hypothesis class $\mathcal{H}$ which of the three prongs of the trichotomy it belongs to.

While the trichotomy of Theorem 2.1.2 is an important characterization, it is far from constituting a complete distribution-dependent theory of supervised learning. To see this, we recall the definition of *learning at rate* $R(n)$ for some function $R : \mathbb{N} \to [0, 1]$, as used in the trichotomy. Roughly (see Definition 2.2.15 below), a class $\mathcal{H}$ is learnable at rate R if there exists a learning algorithm such that for any realizable distribution there exist parameters $C, c \geq 0$ (that depend on the distribution) such that the 0-1 loss of the algorithm after seeing n i.i.d. samples from the distribution is at most $C \cdot R(c \cdot n)$. In other words, each instance, consisting of a hypothesis class and a distribution, determines a pair of parameters $C, c \geq 0$ which together specify the shape of the learning curve.

The characterization of Theorem 2.1.2 explains the general shape of the learning curve (exponential, linear, or arbitrarily slow decay), but it is silent with regard to the parameters C, c that specify its precise shape. In particular, it is not clear in what manner the class and the distribution 'interact' to produce these parameters. This is where the present chapter comes in.

Main Results

Our main contributions are:

1. We solve the main question left open by Antos and Lugosi (1998, 1996). We define a new combinatorial dimension that we call the Vapnik–Chervonenkis–Littlestone dimension, or VCL, and show that it characterizes the magnitude of the strong minimax lower bound up to universal constants, as follows.

   There exist universal constants $\alpha, \beta > 0$ such that for any VC class $\mathcal{H}$, the dimension $d = \mathsf{VCL}(\mathcal{H}) \leq \mathsf{VC}(\mathcal{H})$, and the number d' defined in question Question 2.1.1 satisfies $\alpha \cdot d \leq d' \leq \beta \cdot d$.

2. More generally, we introduce a more refined characterization of distribution-dependent learning curves. For any class $\mathcal{H}$, if the dimension $d = \mathsf{VCL}(\mathcal{H}) \geq 0$ is finite, then the optimal expected loss Opt can be bounded by

$$\frac{\Omega(d)}{n} \leq \mathsf{Opt} \leq \frac{O(d)}{n} + C \cdot e^{-c \cdot n}, \tag{2.1}$$

where $n \in \mathbb{N}$ is the number of i.i.d. samples used, and the inequalities hold as follows: for any learning algorithm there exists a distribution such that the lower bound holds for infinitely many values of n; the upper bound holds for a learning algorithm that we present, for all distributions and all $n \in \mathbb{N}$; the parameters $C, c \geq 0$ depend on $\mathcal{H}$ and the distribution, and the $\Omega(\cdot)$ and $O(\cdot)$ notations hide universal multiplicative constants that are independent of $\mathcal{H}$ and of the distribution.

This bound captures both linear rates (when $d > 0$) and exponential rates (when $d = 0$). We call this type of bound the *fine-grained* rate of $\mathcal{H}$, to distinguish it from the notion of *coarse* rate used in Theorem 2.1.2. See Definition 2.2.16 and Theorem 2.3.1 for a formal statement of this result.

3. Furthermore, for the hard distribution that satisfies the lower bound in Eq. (2.1), the marginal on the domain $\mathcal{X}$ depends only on the class $\mathcal{H}$. In particular, in contrast to the lower bounds of Bousquet et al. (2021), the marginal on the domain does not depend on the learning algorithm. Conceptually, this means that in the distribution-dependent learning curve setting, access to unlabeled data is not helpful for learning classes with finite VCL dimension. Namely, semi-supervised learning and supervised learning require the same number of labeled samples.

   We note that this is a non-trivial result, employing a sophisticated application of Fatou's lemma which enables reversing the order of quantifiers, as well as an argument from Ramsey theory.

4. For any class $\mathcal{H}$, if $\mathsf{VCL}(\mathcal{H}) = \infty$ and $\mathcal{H}$ does not shatter an infinite strong VCL tree (see Definitions 2.2.12 and 2.2.13 below), then $\mathcal{H}$ has a *strongly distribution-dependent linear rate*. Namely, for every $c \geq 0$ there exists a distribution such that $\mathsf{Opt} \geq c/n$ for infinitely many $n \in \mathbb{N}$ (see Definition 2.2.18 and Theorem 2.3.1).[2]

5. We offer an equivalent formulation of our results in a language that is closer to the traditional PAC framework. This provides another viewpoint on our work and how it compares to traditional PAC bounds. See Theorem 2.3.5.

6. As a special case, we recover the lower bound of Antos and Lugosi (1998) for half-spaces. We do so by introducing a technique for proving strong lower bounds via a 'fractal' argument, which may be useful for other classes as well. (See Theorem 2.3.3 and Section 2.5.)

Benefits of the New Characterization

Our upper bound in Eq. (2.1) offers a refinement and reinterpretation of Theorem 2.1.2:

[2] In the remaining case where $\mathsf{VCL}(\mathcal{H}) = \infty$ and $\mathcal{H}$ has an infinite strong VCL tree, $\mathcal{H}$ requires arbitrarily slow rates, as shown by Bousquet et al. (2021).

- **Upper bound refinement.** For the case of linear rates, Bousquet et al. (2021) showed an upper bound of $\frac{c}{n}$, where $c \geq 0$ depends both on the class and on the distribution. In contrast, our expression in the upper bound constitutes a *decomposition* of the error rate into a linear component that depends only on the class, and an exponential component that depends on the class and on the distribution. We view this as a step towards a complete characterization of the optimal distribution-dependent learning rate for supervised learning.

- **Upper bound reinterpretation.** Whereas Theorem 2.1.2 depicts exponential rates and linear rates as being two entirely different beasts, Eq. (2.1) presents them in a more unified light, with exponential rates constituting the special case of ${}^{d}/_{n}$ where $d = 0$.

Our lower bound in Eq. (2.1) offers meaningful improvements over both the previous distribution-dependent lower bound and over the classic 'no free lunch' lower bounds from PAC learning, and also constitutes a partial unification of these two results.

- **Quantitative strengthening of distribution-dependent lower bounds.** For classes with linear learning rates, the best previously known distribution-dependent lower bound that applies to general classes was $\Omega(1/n)$ (Bousquet et al., 2021). This applies equally to all classes that have linear rates, and does not distinguish between different degrees of hardness within that broad set of classes. In contrast, we are able to prove a lower bound of $\mathsf{Opt} \geq \Omega(d/n)$, for d that depends only on the class and is tight up to a universal multiplicative constant (independent of the class and of the distribution).

- **Qualitative strengthening of distribution-dependent lower bounds.** Classic PAC lower bounds (discussed further in the next bullet) have the following formulation:

  > *There exists a distribution $\mathcal{D}_{\mathcal{X}} \in \Delta(\mathcal{X})$ such that for any learning algorithm A there exists a hard distribution $\mathcal{D} \in \Delta(\mathcal{X} \times \{0,1\})$ such that the marginal distribution of $\mathcal{D}$ on $\mathcal{X}$ equals $\mathcal{D}_{\mathcal{X}}$, and the loss of A on distribution $\mathcal{D}$ is large.*[3] $(\star)$

  In contrast, the linear lower bound of Bousquet et al. (2021) offered a weaker statement:

  > *For any learning algorithm A there exists a hard distribution $\mathcal{D} \in \Delta(\mathcal{X} \times \{0,1\})$ such that the loss of A on distribution $\mathcal{D}$ is large.*

  In particular, this weaker formulation left open the possibility that an algorithm that has access to unlabeled samples (as in the semi-supervised learning setting) could beat the lower bound. We show that that is not the case. We strengthen the lower bound of Bousquet et al. (2021), obtaining the stronger formulation as in $(\star)$.

[3] The marginal distribution $\mathcal{D}_{\mathcal{X}}$ is simply a uniform distribution over a subset of the domain of cardinality $\mathsf{VC}(\mathcal{H})$ that is shattered by $\mathcal{H}$ in the VC sense.

- **Qualitative strengthening of PAC lower bounds.** Classic PAC learning theory features ‘no free lunch’ lower bounds (e.g., Shalev-Shwartz and Ben-David, 2014, Theorem 5.1), which imply the VC lower bound appearing in the fundamental theorem of PAC learning (e.g., Shalev-Shwartz and Ben-David, 2014, Theorem 6.8, Item 3). These lower bounds leave something to be desired.

  To see this, fix a hypothesis class of VC dimension d. The VC bound states that for every $\varepsilon > 0$ there exists a hard (worst-case) distribution $\mathcal{D}_\varepsilon$ such that achieving loss at most ε with high probability requires at least $\Omega(d/\varepsilon)$ i.i.d. samples from $\mathcal{D}_\varepsilon$. Namely, for a fixed hypothesis class and a sequence of positive values $\varepsilon_1, \varepsilon_2, \dots$ there exists a sequence of *distinct* hard distributions $\mathcal{D}_1, \mathcal{D}_2, \dots$ such that each $\mathcal{D}_i$ is a hard distribution for achieving loss ε_i — but it is typically not a hard distribution for other values of ε.

  Clearly, the type of lower bound studied in VC bounds is strictly weaker than the distribution-dependent lower bounds studied in this chapter, where there exists a specific hard distribution such that the lower bound holds for infinitely many values of ε. And as we argued above, instance specific lower bounds are a better match to the reality of most machine learning practitioners, who typically face a specific (fixed) unknown distribution, and would like to calculate how many samples are necessary for obtaining loss ε_1, or loss ε_2, or loss ε_3, etc. — all with respect to the *same* fixed unknown distribution.

  Thus, an interesting open question is “for which VC classes is it possible to obtain distribution-dependent linear lower bounds of $\Omega(d/n)$?” (where $d = \mathsf{VC}(\mathcal{H})$ and the bound holds for a single distribution for infinitely many $n \in \mathbb{N}$). This question, which was studied by Antos and Lugosi (1996, 1998), is answered by our characterization as follows. Let $\mathcal{H}$ be a VC class with $0 \leq d' = \mathsf{VCL}(\mathcal{H}) \leq \mathsf{VC}(\mathcal{H}) = d$. If $d' > 0$ then $\mathcal{H}$ has a distribution-dependent linear lower bound of $\Omega({}^{d'}\!/_{n})$. Otherwise, if $d' = 0$ then $\mathcal{H}$ does not have a distribution-dependent linear lower bound; rather, each learning curve decays exponentially and the upper envelope of all the learning curves decays linearly as $\Theta({}^{d}\!/_{n})$. In this sense, our results offer a unified perspective of PAC and distribution-dependent lower bounds.

Related Works

Universal Learning. Our work explores the distribution-dependent setting, also called the *universal learning* setting, which was recently formalized by Bousquet et al. (2021). However, it is worthwhile to note that this framework has been studied by earlier works as well.

Schuurmans (1997) revealed the distinction between exponential and linear rates in the universal setting. In more detail, Schuurmans (1997) characterized the optimal learning rate for classes that are *concept chains*, namely, classes $\mathcal{H}$ such that for every $h_1, h_2 \in \mathcal{H}$, either $h_1 \leq h_2$ everywhere or $h_2 \leq h_1$ everywhere.

van Handel (2013) studied the uniform convergence property via the universal lens. He characterizes those hypothesis classes $\mathcal{H}$ satisfying that the empirical losses of *all* hypotheses in the class simultaneously and uniformly converge to the corresponding population losses as the number of examples tends to infinity. The difference with the (more common) distribution-free uniform convergence is that in the universal variant, the rate of the uniform convergence can depend on the source distribution.

Universal learning is related to *universal consistency*. A learning rule is universally consistent if its expected loss converges to the Bayes optimal risk for every target distribution. In other words, such algorithms learn every distribution (but at a distribution-dependent rate). Stone (1977) showed that such learning is possible, and in particular he established the universal consistency of several algorithms, including histogram, kernel and k-nearest neighbor rules. See Devroye, Györfi, and Lugosi (1996); Bousquet et al. (2021) for further discussion.

No Free Lunch. One of the technical contributions in this work is the identification of the VCL dimension as the combinatorial parameter which characterizes when a strong form of the 'no free lunch' theorem holds. That is, for which classes is it the case that there exists a single fixed distribution which witnesses the strongest lower bound on the error rate for infinitely many sample sizes n.

The work by Antos and Lugosi (1998) explored this question for VC classes; that is, they asked for which VC classes such a strong 'no free lunch' theorem holds. Antos and Lugosi (1998) showed that d-dimensional half-spaces satisfy such a strong 'no free lunch' theorem by proving a lower bound of d/n on the learning rate. (Schuurmans, 1997 also established such a bound in the 1-dimensional case.) However, a characterization of VC classes with this property remained open; in fact, Antos and Lugosi (1998) explicitly concluded that it is "neither the VC dimension nor the rate of increase of shatter coefficients that determine the asymptotic behavior of the concept class". Our work resolves this question by showing that the VCL dimension determines this behavior.

Strong Minimax. A recent work by Ben-David and Blais (2020) studies a similar type of lower bounds for the task of computing boolean functions up to error ε. They introduce a new type of minimax theorem which provides a single hard distribution for arbitrarily small ε.

2.2 Preliminaries

Notation 2.2.1. $\mathbb{N} = \{1, 2, 3, \ldots\}$, *i.e.,* $0 \notin \mathbb{N}$. *For any* $n \in \mathbb{N}$, *we denote* $[n] = \{1, 2, 3, \ldots, n\}$.

Notation 2.2.2. *Let* $\mathcal{X}$ *be a set. We write* $\mathcal{X}^* = \cup_{t=0}^{\infty} \mathcal{X}^t$ *to denote the set of all finite strings or finite vectors with elements from* $\mathcal{X}$. $\mathcal{X}^*$ *includes the empty string, which we denote by* λ.

Notation 2.2.3. *For a set* $\mathcal{X}$, *we write* $\Delta(\mathcal{X})$ *to denote the set of all distribution with support contained in* $\mathcal{X}$ *(with respect to some fixed* σ*-algebra).*

Notation 2.2.4. *For a (finite or infinite) vector* $\mathbf{x} = (x_1, x_2, \dots)$*, we write* $\mathbf{x}_{\leq t}$ *to denote the finite prefix* $(x_1, x_2, \dots, x_t)$*;* $\big((\mathbf{x}_t, \mathbf{y}_t)\big)_{t\in\mathbb{N}}$ *denotes an infinite sequence of pairs of vectors, where for each* t*,* $(\mathbf{x}_t, \mathbf{y}_t)$ *is a pair of vectors; for a (finite or infinite) sequence of pairs of vectors, we denote a finite prefix of the sequence by* $(\mathbf{x}, \mathbf{y})_{\leq t} = \big((\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2) \dots, (\mathbf{x}_t, \mathbf{y}_t)\big)$*.*

Traditional Learning Theory

Definition 2.2.5. *Let* $\mathcal{X}$ *be a set, and let* $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ *be a set of functions. Let* $k \in \mathbb{N}$*,* $X = \{x_1, x_2, \dots, x_k\} \subseteq \mathcal{X}$*. We say that* *$\mathcal{H}$ shatters X* *if for any* $y_1, y_2, \dots, y_k \in \{0,1\}$ *there exists* $h \in \mathcal{H}$ *such that* $h(x_i) = y_i$ *for all* $i \in [k]$*. The Vapnik–Chervonenkis (VC) dimension of* $\mathcal{H}$*, denoted* $\mathsf{VC}(\mathcal{H})$*, is the largest* $d \in \mathbb{N}$ *for which there exist a set* $X \subseteq \mathcal{X}$ *of cardinality* d *that is shattered by* $\mathcal{H}$*. If* $\mathcal{H}$ *shatters sets of cardinality arbitrarily large, we say that* $\mathsf{VC}(\mathcal{H}) = \infty$*.*

Definition 2.2.6. *Let* $\mathcal{X}$ *be a set. A learning algorithm for functions* $\mathcal{X} \to \{0,1\}$ *is an algorithm* $\widehat{h}$ *that takes a sample* $S \in (\mathcal{X} \times \{0,1\})^*$ *and outputs a function* $\widehat{h}_S : \mathcal{X} \to \{0,1\}$*. The mapping* $S \mapsto \widehat{h}_S$ *may be randomized.*

Definition 2.2.7. *Let* $\mathcal{X}$ *be a set, let* $\mathcal{D} \in \Delta(\mathcal{X} \times \{0,1\})$*, and let* $h : \mathcal{X} \to \{0,1\}$ *be a function. The 0-1 loss of* $\mathcal{H}$ *with respect to* $\mathcal{D}$ *is* $\mathrm{L}_{\mathcal{D}}^{0\text{-}1}(\mathcal{H}) = \mathbb{P}_{(x,y)\sim\mathcal{D}}[h(x) \neq y]$*.*

Definition 2.2.8. *Let* $\mathcal{X}$ *be a set, and let* $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ *be a class of functions. The set of realizable distributions for* $\mathcal{H}$ *is*

$$\mathsf{Realizable}(\mathcal{H}) = \left\{\mathcal{D} \in \Delta(\mathcal{X} \times \{0,1\}) : \inf_{h\in\mathcal{H}} \mathrm{L}_{\mathcal{D}}^{0\text{-}1}(h) = 0\right\}.$$

The VCL Dimension

Definition 2.2.9. *Let* $\mathcal{X}$ *be a set, let* $d \in \mathbb{N}$ *and* $\ell \in \mathbb{N} \cup \{0, \infty\}$*. A* d*-VCL tree of depth* ℓ *with respect to* $\mathcal{X}$ *is a set*

$$T = \left\{\mathbf{x}_{\mathbf{u}} \in \mathcal{X}^d : \ \mathbf{u} \in \{0,1\}^{ds}, \ s \in \mathbb{N} \cup \{0\}, \ s \leq \ell\right\}. \tag{2.2}$$

We say that *T is infinite* *if it has depth* $\ell = \infty$*.*

Note that a d-VCL tree of depth 0 is not empty, rather it contains a single node $\mathbf{x}_\lambda$ where λ denotes the empty string.

Definition 2.2.10. *Let* $\mathcal{X}$ *be a set, let* $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ *be a hypothesis class, and let* $d \in \mathbb{N}$*. For each* $s \in \mathbb{N}$*, let* $\mathbf{x}_s = (x_s^1, \dots, x_s^d) \in \mathcal{X}^d$ *and* $\mathbf{y}_s = (y_s^1, \dots, y_s^d) \in \{0,1\}^d$*. Let* $h \in \mathcal{H}$*. For any* $t \in \mathbb{N}$*, we say that the finite sequence* $(\mathbf{x}, \mathbf{y})_{\leq t} = \big((\mathbf{x}_s, \mathbf{y}_s)\big)_{s=1}^{t}$ *is consistent with* h *if*

$$\forall s \in [t] \, \forall i \in [d] : \ h(x_s^i) = y_s^i. \tag{2.3}$$

We say that the infinite sequence $\big((\mathbf{x}_s, \mathbf{y}_s)\big)_{s\in\mathbb{N}}$ *is consistent with* $\mathcal{H}$ *if for any* $t \in \mathbb{N}$ *there exists* $h \in \mathcal{H}$ *such that* $(\mathbf{x}, \mathbf{y})_{\leq t}$ *is consistent with* h*.*

Definition 2.2.11. *Let* $\mathcal{X}$ *be a set, let* $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ *be a hypothesis class, and let* T *be a* d*-VCL tree as in Eq. (2.2). We say that* $\mathcal{H}$ *shatters* T *if for every* $t \in \mathbb{N}$*,* $t \leq \ell$*, and every* $\mathbf{y} \in \{0,1\}^{dt}$ *there exists a hypothesis* $h \in \mathcal{H}$ *that is consistent with* $\big((\mathbf{x}_{\mathbf{y}_{\leq s-1}}, \mathbf{y}_s)\big)_{s=1}^{t}$ *in the sense that*

$$\forall s \in [t]\, \forall j \in [d]: \;\; h(x^j_{\mathbf{y}_{\leq s-1}}) = y^j_s, \tag{2.4}$$

where we use the notation

$$\mathbf{y}_{\leq s} = \Big(\big(y^1_1, \ldots, y^d_1\big), \ldots, \big(y^1_s, \ldots, y^d_s\big)\Big) \in \{0,1\}^{ds}$$

to denote a prefix of $\mathbf{y}$*, and*

$$\mathbf{x}_{\mathbf{y}_{\leq s}} = \Big(x^1_{\mathbf{y}_{\leq s}}, \ldots, x^d_{\mathbf{y}_{\leq s}}\Big) \in \{0,1\}^d$$

to denote the members of $\mathbf{x}_{\mathbf{y}_{\leq s}}$*.*

The d-VCL trees used in this chapter are a variant of the trees used in Bousquet et al. (2021). To distinguish the two, we call their construction *strong* VCL trees.

Definition 2.2.12. *Let* $\mathcal{X}$ *be a set, let* $d \in \mathbb{N}$*. An infinite strong VCL tree with respect to* $\mathcal{X}$ *is a set*

$$T = \Big\{\mathbf{x}_{\mathbf{u}} \in \mathcal{X}^{s+1}: \;\; s \in \mathbb{N} \cup \{0\} \;\wedge\; \mathbf{u} \in \{0,1\}^1 \times \{0,1\}^2 \times \cdots \times \{0,1\}^s\Big\}.$$

Definition 2.2.13. *Let* $\mathcal{X}$ *be a set, let* $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ *be a hypothesis class, and let* T *be an infinite strong VCL tree as in Definition 2.2.12. We say that* $\mathcal{H}$ *shatters* T *if for every* $t \in \mathbb{N}$*, and every* $\mathbf{y} \in \{0,1\}^1 \times \{0,1\}^2 \times \cdots \times \{0,1\}^t$ *there exists a hypothesis* $h \in \mathcal{H}$ *that is consistent with* $\big((\mathbf{x}_{\mathbf{y}_{\leq s-1}}, \mathbf{y}_s)\big)_{s=1}^{t}$ *in the sense that*

$$\forall s \in [t]\, \forall j \in [s]: \;\; h(x^j_{\mathbf{y}_{\leq s-1}}) = y^j_s, \tag{2.5}$$

where we use the notation

$$\mathbf{y}_{\leq s} = \Big(\big(y^1_1\big), \big(y^1_2, y^2_2\big), \big(y^1_3, y^2_3, y^3_3\big), \ldots, \big(y^1_s, \ldots, y^s_s\big)\Big) \in \{0,1\}^{\left(\sum_{k=1}^{s} k\right)}$$

to denote a prefix of $\mathbf{y}$*, and*

$$\mathbf{x}_{\mathbf{y}_{\leq s}} = \Big(x^1_{\mathbf{y}_{\leq s}}, \ldots, x^{s+1}_{\mathbf{y}_{\leq s}}\Big) \in \{0,1\}^{s+1}$$

to denote the members of $\mathbf{x}_{\mathbf{y}_{\leq s}}$*.*

Definition 2.2.14. *Let* $\mathcal{X}$ *be a set and let* $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$*. The Vapnik–Chervonenkis–Littlestone dimension of* $\mathcal{H}$*, denoted* $\mathsf{VCL}(\mathcal{H})$*, is the largest integer* $d \geq 0$ *such that* $\mathcal{H}$ *shatters an infinite* d*-VCL tree. If* $\mathcal{H}$ *does not shatter any infinite* 1*-VCL tree, we say that* $\mathsf{VCL}(\mathcal{H}) = 0$*. If* $\mathcal{H}$ *shatters infinite* d*-VCL trees for* d *arbitrarily large, we say that* $\mathsf{VCL}(\mathcal{H}) = \infty$*.*

Learning Rates

Bousquet et al. (2021) used the following definition of distribution-dependent learning rates.

Definition 2.2.15 (Bousquet et al., 2021, Definition 1.4)**.** *Let $\mathcal{H}$ be a concept class, and let $R : \mathbb{N} \to [0,1]$ with $R(n) \to 0$ be a rate function.*

- *$\mathcal{H}$ is learnable at rate R if there exists a learning algorithm $\widehat{h}$ such that for every $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$, there exist $C, c \geq 0$ such that $\mathbb{E}_{S \sim \mathcal{D}^n}\left[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}\left(\widehat{h}_S\right)\right] \leq C \cdot R(c \cdot n)$ for all $n \in \mathbb{N}$.*

- *$\mathcal{H}$ is learnable with rate no faster than R if for every learning algorithm $\widehat{h}$, there exists a $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$ and $C, c > 0$ for which $\mathbb{E}_{S \sim \mathcal{D}^n}\left[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}\left(\widehat{h}_S\right)\right] \geq C \cdot R(c \cdot n)$ for infinitely many $n \in \mathbb{N}$.*

- *$\mathcal{H}$ is learnable with optimal rate R if $\mathcal{H}$ is learnable at rate R and $\mathcal{H}$ is not learnable faster than R.*

- *$\mathcal{H}$ requires arbitrarily slow rates if, for every $R(n) \to 0, \mathcal{H}$ is learnable at rate no faster than R.*

In this chapter we refine the notion of learning rates, introducing the following more nuanced expressions for linear rates, as follows. Note that our definitions are strictly special cases in the sense that if a class is learnable at rate (learnable at rate no faster than) d/n according to our definition, then it is learnable at rate (learnable at rate no faster than) d/n according to Definition 2.2.15 as well.

Definition 2.2.16. *Let $\mathcal{X}$ be a set, let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class, and let $d \geq 0$ and $\gamma \geq 1$. We say that:*

- *$\mathcal{H}$ is learnable with fine-grained rate d/n if there exists a learning algorithm $\widehat{h}$ such that for any distribution $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$ there exist real numbers $C, c \geq 0$ such that for all $n \in \mathbb{N}$:*
$$\mathbb{E}_{S \sim \mathcal{D}^n}\left[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}\left(\widehat{h}_S\right)\right] \leq \frac{d}{n} + C \cdot \exp(-cn).$$

- *$\mathcal{H}$ is learnable with fine-grained rate no faster than d/n if for any learning algorithm $\widehat{h}$ there exists a distribution $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$ such that the inequality*
$$\mathbb{E}_{S \sim \mathcal{D}^n}\left[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}\left(\widehat{h}_S\right)\right] \geq \frac{d}{n}$$
holds for infinitely many $n \in \mathbb{N}$.

- *$\mathcal{H}$ is learnable with optimal fine-grained rate d/n with gap factor γ if $\mathcal{H}$ is learnable with rate no faster than d/n, and is learnable with rate d'/n, where $d' \leq \gamma d$.*

To distinguish the two notions of rate, we will refer to the rates of Definition 2.2.15 as *coarse rates.*

Remark 2.2.17. *Ideally, we would like to obtain a gap factor γ that is as close as possible to 1, so that $d = d'$ (see Definition 2.2.16). The extent to which this is possible is a topic for further research. Throughout this chapter we use $\gamma = 800$.*

Definition 2.2.18. *Let $\mathcal{X}$ be a set, let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class. We say that $\mathcal{H}$ is learnable with a strongly distribution-dependent linear rate if for any (possibly randomized) learning algorithm $\widehat{h}$ and any $c \geq 0$ there exists $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$ such that the inequality*

$$\frac{c}{n} \leq \mathbb{E}_{S \sim \mathcal{D}^n}\left[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}\left(\widehat{h}_S\right)\right] \tag{2.6}$$

holds for infinitely many $n \in \mathbb{N}$.

Remark 2.2.19. *There are various technical issues related to measure theory that arise in the distribution-dependent learning setting and are germane to our results. We use the same assumptions as Bousquet et al. (2021), and refer the interested reader to their work for an in-depth discussion (e.g., Section 3.3 and Appendices B and C).*

Gale–Stewart Games

We will use some basic concepts and results concerning infinite games. We refer the reader to Appendix A.1 of Bousquet et al. (2021) for additional references and discussion. Briefly, we consider infinite full information two-player games, in which there exists a set Ω and a subset $W \subseteq \Omega^{\mathbb{N}}$, and at each time $t = 1, 2, 3, \dots$, Player 1 selects an item $x_t \in \Omega$, and then Player 2 selects an item $y_t \in \Omega$. Player 1 wins if and only if the resulting infinite sequence $\mathbf{z} = (x_1, y_1, x_2, y_2, \dots)$ satisfies $\mathbf{z} \in W$; otherwise, Player 2 wins.

We say that Player i has a *winning strategy* if there exists a function $f : \Omega^* \to \Omega$ such that if in every time $t \in \mathbb{N}$, Player i selects item $f(\mathbf{z}')$ where $\mathbf{z}'$ is the finite sequence of all items selected so far (by both players), then Player i wins the game (regardless of the selections made by the other player).

A game is called *determined* if precisely one of the players has a winning strategy. An infinite game is called *Gale–Stewart* (or *finitely-decidable*) if for every $\mathbf{w} \in W$ there exists $t \in \mathbb{N}$ such that for any infinite suffix $\backslash \in \Omega^{\mathbb{N}}$, $\mathbf{w}_{\leq t} \circ \backslash \in W$, where '$\circ$' denotes concatenation. Namely, every member of W has a finite prefix that certifies its membership in W. We will use the following result.

Theorem 2.2.20 (Gale and Stewart, 1953)**.** *Every Gale–Stewart game is determined.*

2.3 Technical Overview

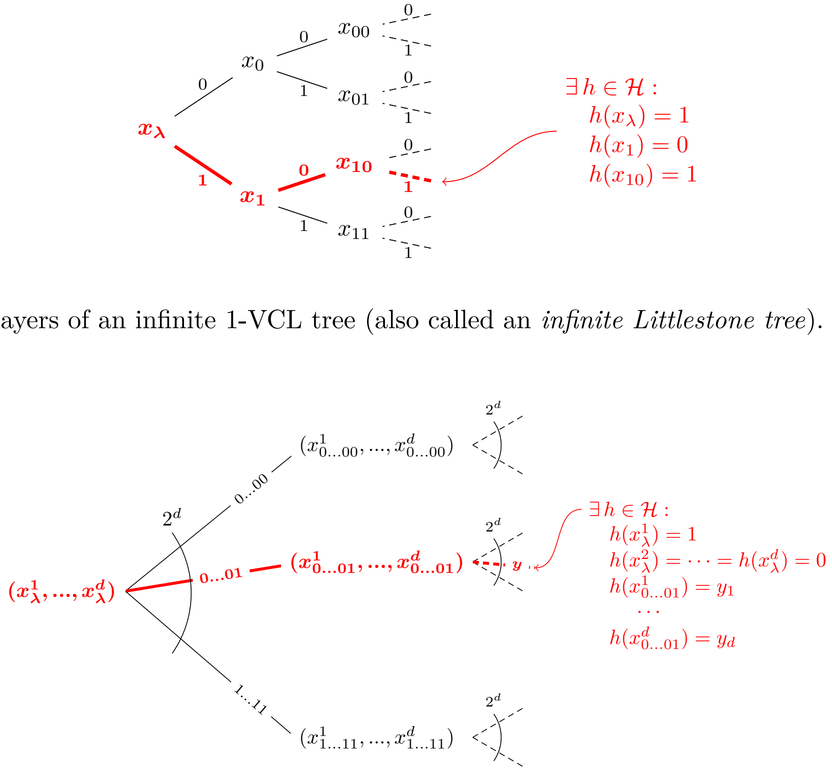
Our first result is the characterization of fine-grained learning rates via the VCL dimension.

Theorem 2.3.1. *There exist constants $\alpha, \beta > 0$ as follows. Let $\mathcal{X}$ be a set, let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class, and let $d = \mathsf{VCL}(\mathcal{H})$.*

1. *If $d < \infty$ then $\mathcal{H}$ is learnable with optimal fine-grained rate d/n with gap factor $\gamma = \beta/\alpha$; furthermore, the marginal of the hard distribution on $\mathcal{X}$ depends only on $\mathcal{H}$. Namely, there exists $\mathcal{D}_{\mathcal{X}} \in \Delta(\mathcal{X})$ such that for any (possibly randomized) learning algorithm $\widehat{h}$ there exists $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$ such that the marginal distribution of $\mathcal{D}$ on $\mathcal{X}$ is $\mathcal{D}_{\mathcal{X}}$, and the inequality*
$$\alpha \cdot \frac{d}{n} \leq \mathbb{E}_{S \sim \mathcal{D}^n}\left[\mathrm{L}^{0\text{-}1}_{\mathcal{D}}\left(\widehat{h}_S\right)\right]$$
holds for infinitely many $n \in \mathbb{N}$; and there exists a learning algorithm h^* such that for any $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$ there exist parameters $C, c > 0$ such that
$$\forall n \in \mathbb{N}: \ \mathbb{E}_{S \sim \mathcal{D}^n}\left[\mathrm{L}^{0\text{-}1}_{\mathcal{D}}(h^*_S)\right] \leq \beta \cdot \frac{d}{n} + C \cdot e^{-c \cdot n}.$$
2. *Otherwise, if $\mathcal{H}$ does not shatter an infinite strong VCL tree, then $\mathcal{H}$ is learnable with a strongly distribution-dependent linear rate. Namely, for any (possibly randomized) learning algorithm $\widehat{h}$ and any $c > 0$ there exists $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$ such that the inequality*
$$\frac{c}{n} \leq \mathbb{E}_{S \sim \mathcal{D}^n}\left[\mathrm{L}^{0\text{-}1}_{\mathcal{D}}\left(\widehat{h}_S\right)\right]$$
holds for infinitely many $n \in \mathbb{N}$; and there exists a learning algorithm h^* such that for any $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$ there exists $c > 0$ such that
$$\forall n \in \mathbb{N}: \ \mathbb{E}_{S \sim \mathcal{D}^n}\left[\mathrm{L}^{0\text{-}1}_{\mathcal{D}}(h^*_S)\right] \leq \frac{c}{n}.$$
3. *Otherwise, $\mathcal{H}$ requires arbitrarily slow rates.*

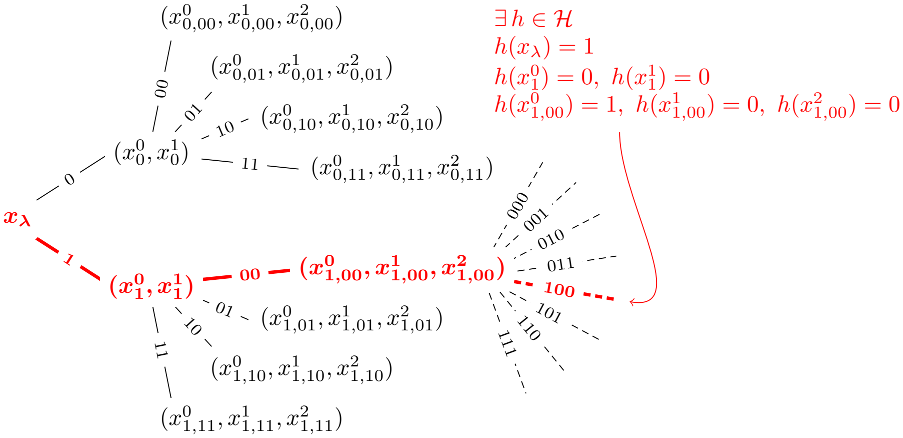
Remark 2.3.2. *Our proofs use $\alpha = 1/100$, $\beta = 8$, and $\gamma = 800$.*

All proofs appear in Sections 2.4 and 2.5. Additionally, we provide a brief overview of the main ideas in each proof.


x_λ
0
1
x_0
x_1
x_{00}
x_{01}
x_{10}
x_{11}
$\exists\, h \in \mathcal{H}:$
$h(x_\lambda) = 1$
$h(x_1) = 0$
$h(x_{10}) = 1$
$(x^1_\lambda, \ldots, x^d_\lambda)$
2^d
0…00
0…01
1…11
$(x^1_{0\ldots00}, \ldots, x^d_{0\ldots00})$
$(x^1_{0\ldots01}, \ldots, x^d_{0\ldots01})$
$(x^1_{1\ldots11}, \ldots, x^d_{1\ldots11})$
y
$\exists\, h \in \mathcal{H}:$
$h(x^1_\lambda) = 1$
$h(x^2_\lambda) = \cdots = h(x^d_\lambda) = 0$
$h(x^1_{0\ldots01}) = y_1$
$\cdots$
$h(x^d_{0\ldots01}) = y_d$

(a) The first few layers of an infinite 1-VCL tree (also called an *infinite Littlestone tree*).

(b) The first few layers of an infinite d-VCL tree. Every arc represents 2^d children, one for each possible labeling of the d points in the preceding node. Introduced in this chapter, d-VCL trees form the basis for our novel characterization.


$(x^0_{0,00}, x^1_{0,00}, x^2_{0,00})$
$(x^0_{0,01}, x^1_{0,01}, x^2_{0,01})$
$(x^0_{0,10}, x^1_{0,10}, x^2_{0,10})$
$(x^0_{0,11}, x^1_{0,11}, x^2_{0,11})$
00
01
10
11
(x^0_0, x^1_0)
0
x_λ
1
(x^0_1, x^1_1)
00
$(x^0_{1,00}, x^1_{1,00}, x^2_{1,00})$
01
10
11
$(x^0_{1,01}, x^1_{1,01}, x^2_{1,01})$
$(x^0_{1,10}, x^1_{1,10}, x^2_{1,10})$
$(x^0_{1,11}, x^1_{1,11}, x^2_{1,11})$
000
001
010
011
100
101
110
111
$\exists\, h \in \mathcal{H}$
$h(x_\lambda) = 1$
$h(x^0_1) = 0,\ h(x^1_1) = 0$
$h(x^0_{1,00}) = 1,\ h(x^1_{1,00}) = 0,\ h(x^2_{1,00}) = 0$

(c) The first few layers of an infinite strong VCL tree. (In Bousquet et al. (2021) this structure was called an *infinite VCL tree*. We add the modifier *strong* to distinguish it from d-VCL trees.)

Figure 2.2: VCL trees. Every finite branch is consistent with a concept $h \in \mathcal{H}$. This is illustrated here for one branch in each tree, shown in red.

Source: Bousquet et al. (2021), adapted with permission.

The proof of Theorem 2.3.1 is similar to the proof of Theorem 2.1.2 from Bousquet et al. (2021). One of the main differences is that we use d-VCL trees for the characterization. Our d-VCL trees (introduced in this chapter, see Definition 2.2.9 and Figure 2.2b), are an intermediary refinement that lies between the 1-VCL trees and the strong VCL trees that were used in their proof. Identifying that this particular combinatorial structure characterizes the fine-grained rate is a non-trivial contribution of this chapter.

Proving the lower bound of Theorem 2.3.1 requires some technical improvements upon the technique of Bousquet et al. (2021). For each learning algorithm, they constructed a hard distribution that is concentrated on an infinite 'target' branch chosen at random in the d-VCL tree, and argued that if a test point is deeper in the tree than all points in the training set, then the leaner will make an incorrect prediction on that test point with probability 1/2. That approach is not suitable for constructing a single marginal distribution $\mathcal{D}_\mathcal{X} \in \Delta(\mathcal{X})$ that is hard for all learning algorithms (because for every target branch there exists an algorithm that returns a hypothesis with low loss on that branch). Instead, we choose a marginal $\mathcal{D}_\mathcal{X}$ that is distributed roughly evenly over all branches in the tree, and construct a distribution $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$ that has marginal $\mathcal{D}_\mathcal{X}$, and has labels corresponding to an infinite target branch in the tree.

In a general d-VCL tree, our approach would be problematic, because even if a test point is deeper in the tree than all the points in the training set, the labels for points in the training set *that do not belong to the target branch* could provide information about the target branch.[4] We overcome this problem as described in the following proof idea.

Proof idea for Theorem 2.3.1. For the upper bound, we show that the nonexistence of d-VCL trees is equivalent to the existence of a winning strategy for the learner in an infinite two-player game called the 'forbidden pattern game'. The equivalence is established via an intermediary online-learning game that is easier to analyze because it is a Gale-Stewart game (whereas the forbidden pattern game is not). A winning strategy for the learner in the forbidden pattern game can be converted into a learning algorithm for the distribution-dependent learning setting, by way of the one-inclusion algorithm of Haussler, Littlestone, and Warmuth (1994). The resulting algorithm has the desired rate of at most d/n.

For the lower bound, we use an elementary lemma from Ramsey theory to show that if $\mathcal{H}$ shatters an infinite d-VCL tree, then it also shattered an infinite d-VCL tree that satisfies an additional *indifference* property that we define. This property implies that for any infinite branch in the d-VCL tree, labels for points that do not belong to the branch provide no information about the labels for points that appear lower down in the tree along the branch. Therefore, when the target branch is chosen randomly, we can argue that if the test point appears on the target branch and is lower than all the training samples, then the learner will make an incorrect prediction with probability 1/2. The lower bound also involves a specific choice of parameters for the hard marginal distribution that enables a delicate application of Fatou's lemma. □

As a corollary of our characterization, we recover the lower bound of Antos and Lugosi (1998) for half-spaces in $\mathbb{R}^d$, up to a constants factor.

Theorem 2.3.3 (Antos and Lugosi, 1998, Corollary 1)**.** *There exists a constant $\alpha > 0$ as follows. Let $d \in \mathbb{N}$ and $\mathcal{X} = \mathbb{R}^d$. Let $\mathcal{H}_d \subseteq \{0,1\}^{\mathcal{X}}$ be the set of closed half-spaces in $\mathbb{R}^d$. For any learning algorithm $\widehat{h}$ there exists a distribution $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H}_d)$ such that the inequality*

$$\mathbb{E}_{S\sim\mathcal{D}^n}\left[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}\left(\widehat{h}_S\right)\right] \geq \alpha \cdot \frac{d}{n}$$

holds for infinitely many values $n \in \mathbb{N}$.

Proof idea. It suffices to show that $\mathsf{VCL}(\mathcal{H}_d) = d$. We consider the dual class for $\mathcal{H}_d$, and show via a neat 'fractal' argument that one can construct an infinite d-VCL tree for $\mathcal{H}_d$. □

We believe that the 'fractal' argument from this proof could be used to construct d-VCL trees for other classes as well.

[4] Consider the case where for some $x \in \mathcal{X}$ there exist infinite branches $\mathbf{y}^{(0)}$ and $\mathbf{y}^{(1)}$ in the tree, both of which do not contain x, such that for all $b \in \{0,1\}$, it holds that all $h \in \mathcal{H}$ that are consistent with $\mathbf{y}^{(b)}$ satisfy $h(x) = b$. Then knowing the label for x allows the learner to eliminate one of the branches.

Finally, we present another, mostly equivalent viewpoint on our results. Stated in a language that is closer to standard PAC learning, it enables a comparison between PAC bounds and distribution-dependent bounds.

Definition 2.3.4. *Let $\mathcal{X}$ be a set and let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class. Let $m : (0,1)^2 \to \mathbb{N}$ and $k : \mathsf{Realizable}(\mathcal{H}) \to \mathbb{N}$ be functions. We say that $\underline{\mathcal{H}}$ is eventually PAC learnable with sample complexity m and kick-in time k if there exist an algorithm $\widehat{h}$ such that for any distribution $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$ and for any $\varepsilon, \delta \in (0,1)$ the inequality*

$$\mathbb{P}_{S \sim \mathcal{D}^n}\left[\mathrm{L}^{0\text{-}1}_{\mathcal{D}}\left(\widehat{h}_S\right) \leq \varepsilon\right] \geq 1 - \delta$$

holds for all $n \geq \max\{m(\varepsilon,\delta), k(\mathcal{D})\}$.

Theorem 2.3.5. *There exist constants $\alpha, \beta > 0$ as follows. Let $\mathcal{X}$ be a set, let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class, and let $d \in \mathbb{N}$.*

1. *If $d = \mathsf{VCL}(\mathcal{H}) < \infty$ then $\mathcal{H}$ is eventually PAC learnable with sample complexity $m(\varepsilon,\delta) \leq \alpha d \log(1/\delta)/\varepsilon$.*
2. *If $\mathcal{H}$ is eventually PAC learnable with sample complexity*

$$m(\varepsilon,\delta) = d \log(1/\delta)/\varepsilon,$$

then $\mathsf{VCL}(\mathcal{H}) \leq \beta d$.

Proof idea. This follows from Theorem 2.3.1, together with a standard amplification argument for converting an algorithm with bounded expected error to a PAC learner. □

2.4 Proof of the Fine-Grained Characterization

Upper Bound

Throughout this section, let $\mathcal{X}$ be a set, let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class, and let $d \in \mathbb{N}$.

Definition 2.4.1. *The online learning game for $\mathcal{H}$ of size d, denoted $\mathrm{G}^{\mathsf{online}}_d(\mathcal{H})$, is an infinite full information game played between two players, a* learner *and an* adversary*. At each time step $t = 1, 2, 3, \ldots$:*

1. *The adversary chooses $\mathbf{x}_t = (x^1_t, \ldots, x^d_t) \in \mathcal{X}^d$.*
2. *The learner chooses $\mathbf{y}_t = (y^1_t, \ldots, y^d_t) \in \{0,1\}^d$.*

For each $t \in \mathbb{N}$, the version space is defined by

$$\mathcal{H}_t = \mathcal{H}_{\mathbf{x}_1,\mathbf{y}_1,\ldots,\mathbf{x}_t,\mathbf{y}_t} = \left\{h \in \mathcal{H} : \left(\forall s \in [t] \; \forall i \in [d] : \; h(x^i_s) = y^i_s\right)\right\}.$$

If there exists a time step $t \in \mathbb{N}$ such that $\mathcal{H}_t = \varnothing$ then the learner wins the game. Otherwise, the adversary wins the game.

Definition 2.4.2. *The forbidden pattern game for $\mathcal{H}$ of size d, denoted $\mathrm{G}_{\mathcal{H}}^{\mathsf{forbidden}}(d)$, is an infinite full information game played between two players, a* learner *and an* adversary*. At each time step $t = 1, 2, 3, \ldots$:*

1. *The adversary chooses $\mathbf{x}_t \in \mathcal{X}^d$.*
2. *The learner chooses $\widehat{\mathbf{y}}_t \in \{0,1\}^d$.*
3. *The adversary chooses $\mathbf{y}_t \in \{0,1\}^d$.*

The adversary wins the game if the adversary's infinite sequence $\Big((\mathbf{x}_t, \mathbf{y}_t)\Big)_{t\in\mathbb{N}}$ is consistent with $\mathcal{H}$ and $\widehat{\mathbf{y}}_t = \mathbf{y}_t$ for infinitely many $t \in \mathbb{N}$. Otherwise, the learner wins the game.

In other words, the learner wins the forbidden pattern game if $\widehat{\mathbf{y}}_t$ is eventually a 'forbidden pattern' that is not consistent with $\mathcal{H}$.

We show that the existence of d-VCL trees characterizes the winner in the forbidden pattern game. Note that while the online game is a Gale-Stewart game, the forbidden pattern game is not. This makes the online game a convenient stepping stone towards this characterization, as in the following claim.

Lemma 2.4.3. *The following conditions are equivalent:*

1. *There does not exist an infinite d-VCL tree with respect to $\mathcal{X}$ that is shattered by $\mathcal{H}$.*
2. *There exists a winning strategy for the leaner in the online game $\mathrm{G}_d^{\mathsf{online}}(\mathcal{H})$.*
3. *There exists a winning strategy for the leaner in the forbidden pattern game $\mathrm{G}_d^{\mathsf{forbidden}}(\mathcal{H})$.*

The proof of Lemma 2.4.3 is divided between Claims 2.4.4 to 2.4.7.

Claim 2.4.4. *There exists an infinite d-VCL tree with respect to $\mathcal{X}$ that is shattered by $\mathcal{H}$ if and only if there exists a winning strategy for the adversary in $\mathrm{G}_d^{\mathsf{online}}(\mathcal{H})$.*

Proof. Assume that there exists an infinite d-VCL tree

$$T = \Big\{\mathbf{x}_{\mathbf{u}} \in \mathcal{X}^d : \ \mathbf{u} \in \Big(\{0,1\}^d\Big)^*\Big\} \tag{2.7}$$

that is shattered by $\mathcal{H}$. This implies the existence of a winning strategy for the adversary as following. At each time step $t \in \mathbb{N}$, the adversary selects $\mathbf{x}_t = \mathbf{x}_{\mathbf{y}_{\leq t-1}}$. For any choice $\mathbf{y}_t \in \{0,1\}^d$ made by the learner the version space remains not empty, i.e., $\mathcal{H}_t \neq \varnothing$. This holds because $\mathcal{H}$ shatters T, and so in particular there exists a hypothesis $h \in \mathcal{H}$ that is consistent with $\Big((\mathbf{x}_{\mathbf{y}_{\leq s-1}}, \mathbf{y}_s)\Big)_{s\in[t]}$. Hence, the adversary wins the game when playing according to this strategy.

Conversely, assume that there exists a winning strategy for the adversary defined by a function $f : \{0,1\}^* \to \mathcal{X}^d$ such that at any time step $t \in \mathbb{N}$, the adversary chooses $\mathbf{x}_t = f(\mathbf{y}_{\leq t-1})$, where $\mathbf{y}_{\leq t-1}$ is the sequence of choices the learner has made so far. The function f defines an infinite d-VCL tree T as in Eq. (2.7) given by $\mathbf{x}_\mathbf{u} = f(\mathbf{u})$. Seeing as this is a winning strategy for the adversary, $\mathcal{H}_t \neq \varnothing$ for all $t \in \mathbb{N}$ and all possible choices of $\mathbf{y}_{\leq t}$, and this implies that the tree T is shattered by $\mathcal{H}$. □

Claim 2.4.5. *In the context of Lemma 2.4.3, Item 1 $\iff$ Item 2.*

Proof.

$$\begin{aligned}(\text{Item } 1) &\iff \left(\nexists \text{ winning strategy for the adversary in } \mathrm{G}_d^{\mathsf{online}}(\mathcal{H})\right) \\ &\iff (\text{Item } 2)\,,\end{aligned}$$

where the first equivalence is by Claim 2.4.4, and the second equivalence states that the online learning game is determined, which is true by Theorem 2.2.20 because it is a Gale–Stewart game. □

Claim 2.4.6. *In the context of Lemma 2.4.3, Item 2 $\Longrightarrow$ Item 3.*

Assumption: $f : \left(\bigcup_{s=1}^{\infty} \mathcal{X}^{ds}\right) \to \{0,1\}^d$ is a function that defines a winning strategy for the learner in the online game $\mathrm{G}_{\mathcal{H}}^{\mathsf{online}}(d)$.

```
ForbiddenPatternLearner:
  $\xi \leftarrow$ empty sequence
  $\eta \leftarrow$ empty sequence
  for  $t \leftarrow 1, 2, \ldots$ :
      Receive $\mathbf{x}_t$ from the adversary
      Choose $\widehat{\mathbf{y}}_t \leftarrow f(\xi \circ \mathbf{x}_t)$
      Receive $\mathbf{y}_t$ from the adversary
      if  $\widehat{\mathbf{y}}_t = \mathbf{y}_t$:
          $\xi \leftarrow \xi \circ \mathbf{x}_t$
          $\eta \leftarrow \eta \circ \mathbf{y}_t$
```

Algorithm 2.1: A reduction from a winning strategy for the forbidden pattern game to a winning strategy for the online game.

Proof idea. Use Algorithm 2.1. A winning strategy for the learner in the online game empties the version space. So eventually, for any $\mathbf{x}_t$ chosen by the adversary, the learner can choose a $\mathbf{y}_t$ such that $(\mathbf{x}, \mathbf{y})_{\leq t}$ is not consistent with $\mathcal{H}$. □

Proof. Let $f : \left(\bigcup_{s=1}^{\infty} \mathcal{X}^{ds}\right) \to \{0,1\}^d$ be a function that defines a winning strategy for the learner in the online game $\mathrm{G}_{\mathcal{H}}^{\mathsf{online}}(d)$. Namely, in the online game, if in each time step $t \in \mathbb{N}$

the adversary chooses $\mathbf{x}_t$ and the learner chooses $\mathbf{y}_t = f(\mathbf{x}_1, \dots, \mathbf{x}_t)$, then after a finite number of steps $\mathcal{H}_t = \varnothing$.

Given such a function f, Algorithm 2.1 defines a winning strategy for the learner in the forbidden pattern game. To see this, assume for contradiction that the strategy of Algorithm 2.1 is not a winning strategy for the learner, namely, assume that there exists a sequence $\big((\mathbf{x}_t, \mathbf{y}_t)\big)_{t\in\mathbb{N}}$ that is consistent with $\mathcal{H}$ and also $\widehat{\mathbf{y}}_t = \mathbf{y}_t$ for infinitely many t when $\widehat{\mathbf{y}}$ is chosen by the learner according to Algorithm 2.1. This implies that the sequences ξ and η defined by the algorithm are infinite.

We show that if the adversary in the online game plays this infinite sequence ξ and the learner plays according to the strategy f, then the adversary wins the game, in contradiction to the assumption that f defines a winning strategy for the learner in the online game.

Let $\xi = \xi_1, \xi_2, \dots$ and $\eta = \eta_1, \eta_2, \dots$ where $\xi_t \in \mathcal{X}^d$ and $\eta_t \in \{0,1\}^d$ for all $t \in \mathbb{N}$. By construction, $\big((\xi_t, \eta_t)\big)_{t\in\mathbb{N}}$ is consistent with $\mathcal{H}$ because it is a subsequence of $\big((\mathbf{x}_t, \mathbf{y}_t)\big)_{t\in\mathbb{N}}$. In particular, for any finite prefix $(\xi, \eta)_{\leq t}$ there exists a hypothesis $h \in \mathcal{H}$ that is consistent with $(\xi, \eta)_{\leq t}$. This implies that for any $t \in \mathbb{N}$, the version space $\mathcal{H}_t = \mathcal{H}_{\xi_1, \eta_1, \dots, \xi_t, \eta_t}$ is not empty. However, the sequence $\big((\xi_t, \eta_t)\big)_{t\in\mathbb{N}}$ is constructed by playing according to the strategy f, namely $\eta_t = f(\xi_1, \dots, \xi_t)$ for all $t \in \mathbb{N}$. We conclude that when the adversary in the online game plays ξ and the learner plays according to f, then $\mathcal{H}_t \neq \varnothing$ for all $t \in \mathbb{N}$, yielding the desired contradiction to the choice of f. □

Claim 2.4.7. *In the context of Lemma 2.4.3, Item 3 $\Longrightarrow$ Item 1.*

Proof. We show the contrapositive, namely, if there exists an infinite d-VCL tree shattered by $\mathcal{H}$ then there does not exist a winning strategy for the leaner in the forbidden pattern game $\mathrm{G}_d^{\textsf{forbidden}}(\mathcal{H})$ (this is similar to one of the directions in Claim 2.4.4). Indeed, let T be an infinite shattered tree as in Eq. (2.7). Then there exists a winning strategy for the adversary: at each time step $t \in \mathbb{N}$, the adversary chooses $\mathbf{x}_t = \mathbf{x}_{\mathbf{y}_{\leq t-1}}$, and chooses $\mathbf{y}_t \in \{0,1\}^d$ to be any value such that $\mathbf{y}_t \neq \widehat{\mathbf{y}}_t$. Because the tree is shattered, for every $t \in \mathbb{N}$ and every possible $\mathbf{y}_t \in \{0,1\}^d$ there exists $h \in \mathcal{H}$ that is consistent with $(\mathbf{x}, \mathbf{y})_{\leq t}$. Hence, the resulting sequence $\big((\mathbf{x}_t, \mathbf{y}_t)\big)_{t\in\mathbb{N}}$ is consistent with $\mathcal{H}$ while also satisfying $\mathbf{y}_t \neq \widehat{\mathbf{y}}_t$ for all $t \in \mathbb{N}$, and therefore the adversary wins the game. □

Notation 2.4.8. *Fix a function f as in Algorithm 2.1, and consider an execution of that algorithm using f in which the adversary plays the sequence $\big((\mathbf{x}_t, \mathbf{y}_t)\big)_{t\in\mathbb{N}}$. For each $t \in \mathbb{N}$ let $\xi^{(t)}$ denote the value of ξ at the beginning of time step t. We write*

$$\widehat{\mathbf{y}}_t : \ \mathcal{X}^d \to \{0,1\}^d$$

to denote the function given by

$$\widehat{\mathbf{y}}_t(x) = \widehat{\mathbf{y}}_{(\mathbf{x},\mathbf{y})_{\leq t-1}}(x) = f(\xi^{(t)} \circ x)$$

that determines the learner's choice at time t, such that $\widehat{\mathbf{y}}_t = \widehat{\mathbf{y}}_t(\mathbf{x}_t)$ for all $t \in \mathbb{N}$.

Definition 2.4.9. *Let $\mathcal{D} \in \Delta(\mathcal{X}^d \times \{0,1\}^d)$ be a distribution, and let $g : \mathcal{X}^d \to \{0,1\}^d$ be a function. The forbidden pattern loss of g with respect to $\mathcal{D}$ is*

$$\mathrm{L}_{\mathcal{D}}^{\mathsf{forbidden}}(g) = \mathbb{P}_{(X,Y)\sim\mathcal{D}}[g(X) = Y] = 1 - \mathrm{L}_{\mathcal{D}}^{0\text{-}1}(g).$$

The forbidden pattern loss simply captures the learners objective in the forbidden pattern game, which is to avoid having $\widehat{\mathbf{y}}_t = \mathbf{y}_t$.

Claim 2.4.10. *Assume $\mathsf{VCL}(\mathcal{H}) < \infty$. Let $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$ be a distribution, and let $S = \big((X_1,Y_1),(X_2,Y_2),\dots\big)$ be an infinite sequence of i.i.d. samples from $\mathcal{D}$. Consider an instance of the forbidden pattern game where the adversary plays the sequence S, and the learner plays according to the function $\widehat{\mathbf{y}}_t = \widehat{\mathbf{y}}_{S_{\leq t-1}}$ as in Notation 2.4.8. Then the forbidden pattern loss satisfies*

$$\lim_{t\to\infty} \mathop{\mathbb{P}}_{S\sim\mathcal{D}^{\mathbb{N}}}\Big[\mathrm{L}_{\mathcal{D}}^{\mathsf{forbidden}}(\widehat{\mathbf{y}}_t) > 0\Big] = 0.$$

Proof. By the proof of Lemma 2.4.3 and the assumption that $\mathsf{VCL}(\mathcal{H}) < \infty$, the strategy $\widehat{\mathbf{y}}_t$ is a winning strategy for the learner in the forbidden pattern game.

First, assume that S is consistent with $\mathcal{H}$. Then there exists a random variable $T \in \mathbb{N}$ that depends on S, such that

$$\mathbb{P}[\forall t \geq T: \ \widehat{\mathbf{y}}_t(X_t) \neq Y_t] = 1. \tag{2.8}$$

This is true because $\widehat{\mathbf{y}}_t$ is a winning strategy for the learner. Furthermore, by construction of the strategy $\widehat{\mathbf{y}}$, the function $\widehat{\mathbf{y}}_t(x)$ only changes if the learner made a mistake, namely

$$\mathbb{P}[\forall t,t' \geq T \ \forall x \in \mathcal{X}: \ \widehat{\mathbf{y}}_t(x) = \widehat{\mathbf{y}}_{t'}(x)] = 1. \tag{2.9}$$

Hence,

$$\begin{aligned}
&\lim_{t\to\infty} \mathop{\mathbb{P}}_{S\sim\mathcal{D}^{\mathbb{N}}}\Big[\mathrm{L}_{\mathcal{D}}^{\mathsf{forbidden}}(\widehat{\mathbf{y}}_t) = 0\Big] \\
&\qquad = \lim_{t\to\infty} \mathbb{P}\left[\left(\lim_{K\to\infty} \frac{1}{K}\sum_{k=1}^{K} \mathbb{1}\left(\widehat{\mathbf{y}}_t(X_{t+k}) = Y_{t+k}\right)\right) = 0\right] \\
&\qquad \geq \lim_{t\to\infty} \mathbb{P}\left[\left(\lim_{K\to\infty} \frac{1}{K}\sum_{k=1}^{K} \mathbb{1}\left(\widehat{\mathbf{y}}_t(X_{t+k})\right) = Y_{t+k}\right) = 0 \ \wedge \ t \geq T\right] \\
&\qquad = \lim_{t\to\infty} \mathbb{P}\left[\left(\lim_{K\to\infty} \frac{1}{K}\sum_{k=1}^{K} \mathbb{1}\left(\widehat{\mathbf{y}}_{t+k}(X_{t+k})\right) = Y_{t+k}\right) = 0 \ \wedge \ t \geq T\right] && \text{(By Eq. (2.9))} \\
&\qquad = \lim_{t\to\infty} \mathbb{P}[t \geq T] = 1. && \text{(By Eq. (2.8))}
\end{aligned}$$

So

$$\lim_{t\to\infty} \mathop{\mathbb{P}}_{S\sim\mathcal{D}^{\mathbb{N}}}\Big[\mathrm{L}_{\mathcal{D}}^{\mathsf{forbidden}}(\widehat{\mathbf{y}}_t) > 0\Big] = 1 - \lim_{t\to\infty} \mathop{\mathbb{P}}_{S\sim\mathcal{D}^{\mathbb{N}}}\Big[\mathrm{L}_{\mathcal{D}}^{\mathsf{forbidden}}(\widehat{\mathbf{y}}_t) = 0\Big] = 0$$

as desired.

It remains to show that $\mathbb{P}[S \text{ is consistent with } \mathcal{H}] = 1$. This is a consequence of the Borel–Cantelli lemma. Seeing as $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$, there exists a sequence $h_1, h_2, \ldots \in \mathcal{H}$ such that $\mathrm{L}_{\mathcal{D}}^{0\text{-}1}(h_k) \leq 2^{-k}$ for all $k \in \mathbb{N}$. For every $t, k \in \mathbb{N}$ let $G_{t,k} = \{\forall i \in [t]: \ h_k(X_i) = Y_i\}$ be the event in which $S_{\leq t}$ is consistent with h_k. Then for every $t \in \mathbb{N}$,

$$\sum_{k \in \mathbb{N}} \mathbb{P}_{S \sim \mathcal{D}^{\mathbb{N}}}[\neg G_{t,k}] \leq \sum_{k \in \mathbb{N}} t \cdot \mathrm{L}_{\mathcal{D}}^{0\text{-}1}(h_k) \leq t < \infty.$$

By Borel–Cantelli, this implies that

$$\forall t \in \mathbb{N}: \ \mathbb{P}_{S \sim \mathcal{D}^{\mathbb{N}}}[\exists k \in \mathbb{N}: \ G_{t,k}] = 1.$$

In words, for every $t \in \mathbb{N}$, with probability 1 over the choice of S, there exists $k \in \mathbb{N}$ such that h_k is consistent with $S_{\leq t}$. Finally,

$$\mathbb{P}_{S \sim \mathcal{D}^{\mathbb{N}}}[S \text{ is consistent with } \mathcal{H}] \geq \mathbb{P}_{S \sim \mathcal{D}^{\mathbb{N}}}\left[\bigcap_{t \in \mathbb{N}} \{\exists k \in \mathbb{N}: \ G_{t,k}\}\right] = 1,$$

because a countable intersection of probability 1 events has probability 1. □

Definition 2.4.11. *In the context of Claim 2.4.10, let*

$$t^* = t^*(\mathcal{D}) = \inf\left(\left\{t \in \mathbb{N}: \ \mathop{\mathbb{P}}_{S \sim \mathcal{D}^{\mathbb{N}}}\left[\mathrm{L}_{\mathcal{D}}^{\mathsf{forbidden}}(\widehat{\mathbf{y}}_t) > 0\right] \leq \frac{1}{8}\right\} \cup \{\infty\}\right).$$

The set of good sample sizes for $\mathcal{D}$ is

$$\mathcal{T}_{\mathcal{D}}^{\mathsf{good}} = \left\{t \in [t^*]: \ \mathop{\mathbb{P}}_{S \sim \mathcal{D}^{\mathbb{N}}}\left[\mathrm{L}_{\mathcal{D}}^{\mathsf{forbidden}}(\widehat{\mathbf{y}}_t) > 0\right] \leq \frac{1}{4}\right\}.$$

Claim 2.4.12. *There exists a function $\widehat{t}: (\mathcal{X} \times \{0,1\})^* \to \mathbb{N}$ as follows. Let $\mathcal{D} \in$ $\mathsf{Realizable}(\mathcal{H})$. There exist parameters $C, c \geq 0$ such that for any $n \in \mathbb{N}$,*

$$\mathbb{P}_{S \sim \mathcal{D}^n}\left[\widehat{t}(S) \in \mathcal{T}_{\mathcal{D}}^{\mathsf{good}}\right] \geq 1 - Ce^{-cn}.$$

Proof. Fix $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$. We show that Algorithm 2.2 satisfies the requirements of the claim. By Claim 2.4.10, $t^* = t^*(\mathcal{D})$ is finite and $\mathcal{T}_{\mathcal{D}}^{\mathsf{good}} \neq \varnothing$.

For each $t \in \mathbb{N}$ let $e_t = \mathbb{P}_{S \sim \mathcal{D}^{\mathbb{N}}}\left[\mathrm{L}_{\mathcal{D}}^{\mathsf{forbidden}}(\widehat{\mathbf{y}}_t) > 0\right]$. Hoeffding's inequality implies that there exist $C_t, c_t \geq 0$ such that $\mathbb{P}[|\widehat{e}_t - e_t| > 1/16] \leq C_t \cdot e^{-c_t \cdot n}$. By a union bound,

$$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^{\mathbb{N}}}[\exists t \in [t^*]: \ |\widehat{e}_t - e_t| > 1/16] \leq \sum_{t \in [t^*]} C_t \cdot e^{-c_t \cdot n} \leq C' \cdot e^{-c' \cdot n}, \tag{2.10}$$

for some suitable $C', c' \geq 0$.

Assumption:

- $S = \Big((X_1, Y_1), \ldots, (X_n, Y_n)\Big) \sim \mathcal{D}^n$ is a labeled training set.
- $\widehat{\mathbf{y}}_t = \widehat{\mathbf{y}}_{S_{\leq t-1}}$ is as in Notation 2.4.8.
- $m = \lfloor n/2 \rfloor$.

```
SampleSizeEstimator:
    $S^{\mathsf{train}}, S^{\mathsf{test}} \leftarrow$ independent disjoint subsets of $S$ of size $m$
    for  $t \in [m]$:
        $k \leftarrow \lfloor m/t \rfloor$
        $S^{\mathsf{train}}_1, \ldots, S^{\mathsf{train}}_k \leftarrow$ independent disjoint subsets of $S^{\mathsf{train}}$ of size $t$
        for  $i \in [k]$:
            $\widehat{e}_{t,i} \leftarrow \mathbb{1}\left(\exists (X, Y) \in S^{\mathsf{test}} :\ \widehat{\mathbf{y}}_{S_i^{\mathsf{train}}}(X) = Y\right)$
        $\widehat{e}_t \leftarrow \frac{1}{k} \sum_{i \in [k]} \widehat{e}_{t,i}$
    $\widehat{t} \leftarrow \inf\left(\{t \in [m] :\ \widehat{e}_t \leq {}^3/_{16}\} \cup \{\infty\}\right)$
    output $\widehat{t}$
```

Algorithm 2.2: An algorithm for finding $\widehat{t}$ such that with high probability, $\widehat{t} \in \mathcal{T}_{\mathcal{D}}^{\mathsf{good}}$.

Assume that $m \geq t^*$ and $\forall t \in [t^*] :\ |\widehat{e}_t - e_t| \leq {}^1/_{16}$. Then in particular, $\widehat{e}_{t^*} \leq e_{t^*} + {}^1/_{16} \leq {}^1/_8 + {}^1/_{16} = {}^3/_{16}$, and therefore the output $\widehat{t}$ selected by Algorithm 2.2 satisfies $\widehat{t} \leq t^*$. Additionally, the selected output satisfies $e_{\widehat{t}} \leq \widehat{e}_{\widehat{t}} + {}^1/_{16} \leq {}^3/_{16} + {}^1/_{16} = {}^1/_4$.

Combining the last paragraph with Eq. (2.10), we conclude that there exist $C, c \geq 0$ such that with probability at least $1 - Ce^{-cn}$, $\widehat{t}$ satisfies $\widehat{t} \leq t^*$ and $e_{t^*} \leq e_{\widehat{t}} \leq {}^1/_4$, so in particular $\widehat{t} \in \mathcal{T}_{\mathcal{D}}^{\mathsf{good}}$, as desired. □

Theorem 2.4.13 (Haussler et al., 1994, Theorem 2.3)**.** *Let $\mathcal{F} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class. There exists a function*

$$A :\ (\mathcal{X} \times \{0,1\})^* \times \mathcal{X} \to \{0,1\}$$

such that for any target function $f \in \mathcal{F}$, any $n \in \mathbb{N}$, and any $(x_1, \ldots, x_n) \in \mathcal{X}^n$, A satisfies

$$\frac{1}{|\mathrm{S}_n|} \sum_{\sigma \in \mathrm{S}_n} L_{\sigma,f}(A) \leq \frac{\mathsf{VC}(\mathcal{F})}{n},$$

where S_n is the set of all permutation functions $[n] \to [n]$, and $L_{\sigma,f}(A)$ is the 0-1 *loss of A with respect to f and the permutation σ, namely,*

$$L_{\sigma,f}(A) = \mathbb{1}\Big(A\Big(x_{\sigma(1)}, f(x_{\sigma(1)}), \ldots, x_{\sigma(n-1)}, f(x_{\sigma(n-1)}), x_{\sigma(n)}\Big) \neq f(x_{\sigma(n)})\Big).$$

Claim 2.4.14. *For any pattern avoidance function $g : \mathcal{X}^d \to \{0,1\}^d$ there exists a function A_g given by*

$$A_g : \ (\mathcal{X} \times \{0,1\})^* \to \{0,1\}^{\mathcal{X}}$$

such that for any distribution $\mathcal{D} \in \Delta\Big(\mathcal{X}^d \times \{0,1\}^d\Big)$ for which $\mathrm{L}_{\mathcal{D}}^{\mathsf{forbidden}}(g) = 0$ and for any $n \in \mathbb{N}$,

$$\mathbb{E}_{S \sim \mathcal{D}^n}\Big[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}(A_g(S))\Big] \leq \frac{d}{n}.$$

Proof. This follows from Theorem 2.4.13, along with an appropriate definition of a VC class from g.

Let

$$\mathcal{F} = \Big\{f \in \{0,1\}^{\mathcal{X}} : \ \Big(\forall \mathbf{x} \in \mathcal{X}^d : \ \Big(f(\mathbf{x}_1), \dots, f(\mathbf{x}_d)\Big) \neq g(\mathbf{x})\Big)\Big\}$$

be the set of all functions that avoid the pattern $g(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}^d$. Note that $\mathsf{VC}(\mathcal{F}) \leq d$ because there does not exist a shattered subset of $\mathcal{X}$ of cardinality d. Let A_g be the function A corresponding to $\mathcal{F}$ whose existence is guaranteed by Theorem 2.4.13. Then

$$\begin{aligned}
&\mathbb{E}_{S \sim \mathcal{D}^n}\Big[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}(A_g(S))\Big] \\
&\quad = \mathbb{E}_{S \sim \mathcal{D}^n, (X,Y) \sim \mathcal{D}}[\mathbb{1}\left((A_g(S))(X) \neq Y\right)] \\
&\quad = \mathbb{E}_{\Big((X_1,Y_1),\dots,(X_{n+1},Y_{n+1})\Big) \sim \mathcal{D}^{n+1}, \sigma \sim \mathrm{U}(\mathrm{S}_{n+1})}[L_{\sigma,f}(A_g)] && (2.11) \\
&\quad = \mathbb{E}_{\Big((X_1,Y_1),\dots,(X_{n+1},Y_{n+1})\Big) \sim \mathcal{D}^{n+1}}\left[\frac{1}{|\mathrm{S}_{n+1}|}\sum_{\sigma \in \mathrm{S}_{n+1}} L_{\sigma,f}(A_g)\right] \\
&\quad \leq \frac{\mathsf{VC}(\mathcal{F})}{n+1} \leq \frac{d}{n+1}. && \text{(By Theorem 2.4.13)}
\end{aligned}$$

In Eq. (2.11), f a function in $\mathcal{F}$ that is consistent with $(X_1, Y_1), \dots, (X_{n+1}, Y_{n+1})$. f is chosen deterministically as a function of $(X_1, Y_1), \dots, (X_{n+1}, Y_{n+1})$. Such an f exists because $\mathrm{L}_{\mathcal{D}}^{\mathsf{forbidden}}(g) = 0$, and $\mathcal{F}$ contains all functions that avoid g. In Eq. (2.11), we have used the fact that

$$\Big(S, (X,Y)\Big) \overset{d}{=} \Big((X_{\sigma(1)}, Y_{\sigma(1)}), \dots, (X_{\sigma(n+1)}, Y_{\sigma(n+1)})\Big). \qquad \square$$

Lemma 2.4.15. *If there exists a winning strategy for the leaner in the forbidden pattern game $\mathrm{G}_d^{\mathsf{forbidden}}(\mathcal{H})$, then $\mathcal{H}$ is learnable with rate d/n.*

Proof of Lemma 2.4.15. Let $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$. We need to show that there exist $C, c \geq 0$ as follows. For any $n \in \mathbb{N}$, let $\widehat{h}_S = \textsc{OptimalRateLearner}(S)$ with $S \sim \mathcal{D}^n$. Then

$$\mathbb{E}_{S \sim \mathcal{D}^n}\Big[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}\Big(\widehat{h}_S\Big)\Big] \leq d/n + Ce^{-cn}.$$

This is established via the following analysis of Algorithm 2.3. By Claim 2.4.12, there exist $C_0, c_0 \geq 0$ such that

$$\mathbb{P}_{S \sim \mathcal{D}^n}\Big[\widehat{t} \in \mathcal{T}_{\mathcal{D}}^{\mathsf{good}}\Big] \geq 1 - C_0 \cdot e^{-c_0 \cdot n}. \tag{2.12}$$

Assumptions:

- $n \in \mathbb{N}$, $m = \left\lfloor \frac{n}{2} \right\rfloor$.
- $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$.
- $S = \Big((X_1, Y_1), \dots, (X_n, Y_n)\Big) \sim \mathcal{D}^n$ is a labeled training set.
- x is the input that needs to be labeled.
- $\widehat{t}$ is a function as in Claim 2.4.12.
- For any sequence $\mathbf{z} \in \left(\mathcal{X}^d \times \{0,1\}^d\right)^*$, $\widehat{\mathbf{y}}_{\mathbf{z}} : \mathcal{X}^d \to \{0,1\}^d$ is a pattern avoidance function as in Notation 2.4.8.
- A_g is a learning algorithm that uses pattern avoidance function g, as in Claim 2.4.14.

```
OptimalRateLearner($S$):
    $\widehat{t} \leftarrow \widehat{t}(S)$
    $k \leftarrow \lfloor m/\widehat{t} \rfloor$
    $S_g, S_a \leftarrow$ partition of $S$ into two disjoint sets of size at least $m$
    $S_1, \dots, S_k \leftarrow$ partition of $S_g$ into $k$ disjoint sets of size at least $\widehat{t}$
    for  $i \in [k]$:
        $g_i \leftarrow \widehat{\mathbf{y}}_{S_i}$
        $a_i \leftarrow A_{g_i}(S_a)$
    $\widehat{h} \leftarrow \Big(x \mapsto \mathsf{Majority}(a_1(x), \dots, a_k(x))\Big)$        ▷ Defining a function $\widehat{h} : \mathcal{X} \to \{0,1\}$
    output $\widehat{h}$
```

Algorithm 2.3: An algorithm that achieves the optimal learning rate for any class with finite VCL dimension.

From the definition of $\mathcal{T}_{\mathcal{D}}^{\mathsf{good}}$, if $\widehat{t} \in \mathcal{T}_{\mathcal{D}}^{\mathsf{good}}$ then for every $i \in [k]$,

$$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^n}\left[\mathrm{L}_{\mathcal{D}}^{\mathsf{forbidden}}(g_i) > 0\right] \leq \frac{1}{4}.$$

By Hoeffding's inequality, there exist $C_1, c_1 \geq 0$ such that

$$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^n}\left[\frac{\left|\left\{i \in [k] : \ \mathrm{L}_{\mathcal{D}}^{\mathsf{forbidden}}(g_i) > 0\right\}\right|}{k} \geq \frac{3}{8} \;\middle|\; \widehat{t} \in \mathcal{T}_{\mathcal{D}}^{\mathsf{good}}\right] \leq C_1 \cdot e^{-c_1 \cdot n}, \tag{2.13}$$

where we have used the fact that if $\widehat{t} \in \mathcal{T}_{\mathcal{D}}^{\mathsf{good}}$ then $k = \Omega(n)$. Applying the inequality $\mathbb{P}[E] \leq \mathbb{P}[E|F] + \mathbb{P}[\neg F]$ to Eqs. (2.12) and (2.13) implies that there exist $C, c \geq 0$ such that

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^n}\left[\frac{\left|\left\{i \in [k] : \ \mathrm{L}_{\mathcal{D}}^{\mathsf{forbidden}}(g_i) > 0\right\}\right|}{k} \geq \frac{3}{8}\right] \leq Ce^{-cn}, \tag{2.14}$$

From Claim 2.4.14, for any $i \in [k]$, if $\mathrm{L}_{\mathcal{D}}^{\mathsf{forbidden}}(g_i) = 0$ then

$$\mathbb{E}_{S\sim\mathcal{D}^n}\left[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}(A_{g_i}(S))\right] \leq \frac{d}{n}. \tag{2.15}$$

Let B be the bad event whose probability is bounded by Eq. (2.14). Then

$$\begin{aligned}\mathbb{E}_{S\sim\mathcal{D}^n}\left[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}\left(\widehat{h}_S\right)\right] &= \mathbb{E}_{S\sim\mathcal{D}^n}\left[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}\left(\widehat{h}_S\right)\cdot \mathbb{1}(B)\right] + \mathbb{E}_{S\sim\mathcal{D}^n}\left[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}\left(\widehat{h}_S\right)\cdot \mathbb{1}(\neg B)\right]\\ &\leq \mathbb{P}_{S\sim\mathcal{D}^n}[B] + \mathbb{E}_{S\sim\mathcal{D}^n}\left[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}\left(\widehat{h}_S\right)\cdot \mathbb{1}(\neg B)\right]\\ &\leq Ce^{-cn} + \mathbb{E}_{S\sim\mathcal{D}^n}\left[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}\left(\widehat{h}_S\right)\cdot \mathbb{1}(\neg B)\right], \end{aligned}\tag{2.16}$$

where the final inequality follows by Eq. (2.14).

The expectation in the previous line can be bounded by

$$\begin{aligned}&\mathbb{E}_{S\sim\mathcal{D}^n}\left[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}\left(\widehat{h}_S\right)\cdot \mathbb{1}(\neg B)\right]\\ &\quad= \mathbb{E}_{S\sim\mathcal{D}^n,(X,Y)\sim\mathcal{D}}[\mathbb{1}\left(\mathsf{Majority}(a_1(X),\ldots,a_k(X)) \neq Y\right)\mathbb{1}(\neg B)]\\ &\quad\leq \mathbb{P}_{S\sim\mathcal{D}^n,(X,Y)\sim\mathcal{D}}[\mathsf{Majority}(a_1(X),\ldots,a_k(X)) \neq Y \ \wedge \ \neg B]\\ &\quad\leq \mathbb{P}\left[\left(\frac{|\{i: \ a_i(X)\neq Y\}|}{k} \geq \frac{1}{2}\right) \wedge \left(\frac{|\{i: \ \mathrm{L}_{\mathcal{D}}^{\mathsf{forbidden}}(g_i) = 0\}|}{k} \geq \frac{5}{8}\right)\right]\\ &\quad\leq \mathbb{P}\left[\left(\frac{|\{i: \ (a_i(X)\neq Y) \wedge (\mathrm{L}_{\mathcal{D}}^{\mathsf{forbidden}}(g_i) = 0)\}|}{k} \geq \frac{1}{8}\right)\right]\\ &\quad\leq \frac{8}{k}\cdot \mathbb{E}\left[|\{i: \ (a_i(X)\neq Y) \wedge (\mathrm{L}_{\mathcal{D}}^{\mathsf{forbidden}}(g_i) = 0)\}|\right] &&\text{(Markov's inequality)}\\ &\quad= \frac{8}{k}\cdot \sum_{i\in[k]} \mathbb{P}\left[(a_i(X)\neq Y) \wedge (\mathrm{L}_{\mathcal{D}}^{\mathsf{forbidden}}(g_i) = 0)\right]\\ &\quad\leq \frac{8}{k}\cdot \sum_{i\in[k]} \frac{d}{n} = \frac{8d}{n}, &&(2.17)\end{aligned}$$

where the last inequality follows from Eq. (2.15).

Finally, plugging the bound of Eq. (Markov's inequality) into Eq. (2.16) yields

$$\mathbb{E}_{S\sim\mathcal{D}^n}\left[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}\left(\widehat{h}_S\right)\right] \leq Ce^{-cn} + \frac{8d}{n},$$

as desired. □

Lower Bound

Lemma 2.4.16. *For any set $\mathcal{X}$ and any hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ satisfying $d = \mathsf{VCL}(\mathcal{H})$ with $1 \leq d < \infty$, there exists a distribution $\mathcal{D}_{\mathcal{X}} \in \Delta(\mathcal{X})$ such that for any (possibly randomized) learning algorithm $\widehat{h}$ there exists $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$ such that the marginal distribution of $\mathcal{D}$ on $\mathcal{X}$ is $\mathcal{D}_{\mathcal{X}}$, and the inequality*

$$\mathbb{E}_{S\sim\mathcal{D}^n}\left[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}\left(\widehat{h}_S\right)\right] \geq \frac{d}{100\cdot n} \tag{2.18}$$

holds for infinitely many $n \in \mathbb{N}$.

Ingredients

The proof employs a claim about *indifferent* d-VCL trees, which is proved using a simple lemma from Ramsey theory.

Notation 2.4.17. *For any $\mathbf{u} \in \left(\{0,1\}^d\right)^*$, let $\mathsf{index}(\mathbf{u}) \in \mathbb{N}$ denote the index of $\mathbf{u}$ in the lexicographical ordering of $\left(\{0,1\}^d\right)^*$.*

Definition 2.4.18. *Let $d \in \mathbb{N}$, let $\mathcal{X}$ be a set, let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class, and let*

$$T = \left\{\mathbf{x}_{\mathbf{u}} \in \mathcal{X}^d : \ \mathbf{u} \in \left(\{0,1\}^d\right)^*\right\}$$

be an infinite d-VCL tree that is shattered by $\mathcal{H}$. Recall that this implies the existence of a collection

$$\mathcal{H}_T = \left\{h_{\mathbf{u}} \in \mathcal{H} : \ \mathbf{u} \in \left(\{0,1\}^d\right)^*\right\}$$

of consistent functions, namely, for each $\mathbf{u} \in \left(\{0,1\}^d\right)^*$, $h_{\mathbf{u}}$ is consistent with the path from the root to node $\mathbf{u}$, as in the definition of shattering a VCL tree (Definition 2.2.11).

We say that such a collection $\mathcal{H}_T$ is <u>indifferent</u> if for every $\mathbf{v}, \mathbf{u}, \mathbf{w} \in \left(\{0,1\}^d\right)^*$, if $\mathsf{index}(\mathbf{v}) < \mathsf{index}(\mathbf{u})$, and $\mathbf{w}$ is a descendant of $\mathbf{u}$ in the tree T, then $h_{\mathbf{u}}(\mathbf{x}_{\mathbf{v}}^j) = h_{\mathbf{w}}(\mathbf{x}_{\mathbf{v}}^j)$ for every $j \in [d]$. In words, the functions for all the descendants of a node that appears after $\mathbf{v}$ agree on $\mathbf{v}$.

We say that T is <u>indifferent</u> if it has a set $\mathcal{H}_T$ of consistent functions that are indifferent.

Intuitively, if T is indifferent, then the labels for a node $\mathbf{v}$ provide no information on the labels of a node $\mathbf{u}$ that appears after $\mathbf{v}$ in the lexicographical order.

The claim about indifferent d-VCL trees is as follows.

Claim 2.4.19. *Let $d \in \mathbb{N}$, let $\mathcal{X}$ be a set, let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class, and let T be an infinite d-VCL tree that is shattered by $\mathcal{H}$. Then there exists an infinite d-VCL tree T' that is shattered by $\mathcal{H}$ that is indifferent.*

Following is the lemma from Ramsey theory used for proving Claim 2.4.19, and a generalized notion of a trees and subtrees used in that lemma.

Definition 2.4.20. *Let $(X, \preceq)$ be a partial order relation. For $a, b \in X$, we say that b is a child of a if $a \preceq b$ and there does not exist $c \in X$ such that $a \preceq c \preceq b$. For $k \in \mathbb{N}$, we say that $(X, \preceq)$ is an infinite k-ary tree if every $a \in X$ has precisely k distinct children. We say that a partial order $(X', \preceq')$ is a subtree of $(X, \preceq)$ if $X' \subseteq X$, and $\forall a, b \in X'$: $a \preceq' b \iff a \preceq b$.*

Lemma 2.4.21. *Let $T = (X, \preceq)$ be an infinite k-ary tree, and let $g : X \to \{0,1\}$ be a two-coloring of T. Then T has a monochromatic infinite k-ary subtree $T' = (X', \preceq')$, namely there exists T' such that T' is a subtree of T, T' is an infinite k-ary tree, and $|g(X')| = |\{g(a) : a \in X'\}| = 1$.*

Proof of Lemma 2.4.21. If there exists $a \in X$ such that the set X' consisting of a and all its descendants satisfies $g(X') = \{1\}$, then we are done (take T' to be the subtree consisting of a and all its descendants). Otherwise, every $a \in X$ has a descendant $b \in X$ such that $g(b) = 0$. This implies that one can construct an infinite k-ary subtree that is 0-monochromatic using the following recursive procedure. Let r be any member of X such that $g(r) = 0$. Let T' be an empty tree, and add r to T'. Subsequently, for each node n added to T' (including r), for each child a of n, add to T' an arbitrary descendant b of a such that $g(b) = 0$. □

Proof of Claim 2.4.19. First, observe that if $T = \left\{x_{\mathbf{u}} : \mathbf{u} \in \left(\{0,1\}^d\right)^*\right\}$ is an infinite d-VCL tree that is shattered by $\mathcal{H}$ with a collection $\{h_{\mathbf{u}} : \mathbf{u} \in \left(\{0,1\}^d\right)^*\}$ of consistent functions, then for any $x \in \mathcal{X}$ there exists an infinite d-VCL tree that is shattered by $\mathcal{H}$ that is a subtree of T and has a collection of consistent functions that agree on x. Indeed, this follows from Lemma 2.4.21 by choosing a two-coloring $g : \left(\{0,1\}^d\right)^* \to \{0,1\}$ of T given by $g(\mathbf{u}) = h_{\mathbf{u}}(x)$.

Second, we use the above observation to construct an infinite d-VCL tree

$$T' = \left\{x'_{\mathbf{u}} : \mathbf{u} \in \left(\{0,1\}^d\right)^*\right\}$$

that is shattered by $\mathcal{H}$ and is indifferent. The construction works by starting with $T' := T$ and then repeatedly modifying T', as specified in Algorithm 2.4. Each modification step replaces a subtree $T'_{\mathbf{u}}$ of T' with one of its own infinite d-VCL subtrees, which is obtained by invoking the above observation on $T'_{\mathbf{u}}$ and $x = \mathbf{x}^j_{\mathbf{v}}$ for some $j \in [d]$ and some $\mathbf{v}$ that precedes $\mathbf{u}$. In each step, the set of nodes of T' decreases (is replaced by one of its subsets), and the collection of consistent functions can be decreased in a corresponding manner (be replaced by a subset of itself that corresponds to the new set of nodes).

```
$T' \leftarrow T$
for $\mathbf{u} \in \left(\{0,1\}^d\right)^*$ in lexicographic order:
    for $\mathbf{v} \in \left(\{0,1\}^d\right)^*$ such that $\mathsf{index}(\mathbf{v}) < \mathsf{index}(\mathbf{u})$:
        for $j \in [d]$:
            replace $T'_{\mathbf{u}}$ with an infinite $2^d$-ary subtree of $T'_{\mathbf{u}}$ that has
                a collection of consistent functions that agree on $\mathbf{x}^j_{\mathbf{v}}$
```

Algorithm 2.4: Construction of an indifferent d-VCL tree. ($T'_{\mathbf{u}}$ denotes the infinite 2^d-ary subtree of T' rooted at node $\mathbf{u}$.)

Algorithm 2.4 never terminates, but it defines an infinite d-VCL tree T'. T' is well-defined because for every $\mathbf{r} \in \left(\{0,1\}^d\right)^*$, the value of $\mathbf{x}'_{\mathbf{r}}$ never changes after the outer loop advances past $\mathbf{r}$ (i.e., $\mathsf{index}(\mathbf{u}) > \mathsf{index}(\mathbf{r})$), and so $\mathbf{x}'_{\mathbf{r}}$ is eventually fixed. T' is an infinite 2^d-ary subtree of T (each replacement maintains that T' is an infinite 2^d-ary subtree of T, so the resulting tree defined by this process is also an infinite 2^d-ary subtree of T). This implies that it is a d-VCL tree that is shattered by $\mathcal{H}$. T' is indifferent by construction, because for each $\mathbf{q}, \mathbf{r}, \setminus \in \left(\{0,1\}^d\right)^*$ and $k \in [d]$, if $\mathsf{index}(\mathbf{q}) < \mathsf{index}(\mathbf{r})$, and $\setminus$ is a descendant of $\mathbf{r}$, then during the iteration of the innermost loop in which $\mathbf{u} = \mathbf{r}$, $\mathbf{v} = \mathbf{q}$, and $j = k$, the subtree $T'_{\mathbf{r}}$ was replaced with a subtree that has a collection of consistent functions that agree on $(\mathbf{x}'_{\mathbf{q}})^k$. In particular this implies that $h_{\mathbf{r}}((\mathbf{x}'_{\mathbf{q}})^k) = h_{\setminus}((\mathbf{x}'_{\mathbf{q}})^k)$. This agreement continues to hold from that point onwards, because the collection of consistent functions for descendants of $h_{\mathbf{r}}$ can only decrease at each step. □

When a tree is indifferent, it admits a notion of *branch functions*, as follows.

Notation 2.4.22. $\mathcal{Y} = \left(\{0,1\}^d\right)^{\mathbb{N}}$.

Definition 2.4.23. *Let $d \in \mathbb{N}$, let $\mathcal{X}$ be a set, let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class, and let*

$$T = \left\{\mathbf{x}_{\mathbf{u}} \in \mathcal{X}^d : \ \mathbf{u} \in \left(\{0,1\}^d\right)^*\right\}$$

be an infinite d-VCL tree that is shattered by $\mathcal{H}$ with a collection

$$\mathcal{H}_T = \left\{h_{\mathbf{u}} \in \mathcal{H} : \ \mathbf{u} \in \left(\{0,1\}^d\right)^*\right\}$$

of consistent functions that are indifferent. Let

$$\mathcal{X}_T = \{\mathbf{x}^i_{\mathbf{u}} : \ \mathbf{u} \in \left(\{0,1\}^d\right)^* \ \wedge \ i \in [d]\}.$$

For every $\mathbf{y} \in \mathcal{Y}$, the branch function for $\mathbf{y}$ is the unique function $f_{\mathbf{y}} : \ \mathcal{X}_T \to \{0,1\}$ such that for each $\mathbf{v} \in \left(\{0,1\}^d\right)^*$ and $j \in [d]$,

$$f_{\mathbf{y}}(\mathbf{x}^j_{\mathbf{v}}) = h_{\mathbf{u}}(\mathbf{x}^j_{\mathbf{v}})$$

for a node $\mathbf{u}$ *such that* $\mathbf{y}_{\leq|\mathbf{u}|} = \mathbf{u}$ *and* $\mathsf{index}(\mathbf{u}) > \mathsf{index}(\mathbf{v})$. *In words,* $f_{\mathbf{y}}(\mathbf{x}_{\mathbf{v}}^{j})$ *is the value assigned to* $\mathbf{x}_{\mathbf{v}}^{j}$ *by the consistent function of any node on the infinite branch* $\mathbf{y}$ *that appears after* $\mathbf{v}$ *in lexicographic order. (Due to the indifference property,* $h_{\mathbf{u}}(\mathbf{x}_{\mathbf{v}}^{j})$ *is the same for any such node* $\mathbf{u}$*.)*

We note some consequences of the definitions of indifference and branch functions.

Claim 2.4.24. *Let* T *be an indifferent infinite* d*-VCL tree with a collection of branch functions* $\{f_{\mathbf{y}}\}_{\mathbf{y}\in\mathcal{Y}}$*. Then:*

1. *Every branch function* $f_{\mathbf{y}}$ *is* finitely realizable*, meaning that for any finite set* $\{x_1, \dots x_m\} \subseteq \mathcal{X}_T$*, there exists a function* $h \in \mathcal{H}$ *such that for all* $i \in [m]$*,* $f_{\mathbf{y}}(x_i) = h(x_i)$*.*

2. *Each element in* T *is unique. Namely, for every* $\mathbf{u}, \mathbf{v} \in \left(\{0,1\}^d\right)^*$ *and every* $i, j \in [d]$*, if* $\mathbf{u} \neq \mathbf{v}$ *or* $i \neq j$ *then* $\mathbf{x}_{\mathbf{u}}^{i} \neq \mathbf{x}_{\mathbf{v}}^{j}$*.*

3. *Let* $\mathbf{v}, \mathbf{u} \in \left(\{0,1\}^d\right)^*$*. If* $\mathsf{index}(\mathbf{u}) > \mathsf{index}(\mathbf{v})$ *then there exists* $\mathbf{b} \in \{0,1\}^d$ *such that for any* $\mathbf{y} \in \mathcal{Y}$*, if* $\mathbf{u} = \mathbf{y}_{\leq|\mathbf{u}|}$ *then* $f_{\mathbf{y}}(\mathbf{x}_{\mathbf{v}}^{j}) = \mathbf{b}_j$ *for all* $j \in [d]$*. In words, if* $\mathbf{v}$ *precedes* $\mathbf{u}$ *in lexicographical order, then all the branch functions for branches that pass through node* $\mathbf{u}$ *agree on node* $\mathbf{v}$*.*

We think of Item 3 as an indifference property for branch functions. Intuitively, it means that knowing the labels for $\mathbf{v}$ does not provide any information on which of the branch functions for branches that pass through $\mathbf{u}$ is more likely to be the correct labeling function. The branch functions that pass through $\mathbf{u}$ are *indifferent* to the labels of $\mathbf{v}$.

Proof of Claim 2.4.24. Item 1 is immediate from the definition of $f_{\mathbf{y}}$. For Item 2, clearly if $\mathbf{u} = \mathbf{v}$ then $\mathbf{x}_{\mathbf{u}}^{i} \neq \mathbf{x}_{\mathbf{v}}^{j}$, since otherwise node $\mathbf{u}$ could not have 2^d children, in contradiction to T being a d-VCL tree. Assume for contradiction that $\mathsf{index}(\mathbf{v}) < \mathsf{index}(\mathbf{u})$ and $\mathbf{x}_{\mathbf{u}}^{i} = \mathbf{x}_{\mathbf{v}}^{j}$. Then all consistent functions for the children of $\mathbf{u}$ must agree on $\mathbf{x}_{\mathbf{v}}^{j}$, but that implies that they agree on $\mathbf{x}_{\mathbf{u}}^{i}$ as well, which is again a contradiction to $\mathbf{u}$ having 2^d children. Finally, Item 3 is immediate from the definition of $f_{\mathbf{y}}$ and from the indifference of T. □

The proof of the lower bound also employs the reverse Fatou's lemma.

Lemma 2.4.25 (Reverse Fatou; e.g., Theorem 10.17 in Browder 1996, and ProofWiki)**.** *Let* $(\Omega, \mathcal{F}, \mu)$ *be a measure space. Let* $g : \Omega \to \mathbb{R}$ *be a non-negative measurable function such that* $\int_{\Omega} g \, d\mu < \infty$*. For each* $n \in \mathbb{N}$ *let* $f_n : \Omega \to \mathbb{R}$ *be a measurable function such that* $\forall \omega \in \Omega : f_n(\omega) \leq g(\omega)$*. Then*

$$\int_{\Omega} \limsup_{n\to\infty} f_n \, d\mu \geq \limsup_{n\to\infty} \int_{\Omega} f_n \, d\mu.$$

Proof of Lower Bound

Proof of Lemma 2.4.16. We will define a set of distributions

$$\{P_{\mathbf{y}}\}_{\mathbf{y}\in\mathcal{Y}} \subseteq \mathsf{Realizable}(\mathcal{H})$$

that depends on $\mathcal{H}$ such that all the distributions in the set have the same marginal distribution over $\mathcal{X}$. The proof uses the probabilistic method to show that for every learning algorithm $\widehat{h}$ for $\mathcal{H}$ there exists $\mathbf{y}^* \in \mathcal{Y}$ (that depends on $\widehat{h}$) such that $P_{\mathbf{y}^*}$ is a hard distribution for $\widehat{h}$, namely, that Eq. (2.18) holds for $\mathcal{D} = P_{\mathbf{y}^*}$ for infinitely many values of n.

The set $\{P_{\mathbf{y}}\}_{\mathbf{y}\in\mathcal{Y}}$ is defined as follows. By Claim 2.4.19 and the assumption that $\mathsf{VCL}(\mathcal{H}) = d$, there exist an indifferent infinite d-VCL tree

$$T = \left\{\mathbf{x}_{\mathbf{u}} \in \mathcal{X}^d : \ \mathbf{u} \in \left(\{0,1\}^d\right)^*\right\}$$

with a corresponding collection of branch functions

$$\mathcal{F} = \left\{f_{\mathbf{y}} \in \{0,1\}^{\mathcal{X}_T} : \ \mathbf{y} \in \mathcal{Y}\right\}.$$

Fix such a pair $(T, \mathcal{F})$. For each $\mathbf{y} \in \mathcal{Y}$ let

$$P_{\mathbf{y}}\Big((x,y)\Big) = \sum_{\mathbf{u}\in\left(\{0,1\}^d\right)^*} (d-1)d^{-\mathsf{index}(\mathbf{u})-1} \sum_{i=1}^{d} \mathbb{1}\left(x = \mathbf{x}_{\mathbf{u}}^i \ \wedge \ y = f_{\mathbf{y}}\Big(\mathbf{x}_{\mathbf{u}}^i\Big)\right).$$

In words, $P_{\mathbf{y}}$ corresponds to the following sampling procedure:

1. Sample an index $k \in \mathbb{N}$ such that $\forall s \in \mathbb{N}: \ \mathbb{P}[k = s] = (d-1)d^{-s}$.[5]
2. Let $\mathbf{u} \in \left(\{0,1\}^d\right)^*$ be the k-th string in the lexicographical ordering of $\left(\{0,1\}^d\right)^*$.
3. Sample $j \in [d]$ independently and uniformly at random.
4. Output $(\mathbf{x}_{\mathbf{u}}^j, f_{\mathbf{y}}(\mathbf{x}_{\mathbf{u}}^j))$.

Note that the marginal distribution of $P_{\mathbf{y}}$ on $\mathcal{X}$ (the distribution of $\mathbf{x}_{\mathbf{u}}^i$) is the same for all $\mathbf{y} \in \mathcal{Y}$; this is the marginal distribution $\mathcal{D}_{\mathcal{X}}$ mentioned in the statement.

To see that $P_{\mathbf{y}}$ is realizable, note that for every $\varepsilon > 0$ there exists $k_\varepsilon \in \mathbb{N}$ such that in Step 1 of the sampling procedure, $\mathbb{P}[k > k_\varepsilon] \leq \varepsilon$. $f_{\mathbf{y}}$ is finitely-realizable by $\mathcal{H}$ (Item 1 in Claim 2.4.24), so in particular there exists $h_\varepsilon \in \mathcal{H}$ that is consistent with $Z_\varepsilon = \left\{(\mathbf{x}_{\mathbf{u}}^j, f_{\mathbf{y}}(\mathbf{x}_{\mathbf{u}}^j)) : \ \mathsf{index}(\mathbf{u}) \leq k_\varepsilon \ \wedge \ j \in [d]\right\}$. Hence, $\mathrm{L}_{P_{\mathbf{y}}}^{0\text{-}1}(h_\varepsilon) \leq \mathbb{P}_{(x,y)\sim P_{\mathbf{y}}}[(x,y) \notin Z_\varepsilon] \leq \mathbb{P}[k > k_\varepsilon] \leq \varepsilon$.

For a fixed algorithm $\widehat{h}$ and for each $n \in \mathbb{N}$, consider the following experiment:

[5] Recall that for a geometric series, $\sum_{s=1}^{\infty} d^{-s} = \frac{1}{d-1}$ when $d > 1$, and therefore $\sum_{s=1}^{\infty} \mathbb{P}[k = s] = \sum_{s=1}^{\infty} (d-1)d^{-s} = 1$.

- A value $\mathbf{y} \in \mathcal{Y}$ is sampled from the uniform distribution $\mathrm{U}(\mathcal{Y})$, namely each bit in $\mathbf{y}$ is chosen independently and uniformly at random.
- An i.i.d. training set $S = \big((X_1, Y_1, K_1), (X_2, Y_2, K_2), \ldots, (X_n, Y_n, K_n)\big) \sim P_{\mathbf{y}}^n$ is generated according to the sampling procedure of Steps 1 to 4, where for each $i \in [n]$, $K_i \in \mathbb{N}$ is the index selected at Step 1, and (X_i, Y_i) is the output at Step 4.
- An additional test sample $(X, Y, K) \sim P_{\mathbf{y}}$ is generated in the same manner.
- A randomness value ρ is sampled for the algorithm $\widehat{h}$, and then $\widehat{h}$ is executed with training set S and randomness ρ and produces a hypothesis $\widehat{h}_S$.
- $\widehat{h}_S$ is used to predict a label $\widehat{h}_S(X)$ for X.

This experiment defines a joint distribution

$$(\mathbf{y}, S, X, Y, K, \rho) \tag{2.19}$$

that is used throughout the remainder of the proof.

For any $\kappa \in \mathbb{N}$, let $G(\kappa)$ denote the event in which the following conditions hold:

- $K = \kappa \geq \max\{K_1, \ldots, K_n\}$.
- $|\{i \in [n] : K_i = \kappa\}| < d/2$.
- $X \notin \{X_i : \ i \in [n]\}$.

We make two observations concerning $G(\kappa)$. The first observation is that

$$\mathbb{P}[G(\kappa)] \geq (d-1)d^{-\kappa}/4 \tag{2.20}$$

when $n = n_\kappa = \left\lfloor \frac{d^{\kappa+1}}{8(d-1)} \right\rfloor$. To see this, let

$$C_{=\kappa} = \left|\{i \in [n_\kappa] : \ K_i = \kappa\}\right|, \quad C_{>\kappa} = \left|\{i \in [n_\kappa] : \ K_i > \kappa\}\right|.$$

Then

$$\mathbb{E}[C_{=\kappa}] = n_\kappa \cdot (d-1)d^{-\kappa} \leq \frac{d^{\kappa+1}}{8(d-1)} \cdot (d-1)d^{-\kappa} = \frac{d}{8},$$

and

$$\begin{aligned}\mathbb{E}[C_{>\kappa}] &= n_\kappa \cdot \sum_{s=\kappa+1}^{\infty} (d-1)d^{-s} \\ &\leq \frac{d^{\kappa+1}}{8(d-1)} \cdot 2(d-1)d^{-\kappa-1} = \frac{1}{4}.\end{aligned}$$

By Markov's inequality,

$$\mathbb{P}\left[C_{=\kappa} \geq \frac{d}{2}\right] \leq \frac{1}{4}, \quad \text{and} \quad \mathbb{P}[C_{>\kappa} \geq 1] \leq \frac{1}{4}.$$

By a union bound,

$$\mathbb{P}\left[C_{=\kappa} < \frac{d}{2} \ \wedge \ C_{>\kappa} = 0\right] \geq \frac{1}{2}. \tag{2.21}$$

Hence, for any $\boldsymbol{\eta} \in \mathcal{Y}$,

$$\begin{aligned}
&\mathbb{P}_{\mathbf{y},S,X,Y,K}[G(\kappa) \mid \mathbf{y} = \boldsymbol{\eta}] \\
&\quad = \mathbb{P}[K = \kappa \mid \mathbf{y} = \boldsymbol{\eta}] \cdot \mathbb{P}\left[C_{=\kappa} < \frac{d}{2} \ \wedge \ C_{>\kappa} = 0 \mid \mathbf{y} = \boldsymbol{\eta}\right] \\
&\qquad \cdot \mathbb{P}\left[X \notin \{X_i : \ i \in [n_\kappa]\} \ \middle| \ C_{=\kappa} < \frac{d}{2} \ \wedge \ C_{>\kappa} = 0 \ \wedge \ K = \kappa \ \wedge \ \mathbf{y} = \boldsymbol{\eta}\right] \\
&\quad \geq (d-1)d^{-\kappa} \cdot \frac{1}{2} \\
&\qquad \cdot \mathbb{P}\left[X \notin \{X_i : \ i \in [n_\kappa]\} \ \middle| \ C_{=\kappa} < \frac{d}{2} \ \wedge \ C_{>\kappa} = 0 \ \wedge \ K = \kappa \ \wedge \ \mathbf{y} = \boldsymbol{\eta}\right] \\
&\hspace{20em} \text{(By Eq. (2.21), } (C_{=\kappa}, C_{>\kappa}) \perp \mathbf{y}\text{)} \\
&\quad \geq (d-1)d^{-\kappa} \cdot \frac{1}{2} \cdot \frac{1}{2}.
\end{aligned}$$

For the last inequality, recall that the elements in T are unique (Item 2 in Claim 2.4.24). Consequently, for every $i \in [n_\kappa]$, if $K_i < \kappa = K$ then $X_i \neq X$. The conditions $C_{=\kappa} < d/2$ and $K = \kappa$, and the sampling of $j \sim \mathrm{U}([d])$ in Step 3 imply that with probability at least 1/2, $X \notin \{X_i : \ i \in [n_\kappa] \ \wedge \ K_i = \kappa\}$. This establishes Eq. (2.20), which is our first observation about $G(\kappa)$.

Our second observation is that for any κ corresponding to a node on the branch $\mathbf{y}$, if $G(\kappa)$ occurs then $\widehat{h}$ makes an incorrect prediction with probability 1/2.

Formally, for any $t \in \mathbb{N}$, let $\kappa_{\mathbf{y},t} = \mathsf{index}(\mathbf{y}_{<t})$, where $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \dots)$ and $\mathbf{y}_{<t} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{t-1})$. In words, $\kappa_{\mathbf{y},t}$ is the index in the lexicographic ordering of $(\{0,1\}^d)^*$ corresponding to the t-th node in the branch $\mathbf{y}$. Let $n_{\mathbf{y},t} = n_{\kappa_{\mathbf{y},t}}$. The second observation states that for any $t \in \mathbb{N}$,

$$\begin{aligned}
&\mathbb{E}_{\mathbf{y}\sim \mathrm{U}(\mathcal{Y})}\left[\mathbb{P}_{S\sim P_{\mathbf{y}}^{n_{\mathbf{y},t}},(X,Y,K)\sim P_{\mathbf{y},\rho}}\left[\widehat{h}_S(X) \neq Y \mid G(\kappa_{\mathbf{y},t})\right]\right] \\
&\qquad = \mathbb{P}_{\mathbf{y}\sim \mathrm{U}(\mathcal{Y}),S\sim P_{\mathbf{y}}^{n_{\mathbf{y},t}},(X,Y,K)\sim P_{\mathbf{y},\rho}}\left[\widehat{h}_S(X) \neq Y \mid G(\kappa_{\mathbf{y},t})\right] = \frac{1}{2}.
\end{aligned} \tag{2.22}$$

This probability pertains to the special case of the experiment of Eq. (2.19) in which the number n of samples in S depends on $\mathbf{y}$, satisfying $n = n_{\mathbf{y},t}$. It is a conditional probability given that $G(\kappa_{\mathbf{y},t})$ occurred, where $G(\kappa_{\mathbf{y},t})$ is an event involving $(\mathbf{y}, X, X_1, \ldots, X_n, K, K_1, \ldots, K_n)$. To establish Eq. (2.22), it suffices to show that for any $t \in \mathbb{N}$,

$$\mathbb{P}_{\mathbf{y}\sim \mathrm{U}(\mathcal{Y}), S\sim P_{\mathbf{y}}^{n_{\mathbf{y},t}}, (X,Y,K)\sim P_{\mathbf{y},\rho}}\Big[Y = 1 \mid X, \{X_i, Y_i\}_{i\in[n_{\mathbf{y},t}]}, G(\kappa_{\mathbf{y},t})\Big] = \frac{1}{2}, \tag{2.23}$$

because the prediction $\widehat{h}_S(X)$ depends only on $(X, \{X_i, Y_i\}_{i\in[n_{\mathbf{y},t}]}, \rho)$. Roughly, Eq. (2.23) follows from the indifference of $\{f_{\mathbf{y}}\}_{\mathbf{y}\in\mathcal{Y}}$ (Item 3 in Claim 2.4.24), which states that if X is a member of the K-th node in the tree T, then for any X_i with $K_i < K$ there exists a bit $b \in \{0,1\}$ such that for all branches $\mathbf{y} \in \mathcal{Y}$ that contain node K, $f_{\mathbf{y}}(X_i) = b$. In particular, $Y = f_{\mathbf{y}}(X)$ is a uniformly random bit independent of $\{X_i, Y_i = f_{\mathbf{y}}(X_i)\}_{i\in[n_{\mathbf{y},t}]} \cup \{X\}$ given $G(\kappa_{\mathbf{y},t})$.

To flesh out the argument for Eq. (2.23) in further detail, fix $\kappa \in \mathbb{N}$, $(\kappa_1, \ldots, \kappa_{n_\kappa}) \in \mathbb{N}^{n_\kappa}$, $(\xi, \xi_1, \ldots, \xi_{n_\kappa}) \in \mathcal{X}^{n_\kappa+1}$, and $(\eta_1, \ldots, \eta_{n_\kappa}) \in \{0,1\}^{n_\kappa}$. Consider the following conditional probability of Y for a fixed $t \in \mathbb{N}$, assuming the event being conditioned upon has a positive probability.

$$\mathbb{P}\left[Y = 1 \;\middle|\; \begin{array}{c} \kappa_{\mathbf{y},t} = \kappa \\ \forall i \in [n_{\mathbf{y},t}]: \; X_i = \xi_i \;\wedge\; Y_i = \eta_i \;\wedge\; K_i = \kappa_i \\ K = \kappa_{\mathbf{y},t} \geq \max\{K_i : \; i \in [n_{\mathbf{y},t}]\} \\ X = \xi \notin \{X_i : \; i \in [n_{\mathbf{y},t}]\} \end{array}\right]$$

$$= \mathbb{P}\left[f_{\mathbf{y}}(X) = 1 \;\middle|\; \begin{array}{c} \kappa_{\mathbf{y},t} = \kappa \\ \forall i \in [n_{\mathbf{y},t}]: \; X_i = \xi_i \;\wedge\; f_{\mathbf{y}}(X_i) = \eta_i \;\wedge\; K_i = \kappa_i \\ K = \kappa_{\mathbf{y},t} \geq \max\{K_i : \; i \in [n_{\mathbf{y},t}]\} \\ X = \xi \notin \{X_i : \; i \in [n_{\mathbf{y},t}]\} \end{array}\right]$$

(Choice of Y and Y_i)

$$= \mathbb{P}\left[f_{\mathbf{y}}(X) = 1 \;\middle|\; \begin{array}{c} \kappa_{\mathbf{y},t} = \kappa \\ \forall i \in [n_{\mathbf{y},t}]: \; X_i = \xi_i \;\wedge\; K_i = \kappa_i \\ K = \kappa_{\mathbf{y},t} \geq \max\{K_i : \; i \in [n_{\mathbf{y},t}]\} \\ X = \xi \notin \{X_i : \; i \in [n_{\mathbf{y},t}]\} \end{array}\right]$$

(Indifference of $\{f_{\mathbf{y}}\}_{\mathbf{y}\in\mathcal{Y}}$ – Item 3 in Claim 2.4.24)

$$= \mathbb{P}\left[\mathbf{y}_t^j = 1 \;\middle|\; \begin{array}{c} \kappa_{\mathbf{y},t} = \kappa \\ \forall i \in [n_{\mathbf{y},t}]: \; X_i = \xi_i \;\wedge\; K_i = \kappa_i \\ K = \kappa_{\mathbf{y},t} \geq \max\{K_i : \; i \in [n_{\mathbf{y},t}]\} \\ X = \xi \notin \{X_i : \; i \in [n_{\mathbf{y},t}]\} \end{array}\right] = \frac{1}{2},$$

where j is the index of X in the K-th node in the tree. In the last line we have used the fact that $K = \kappa_{\mathbf{y},t}$ implies that X is on the branch corresponding to $\mathbf{y}$, and the final equality holds because $\mathbf{y}_t$ is a vector of uniformly random bits chosen independently of $\{X_i, K_i\}_{i\in[n_{\mathbf{y},t}]} \cup \{X, K, \kappa_{\mathbf{y},t}\}$ (note that $\kappa_{\mathbf{y},t}$ and $n_{\mathbf{y},t}$ are fully determined by t and $\mathbf{y}_{<t}$). This establishes Eq. (2.22), our second observation.

The first observation is used as follows. For every $\mathbf{y} \in \mathcal{Y}$,

$$\begin{aligned}
&\limsup_{n\to\infty} n \cdot \mathbb{E}_{\rho,S\sim P_{\mathbf{y}}^n}\Big[\mathrm{L}^{\text{0-1}}_{P_{\mathbf{y}}}\Big(\widehat{h}_S\Big)\Big] \\
&\quad \geq \limsup_{t\to\infty} n_{\mathbf{y},t} \cdot \mathbb{E}_{\rho,S\sim P_{\mathbf{y}}^{n_{\mathbf{y},t}}}\Big[\mathrm{L}^{\text{0-1}}_{P_{\mathbf{y}}}\Big(\widehat{h}_S\Big)\Big] \\
&\qquad\qquad\qquad\qquad\qquad\qquad (\text{If } b_j \text{ is a subsequence of } a_j \text{ then } \limsup a_j \geq \limsup b_j) \\
&\quad = \limsup_{t\to\infty} n_{\mathbf{y},t} \cdot \mathbb{P}_{\rho,S\sim P_{\mathbf{y}}^{n_{\mathbf{y},t}},(X,Y,K)\sim P_{\mathbf{y}},\rho}\Big[\widehat{h}_S(X) \neq Y\Big] \\
&\quad \geq \limsup_{t\to\infty} n_{\mathbf{y},t} \cdot \mathbb{P}_{\rho,S\sim P_{\mathbf{y}}^{n_{\mathbf{y},t}},(X,Y,K)\sim P_{\mathbf{y}}}\Big[\Big(\widehat{h}_S(X) \neq Y\Big) \ \wedge\ G(\kappa_{\mathbf{y},t})\Big] \\
&\quad = \limsup_{t\to\infty} n_{\mathbf{y},t} \cdot \mathbb{P}[G(\kappa_{\mathbf{y},t})] \cdot \mathbb{P}\Big[\widehat{h}_S(X) \neq Y \mid G(\kappa_{\mathbf{y},t})\Big] \\
&\quad \geq \limsup_{t\to\infty} \frac{d^{\kappa_{\mathbf{y},t}+1}}{9(d-1)} \cdot \frac{(d-1)d^{-\kappa_{\mathbf{y},t}}}{4} \cdot \mathbb{P}\Big[\widehat{h}_S(X) \neq Y \mid G(\kappa_{\mathbf{y},t})\Big] \\
&\qquad\qquad\qquad\qquad\qquad\qquad (\text{By Eq. (2.20) and choice of } n_{\mathbf{y},t}) \\
&\quad = \limsup_{t\to\infty} \frac{d}{36} \cdot \mathbb{P}_{\rho,S\sim P_{\mathbf{y}}^{n_{\mathbf{y},t}},(X,Y,K)\sim P_{\mathbf{y}}}\Big[\widehat{h}_S(X) \neq Y \mid G(\kappa_{\mathbf{y},t})\Big]. \qquad (2.24)
\end{aligned}$$

To complete the proof we use our second observation and Fatou's lemma as follows.

$$\begin{aligned}
&\mathbb{E}_{\mathbf{y}\sim \mathrm{U}(\mathcal{Y})}\Big[\limsup_{n\to\infty} n \cdot \mathbb{E}_{\rho,S\sim P_{\mathbf{y}}^n}\Big[\mathrm{L}^{\text{0-1}}_{P_{\mathbf{y}}}\Big(\widehat{h}_S\Big)\Big]\Big] \\
&\quad \geq \frac{d}{36} \cdot \mathbb{E}_{\mathbf{y}\sim \mathrm{U}(\mathcal{Y})}\Big[\limsup_{t\to\infty} \mathbb{P}_{\rho,S\sim P_{\mathbf{y}}^{n_{\mathbf{y},t}},(X,Y,K)\sim P_{\mathbf{y}}}\Big[\widehat{h}_S(X) \neq Y \mid G(\kappa_{\mathbf{y},t})\Big]\Big] \qquad (\text{By Eq. (2.24)}) \\
&\quad \geq \frac{d}{36} \cdot \limsup_{t\to\infty} \mathbb{E}_{\mathbf{y}\sim \mathrm{U}(\mathcal{Y})}\Big[\mathbb{P}_{\rho,S\sim P_{\mathbf{y}}^{n_{\mathbf{y},t}},(X,Y,K)\sim P_{\mathbf{y}}}\Big[\widehat{h}_S(X) \neq Y \mid G(\kappa_{\mathbf{y},t})\Big]\Big] \\
&\qquad\qquad\qquad\qquad\qquad\qquad (\text{Fatou's lemma (Lemma 2.4.25), } \mathbb{P}[\cdot] \leq 1) \\
&\quad = \frac{d}{36} \cdot \frac{1}{2} = \frac{d}{72}. \qquad (\text{By Eq. (2.22)})
\end{aligned}$$

This implies that there exists $\mathbf{y} \in \mathcal{Y}$ such that

$$\limsup_{n\to\infty} n \cdot \mathbb{E}_{S\sim P_{\mathbf{y}}^n}\Big[\mathrm{L}^{\text{0-1}}_{P_{\mathbf{y}}}\Big(\widehat{h}_S\Big)\Big] \geq \frac{d}{72}.$$

By the definition of $\limsup$, the inequality

$$\mathbb{E}_{S\sim P_{\mathbf{y}}^n}\Big[\mathrm{L}^{\text{0-1}}_{P_{\mathbf{y}}}\Big(\widehat{h}_S\Big)\Big] \geq \frac{d}{73 \cdot n}$$

holds for infinitely many values of $n \in \mathbb{N}$, as desired. □

2.5 Result for Half-Spaces

Notation 2.5.1. *Let $d \in \mathbb{N}$. We write $\mathbb{S}^{d-1} = \{x \in \mathbb{R}^d : \ \|x\|_2 = 1\}$ to denote the unit sphere in $\mathbb{R}^d$.*

Definition 2.5.2. *Let $d \in \mathbb{N}$. For any $\mathbf{w} \in \mathbb{S}^{d-1}$, let $h_{\mathbf{w}} : \mathbb{R}^d \to \{0,1\}$ be the half-space given by $h_{\mathbf{w}}(\mathbf{x}) = \mathbb{1}(\langle \mathbf{w}, \mathbf{x} \rangle > 0)$. The class of homogeneous half-spaces in $\mathbb{R}^d$ is $\mathcal{H}_d = \left\{h_{\mathbf{w}} : \mathbf{w} \in \mathbb{S}^{d-1}\right\}$.*

Definition 2.5.3. *Let $d \in \mathbb{N}$, let $H \subseteq \mathbb{S}^{d-1}$ be a set. We say that a set of points $\{x_1, \ldots, x_m\} \subseteq \mathbb{R}^d$ is openly shattered by H if for every vector $\mathbf{y} = (y_1, \ldots, y_m) \in \{0,1\}^m$, there exists an open set $W_{\mathbf{y}} \subseteq H$ such that*

$$\forall \mathbf{w} \in W_{\mathbf{y}} \; \forall i \in [m] : \; h_{\mathbf{w}}(x_i) = y_i. \tag{2.25}$$

Lemma 2.5.4. *Let $d \in \mathbb{N}$, and let $H \subseteq \mathbb{S}^{d-1}$ be an open set. Then there exists a set $X \subseteq \mathbb{S}^{d-1}$ such that $|X| = d-1$ and X is openly shattered by H.*

Proof. Fix a point x_0 in the interior of the H. Let $x_1, \ldots, x_{d-1} \in \mathbb{S}^{d-1}$ be points such that $x_0, x_1, \ldots, x_{d-1}$ is an orthonormal basis of $\mathbb{R}^d$.

For each $\mathbf{y} = (y_1, \ldots, y_{d-1}) \in \{0,1\}^{d-1}$, let

$$w'_{\mathbf{y}} = x_0 + \varepsilon \cdot \sum_{i \in [d-1]} \operatorname{sign}(y_i - 1/2) \cdot x_i$$

be a point with $\varepsilon > 0$ small enough such that $w_{\mathbf{y}}$ is in the interior of H, where $w_{\mathbf{y}}$ is the projection of $w'_{\mathbf{y}}$ onto $\mathbb{S}^{d-1}$. From the orthogonality of $\{x_0, \ldots, x_{d-1}\}$,

$$\forall i \in [d-1] : \; h_{w_{\mathbf{y}}}(x_i) = y_i.$$

For each $i \in [d-1]$ and $y \in \{0,1\}$, let $Q_{i,y} \subseteq \mathbb{S}^{d-1}$ be the set of $\mathbf{w}$ such that $h_{\mathbf{w}}(x_i) = y$. Because we use open half-spaces, $Q_{i,y}$ is open. Observe that for each $\mathbf{y} \in \{0,1\}^{d-1}$,

$$W_{\mathbf{y}} = H \cap \bigcap_{i \in [d-1]} Q_{i,y_i}$$

is open (as a finite intersection of open sets), and is non-empty because it contains $w_{\mathbf{y}}$. □

Lemma 2.5.5. *Let $d \in \mathbb{N}$. Then $\mathsf{VCL}(\mathcal{H}_d) \geq d-1$.*

Proof. We recursively construct an infinite $(d-1)$-VCL tree that is shattered by $\mathcal{H}_d$. Let $H_\lambda = \mathbb{S}^{d-1}$. For every $s \in 0,1,2,...$ do the following. For every $\mathbf{u} \in \{0,1\}^{ds}$, note that $H_{\mathbf{u}} \subseteq \mathbb{S}^{d-1}$ is open. Therefore, by Lemma 2.5.4, there exists $\mathbf{x}_{\mathbf{u}} = (x^1_{\mathbf{u}}, \ldots, x^{d-1}_{\mathbf{u}}) \subseteq \mathbb{S}^{d-1}$ of cardinality $d-1$ that is openly shattered by $H_{\mathbf{u}}$. Namely, for each $\mathbf{y} \in \{0,1\}^d$ there exists an open set $W_{\mathbf{y}} \subseteq H_{\mathbf{u}}$ such that Eq. (2.25) holds (for $x_i = x^i_{\mathbf{u}}$ and $m = d$). For each $\mathbf{y} \in \{0,1\}^d$, define $H_{\mathbf{u} \circ \mathbf{y}} = W_{\mathbf{y}}$.

We claim that $T = \{\mathbf{x}_{\mathbf{u}} : \mathbf{u} \in \left(\{0,1\}^d\right)^*\}$ is a $(d-1)$-VCL tree that is shattered by $\mathcal{H}_d$. Indeed, fix $t \in \mathbb{N}$ and $\mathbf{y} \in \{0,1\}^{td}$. Let $\mathbf{w} \in H_{\mathbf{u}}$. Then the choice of $\mathbf{x}_{\mathbf{u}}$ and $H_{\mathbf{u}}$ implies that

$$\forall s \in [t] \, \forall j \in [d] : \; h_{\mathbf{w}}(x^j_{\mathbf{y}_{\leq s-1}}) = y^j_s,$$

as desired. □

2.6 Directions for Future Work

We have shown a characterization of fine-grained learning rates in the instance specific setting. Directions for future work include characterizing the precise parameters $C, c \geq 0$ in Eq. (2.1), and obtaining an optimal gap factor (or equivalently, optimal parameters $\alpha, \beta \geq 0$ in Theorem 2.3.1).

Like the results of Bousquet et al. (2021), our results describe the *asymptotic* rate at which learning curves decay – but the results are silent as to the properties of learning curves for any finite number of samples. Devising a theory of learning curves that explains both asymptotic and non-asymptotic behavior in a unified way would be valuable.

Our result on semi-supervised learning (Item 3 in Section 2.1, Main Results) suggests that unlabeled data is not helpful in the setting of distribution-dependent learning curves. However, there are good reasons to believe that unlabeled data is helpful for learning in some real-world scenarios. We wonder how this tension could be resolved.

Chapter 3

A Trichotomy for Transductive Online Learning

3.1 Introduction

In classification tasks like PAC learning and online learning, the learner simultaneously confronts two distinct types of uncertainty: *labeling-related* uncertainty regarding the best labeling function, and *instance-related* uncertainty regarding the instances that the learner will be required to classify in the future. To gain insight into the role played by each type of uncertainty, researchers have studied modified classification tasks in which the learner faces only one type of uncertainty, while the other type has been removed.

In the context of PAC learning, Ben-David and Ben-David (2011) studied a variant of proper PAC learning in which the true labeling function is known to the learner, and only the distribution over the instances is not known. They show bounds on the sample complexity in this setting, which conceptually quantify the instance-related uncertainty. Conversely, labeling-related uncertainty is captured by PAC learning with respect to a fixed (e.g., uniform) domain distribution (Benedek and Itai, 1991), a setting which has been studied extensively.

In this chapter we improve upon the work of Ben-David et al. (1997), who quantified the label-related uncertainty in online learning. They introduced a model of *transductive online learning*,[1] in which the adversary commits in advance to a specific sequence of instances,

[1] Ben-David et al. (1997) call their model 'off-line learning with the worst sequence', but in this chapter we opt for 'transductive online learning', a name that has appeared in a number of publications, including Kakade and Kalai (2005); Pechyony (2008); Cesa-Bianchi and Shamir (2013); Syrgkanis, Krishnamurthy, and Schapire (2016). We remark there are at least two different variants referred to in the literature as 'transductive online learning'. For example, Syrgkanis et al. (2016) write of "a transductive setting (Ben-David et al., 1997) in which the learner knows the arriving contexts a priori, or, less stringently, knows only the set, but not necessarily the actual sequence or multiplicity with which each context arrives." That is, in one setting, the learner knows the sequence $(x_1, \dots, x_n)$ in advance, but in another setting the learner only knows the set $\{x_1, \dots, x_n\}$. One could distinguish between these two settings by calling them 'sequence-transductive' and 'set-transductive', respectively. Seeing as the current chapter deals exclusively with the sequence-transductive setting, we refer to it herein simply as the 'transductive' setting.

thereby eliminating the instance-related uncertainty.

The Transductive Online Learning Model

The model of learning studied in this chapter, due to Ben-David et al. (1997), is a zero-sum, finite, complete-information, sequential game involving two players, the *learner* and the *adversary*. Let $n \in \mathbb{N}$, let $\mathcal{X}$ and $\mathcal{Y}$ be sets, and let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a collection of functions called the *hypothesis class*.

The game proceeds as follows (see Section 3.2 for further formal definitions). First, the adversary selects an arbitrary sequence of *instances*, $x_1, \ldots, x_n \in \mathcal{X}$. Then, for each $t = 1, \ldots, n$:

1. The learner selects a *prediction*, $\hat{y}_t \in \mathcal{Y}$.

2. The adversary selects a *label*, $y_t \in \mathcal{Y}$.

In each step $t \in [n]$, the adversary must select a label y_t such that the sequence $(x_1, y_1), \ldots, (x_t, y_t)$ is *realizable* by $\mathcal{H}$, meaning that there exists $h \in \mathcal{H}$ satisfying $h(x_i) = y_i$ for all $i \in [t]$. The learner's objective is to minimize the quantity

$$M(A, x, h) = \left|\{t \in [n] : \ \hat{y}_t \neq h(x_t)\}\right|,$$

which is the number of mistakes when the learner plays strategy A and the adversary chooses sequence $x \in \mathcal{X}^n$ and labels consistent with hypothesis $h \in \mathcal{H}$. We are interested in understanding the value of this game,

$$M(\mathcal{H}, n) = \inf_{A \in \mathcal{A}} \ \sup_{x \in \mathcal{X}^n} \ \sup_{h \in \mathcal{H}} \ M(A, x, h),$$

where $\mathcal{A}$ is the set of all learner strategies. Note that neither party can benefit from using randomness, so without loss of generality we consider only deterministic strategies.

A Motivating Example

Transductive predictions of the type studied in this chapter appear in many real-world situations, essentially in any case where a party has a schedule or a to-do list known in advance of specific tasks that need to be completed in order, and there is some uncertainly as to the precise conditions that will arise in each task. As a concrete example, consider the logistics that the management of an airport faces when scheduling the work of passport-control officers.

Example 3.1.1. An airport knows in advance what flights are scheduled for each day. However, it does not know in advance exactly how many passengers will go through passport control each day (because tickets can be booked and cancelled in the last minute, and entire flights can be cancelled, delayed or rerouted, etc.). Each day can be 'regular' (the number of passengers is normal), or it can be 'busy' (more passengers than usual). Correspondingly, each day the airport must decide whether to schedule a 'regular shift' of passport-control officers, or an 'extended shift' that contains more officers.

If the airport assigns a standard shift for a busy day, then passengers experience long lines at passport control, and the airport suffers a loss of 1; if the airport assigns an extended shift for a regular day, then it wastes money on excess manpower, and it again experiences a loss of 1; If the airport assigns a regular shift to a regular day, or an extended shift to a busy day, then it experiences a loss of 0.

Hence, when the airport schedules its staff, it is essentially attempting to predict for each day whether it will be a regular day or a busy day, using information it knows well in advance about which flights are scheduled for each day. This is precisely a transductive online learning problem.

Our Contributions

I. Trichotomy. We show the following *trichotomy*. It shows that the rate at which $M(\mathcal{H}, n)$ grows as a function of n is determined by the VC dimension and the Littlestone dimension (LD).

Theorem (Informal Version of Theorem 3.4.1). Every hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ satisfies precisely one of the following:

1. $M(\mathcal{H}, n) = n$. This happens if $\mathsf{VC}(\mathcal{H}) = \infty$.
2. $M(\mathcal{H}, n) = \Theta(\log(n))$. This happens if $\mathsf{VC}(\mathcal{H}) < \infty$ and $\mathsf{LD}(\mathcal{H}) = \infty$.
3. $M(\mathcal{H}, n) = \Theta(1)$. This happens if $\mathsf{LD}(\mathcal{H}) < \infty$.

The $\Theta(\cdot)$ notations in Items 2. and 3. hide a dependence on $\mathsf{VC}(\mathcal{H})$, and $\mathsf{LD}(\mathcal{H})$, respectively.

The proof uses bounds on the number of mistakes in terms of the *threshold dimension* (Section 3.3), among other tools.

II. Littlestone classes. The minimal constant upper bound in the $\Theta(1)$ case of Theorem 3.4.1 is some value $C(\mathcal{H})$ that depends on the class $\mathcal{H}$, but the precise mapping $\mathcal{H} \mapsto C(\mathcal{H})$ is not known in general. Ben-David et al. (1997) showed that $C(\mathcal{H}) = \Omega\left(\sqrt{\log(\mathsf{LD}(\mathcal{H}))}\right)$. In Section 3.3 and Appendix A.1 we improve upon their result as follows.

Theorem (Informal Version of Theorem 3.3.1). Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ such that $\mathsf{LD}(\mathcal{H}) = d < \infty$. Then $M(\mathcal{H}, n) = \Omega(\log(d))$.

III. Multiclass setting. In Section 3.5, we generalize Theorem 3.4.1 to the multiclass setting with a finite label set $\mathcal{Y}$, showing a trichotomy based on the Natarajan dimension. The proof uses a simple result from Ramsey theory, among other tools.

Additionally, we show that the DS dimension of Daniely and Shalev-Shwartz (2014) does not characterize multiclass transductive online learning.

IV. Agnostic setting. In the *standard* (non-transductive) agnostic online setting, Ben-David, Pál, and Shalev-Shwartz (2009) showed that $R_{\mathsf{online}}(\mathcal{H}, n)$, the agnostic online regret for a hypothesis class $\mathcal{H}$ for a sequence of length n satisfies

$$\Omega\left(\sqrt{\mathsf{LD}(\mathcal{H}) \cdot n}\right) \leq R_{\mathsf{online}}(\mathcal{H}, n) \leq O\left(\sqrt{\mathsf{LD}(\mathcal{H}) \cdot n \cdot \log n}\right). \tag{3.1}$$

Later, Alon, Ben-Eliezer, Dagan, Moran, Naor, and Yogev (2021) showed an improved bound of $R_{\mathsf{online}}(\mathcal{H}, n) = \Theta\left(\sqrt{\mathsf{LD}(\mathcal{H}) \cdot n}\right)$.

In Section 3.6 we show a result similar to Eq. (3.1), for the *transductive* agnostic online setting.

Theorem (Informl Version of Theorem 3.6.1). Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, such that $0 < \mathsf{VC}(\mathcal{H}) < \infty$. Then the agnostic transductive regret for $\mathcal{H}$ is

$$\Omega\left(\sqrt{\mathsf{VC}(\mathcal{H}) \cdot n}\right) \leq R(\mathcal{H}, n) \leq O\left(\sqrt{\mathsf{VC}(\mathcal{H}) \cdot n \cdot \log n}\right).$$

Related Works

The general idea of bounding the number of mistakes by learning algorithms in sequential prediction problems was introduced in the seminal work of Littlestone (1988). That work introduced the *online* learning model, where the sequence of examples is revealed to the learner one example at a time. After each example x is revealed, the learner makes a prediction, after which the true target label y is revealed. The constraint, which makes learning even plausible, is that this sequence of (x, y) pairs should maintain the property that there is an (unknown) target concept in a given concept class $\mathcal{H}$ which is correct on the entire sequence. Littlestone (1988) also identified the optimal predictor for this problem (called the *SOA*, for *Standard Optimal Algorithm*), and a general complexity measure which is precisely equal to the optimal bound on the number of mistakes: a quantity now referred to as the *Littlestone dimension*.

Later works discussed variations on this framework. In particular, as mentioned, the transductive model discussed in the present work was introduced in the work of Ben-David et al. (1997). The idea (and terminology) of transductive learning was introduced by Vapnik and Chervonenkis (1974); Vapnik (1982); Kuhlmann (1999), to capture scenarios where learning may be easier due to knowing in advance which examples the learner will be tested on. Vapnik and Chervonenkis (1974); Vapnik (1982); Kuhlmann (1999) study transductive learning in a model closer in spirit to the PAC framework, where some uniform random subset of examples have their labels revealed to the learner and it is tasked with predicting the labels of the remaining examples. In contrast, Ben-David et al. (1997) study transductive learning in a sequential prediction setting, analogous to the online learning framework of Littlestone. In this case, the sequence of examples x is revealed to the learner all at once, and only the target labels (the y's) are revealed in an online fashion, with the label of each example revealed just after its prediction for that example in the given sequential order. Since a mistake bound in this setting is still required to hold for *any* sequence, for the purpose of analysis we may think of the sequence of x's as being a *worst case* set of examples and ordering thereof, for a given learning algorithm. Ben-David et al. (1997) compare and contrast the optimal mistake bound for this setting to that of the original online model of Littlestone (1988). Denoting by d the Littlestone dimension of the concept class, it is clear that the optimal mistake bound would be no larger than d. However, they also argue that the optimal mistake bound in the transductive model is never smaller than $\Omega(\sqrt{\log(d)})$ (as mentioned, we improve this to $\log(d)$ in the present work). They further exhibit a family of concept classes of variable d for which the transductive mistake bound is strictly smaller by a factor of $\frac{3}{2}$. They additionally provide a general equivalent description of the optimal transductive mistake bound in terms of the maximum possible rank among a certain family of trees, each representing the game tree for the sequential game on a given sequence of examples x.

In addition to these two models of sequential prediction, the online learning framework has also been explored in other variations, including exploring the optimal mistake bound under a *best-case* order of the data sequence x, or even a *self-directed* adaptive order in which the learning algorithm selects the next point for prediction from the remaining x's from the given sequence on each round (Ben-David et al., 1997; Ben-David, Eiron, and Kushilevitz, 1995; Goldman and Sloan, 1994; Ben-David and Eiron, 1998; Kuhlmann, 1999).

Unlike the online learning model of Littlestone, the transductive model additionally allows for nontrivial mistake bounds in terms of the sequence *length* n (the online model generally has $\min\{d, n\}$ as the optimal mistake bound). In this case, it follows immediately from the Sauer–Shelah–Perles lemma and a Halving technique that the optimal transductive mistake bound is no larger than $O(v \log(n))$ Kakade and Kalai (2005), where v is the VC dimension of the concept class Vapnik and Chervonenkis (1971, 1974).

3.2 Preliminaries

Notation 3.2.1. *Let $\mathcal{X}$ be a set and $n, k \in \mathbb{N}$. For a sequence $x = (x_1, \ldots, x_n) \in \mathcal{X}^n$, we write $x_{\leq k}$ to denote the subsequence $(x_1, \ldots, x_k)$. If $k \leq 0$ then $x_{\leq k}$ denotes the empty sequence, $\mathcal{X}^0$.*

Definition 3.2.2. *Let $k \in \mathbb{N}$, let $\mathcal{X}$ and $\mathcal{Y}$ be sets, and let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$. A sequence $(x_1, y_1), \ldots, (x_k, y_k) \in (\mathcal{X} \times \mathcal{Y})^k$ is realizable by $\mathcal{H}$, or $\mathcal{H}$-realizable, if*

$$\exists h \in \mathcal{H}\ \forall i \in [k]:\ h(x_i) = y_i.$$

Definition 3.2.3. *Let $\mathcal{X}$ be a set, let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, let $d \in \mathbb{N}$, and let $X = \{x_1, \dots, x_d\} \subseteq \mathcal{X}$. We say that $\mathcal{H}$ shatters X if for every $y \in \{0,1\}^d$ there exists $h \in \mathcal{H}$ such that for all $i \in [d]$, $h(x_i) = y_i$. The Vapnik–Chervonenkis (VC) dimension of $\mathcal{H}$ is $\mathsf{VC}(\mathcal{H}) = \sup\{|X| : X \subseteq \mathcal{X} \text{ finite} \wedge \mathcal{H} \text{ shatters } X\}$.*

Definition 3.2.4 (Littlestone, 1988)**.** *Let $\mathcal{X}$ be a set and let $d \in \mathbb{N}$. A Littlestone tree of depth d with domain $\mathcal{X}$ is a set*

$$T = \left\{x_u \in \mathcal{X} : \ u \in \bigcup_{s=0}^{d} \{0,1\}^s\right\}. \tag{3.2}$$

Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$. We say that $\mathcal{H}$ shatters a tree T as in Eq. (3.2) if for every $u \in \{0,1\}^{d+1}$ there exists $h_u \in \mathcal{H}$ such that

$$\forall i \in [d+1] : \ h(x_{u_{\leq i-1}}) = u_i.$$

The Littlestone dimension of $\mathcal{H}$, denoted $\mathsf{LD}(\mathcal{H})$, is the supremum over all $d \in \mathbb{N}$ such that there exists a Littlestone tree of depth d with domain $\mathcal{X}$ that is shattered by $\mathcal{H}$.

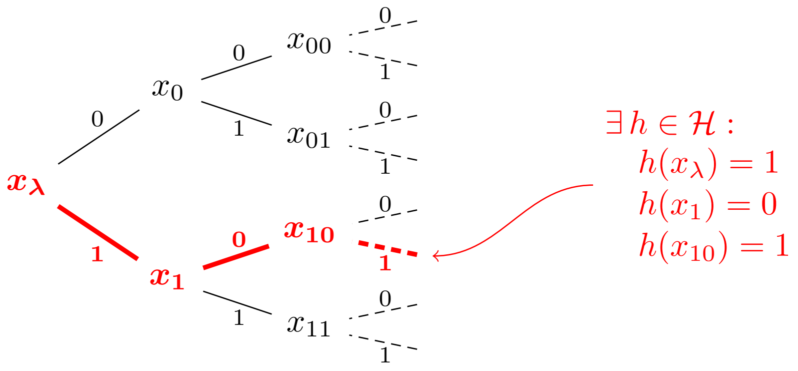


Figure 3.1: A shattered Littlestone tree of depth 2. The empty sequence is denoted by λ.

Source: Bousquet et al. (2021).

Theorem 3.2.5 (Littlestone, 1988)**.** *Let $\mathcal{X}$ be a set and let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ such that $d = \mathsf{LD}(\mathcal{H}) < \infty$. Then there exists a strategy for the learner that guarantees that the learner will make at most d mistakes in the standard (non-transductive) online learning setting, regardless of the adversary's strategy and of number of instances to be labeled.*

Theorem 3.2.6 (Sauer–Shelah–Perles; Shelah, 1972; Sauer, 1972)**.** *Let $n, d \in \mathbb{N}$, let $\mathcal{X}$ be a set of cardinality n, and let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ such that $\mathsf{VC}(\mathcal{H}) = d$. Then $|\mathcal{H}| \leq \sum_{i=0}^{n} \binom{n}{i} \leq \left(\frac{en}{d}\right)^d$.*

3.3 Quantitative Bounds

Littlestone Dimension: A Tighter Lower Bound

The Littlestone dimension is an upper bound on the number of mistakes, namely

$$\forall n \in \mathbb{N} : \; M(\mathcal{H}, n) \leq \mathsf{LD}(\mathcal{H}) \tag{3.3}$$

for any class $\mathcal{H}$. This holds because $\mathsf{LD}(\mathcal{H})$ is an upper bound on the number of mistakes for standard (non-transductive) online learning (Littlestone, 1988), and the adversary in the transductive setting is strictly weaker.

The Littlestone dimension also supplies a lower bound. We give a quadratic improvement on the previous lower bound of Ben-David et al. (1997), as follows.

Theorem 3.3.1. *Let $\mathcal{X}$ be a set, let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ such that $d = \mathsf{LD}(\mathcal{H}) < \infty$, and let $n \in \mathbb{N}$. Then*

$$M(\mathcal{H}, n) \geq \min\left\{\lfloor \log(d)/2 \rfloor, \lfloor \log\log(n)/2 \rfloor\right\}.$$

Proof idea for Theorem 3.3.1. Let T be a Littlestone tree of depth d that is shattered by $\mathcal{H}$, and let $\mathcal{H}_1 \subseteq \mathcal{H}$ be a collection of 2^{d+1} functions that witness the shattering. The adversary selects the sequence consisting of the nodes of T in breadth-first order. For each time step $t \in [n]$, let $\mathcal{H}_t$ denote the version space, i.e., the subset of $\mathcal{H}_1$ that is consistent with all previously-assigned labels. The adversary's adaptive labeling strategy at time t is as follows. If $\mathcal{H}_t$ is very unbalanced, meaning that a large majority of functions in $\mathcal{H}_t$ assign the same value to x_t, then the adversary chooses y_t to be that value. Otherwise, if $\mathcal{H}_t$ is fairly balanced, the adversary forces a mistake (it can do so without violating $\mathcal{H}$-realizability). The pivotal observation is that: (1) under this strategy, the version space decreases in cardinality significantly more during steps where the adversary forces a mistake compared to steps where it did not force a mistake; (2) let x_t be the t-th node in the breadth-first order. It has distance $\ell = \lfloor \log(t) \rfloor$ from the root of T. Because T is a binary tree, the subtree T' of T rooted at x_t is a tree of depth $d - \ell$. In particular, seeing as $\mathcal{H}_t$ contains only functions necessary for shattering T', $|\mathcal{H}_t| \leq 2^{d-\ell+1}$, so $\mathcal{H}_t$ must decrease not too slowly with t. Combining (1) and (2) yields that the adversary must be able to force a mistake not too rarely. A careful quantitative analysis shows that the number of mistakes the adversary can force is at least logarithmic in d. □

The full proof of Theorem 3.3.1 appears in Appendix A.1.

Threshold Dimension

We also show some bounds on the number of mistakes in terms of the threshold dimension.

Definition 3.3.2. *Let $\mathcal{X}$ be a set, let $X = \{x_1, \ldots, x_k\} \subseteq \mathcal{X}$, and let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$. We say that X is threshold-shattered by $\mathcal{H}$ if there exist $h_1, \ldots, h_k \in \mathcal{H}$ such that $h_i(x_j) = \mathbb{1}(j \leq i)$ for all $i, j \in [k]$. The threshold dimension of $\mathcal{H}$, denoted $\mathsf{TD}(\mathcal{H})$, is the supremum of the set of integers k for which there exists a threshold-shattered set of cardinality k.*

The following connection between the threshold and Littlestone dimensions is well-known.

Theorem 3.3.3 (Shelah, 1990; Hodges, 1997)**.** *Let $\mathcal{X}$ be a set, let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, and let $d \in \mathbb{N}$. Then:*

1. *If $\mathsf{LD}(\mathcal{H}) \geq d$ then $\mathsf{TD}(\mathcal{H}) \geq \lfloor \log d \rfloor$.*
2. *If $\mathsf{TD}(\mathcal{H}) \geq d$ then $\mathsf{LD}(\mathcal{H}) \geq \lfloor \log d \rfloor$.*

Item 1 in Theorem 3.3.3 and Eq. (3.3) imply that

$$\forall n \in \mathbb{N}: \; M(\mathcal{H}, n) \leq 2^{\mathsf{TD}(\mathcal{H})}$$

for any class $\mathcal{H}$. Similarly, Item 2 in Theorem 3.3.3 and Theorem 3.3.1 imply a mistake lower bound of $\Omega(\log\log(\mathsf{TD}(\mathcal{H})))$. However, one can do exponentially better than that, as follows.

Claim 3.3.4. *Let $\mathcal{X}$ be a set, let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ such that $d = \mathsf{TD}(\mathcal{H}) < \infty$, and let $n \in \mathbb{N}$. Then*

$$M(\mathcal{H}, n) \geq \min\left\{\lfloor \log(d) \rfloor, \lfloor \log(n) \rfloor\right\}.$$

One of the ideas used in this proof appeared in an example called σ_{worst} in Section 4.1 of Ben-David et al. (1997).

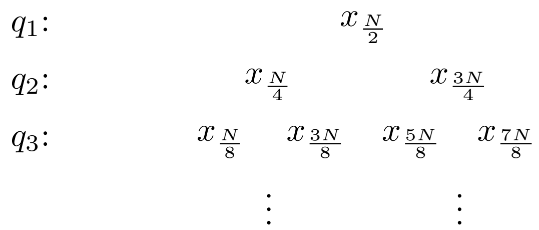


Figure 3.2: Construction of the sequence q in the proof of Claim 3.3.4. q is a breadth-first enumeration of the depicted binary tree.

Proof of Claim 3.3.4. Let $k = \min\left\{\lfloor \log(d) \rfloor, \lfloor \log(n) \rfloor\right\}$ and let $N = 2^k$. Let

$$X = \{x_1, \dots, x_{N-1}\} \subseteq \mathcal{X}$$

be a set that is threshold-shattered by functions $h_1, \dots, h_{N-1} \in \mathcal{H}$ and $h_i(x_j) = \mathbb{1}(j \leq i)$ for all $i, j \in [N-1]$. The strategy for the adversary is to present X in dyadic order, namely

$$x_{\frac{N}{2}}, x_{\frac{N}{4}}, x_{\frac{3N}{4}}, x_{\frac{N}{8}}, x_{\frac{3N}{8}}, x_{\frac{5N}{8}}, x_{\frac{7N}{8}}, \dots, x_{\frac{(2^k-1)N}{2^k}}.$$

More explicitly, the adversary chooses the sequence $q = q_1 \circ q_2 \circ \cdots \circ q_k$, where '$\circ$' denotes sequence concatenation and

$$q_i = \left(x_{\frac{1}{2^i}N}, x_{\frac{3}{2^i}N}, x_{\frac{5}{2^i}N}, x_{\frac{7}{2^i}N}, \dots, x_{\frac{(2^i-1)}{2^i}N}\right)$$

for all $i \in [k]$. See Figure 3.2.

We prove by induction that for each $i \in [k]$, all labels chosen by the adversary for the subsequences prior to q_i are $\mathcal{H}$-realizable, and additionally there exists an instance in subsequence q_i on which the adversary can force a mistake regardless of the learners predictions. The base case is that the adversary can always force a mistake on the first instance, q_1, by choosing the label opposite to the learner's prediction (both labels 0 and 1 are $\mathcal{H}$-realizable for this instance). Subsequently, for any $i > 1$, note that by the induction hypothesis, the labels chosen by the adversary for all instances in the previous subsequences are $\mathcal{H}$-realizable. In particular there exists an index $a \in [N]$ such that instance x_a has already been labeled, and all the labels chosen so far are consistent with h_a. Let b be the minimal integer such that $b > a$ and x_b has also been labeled. Then x_a and all labeled instances with smaller indices received label 1, while x_b and all labeled instances with larger indices received label 0. Because the sequence is dyadic, subsequence q_i contains an element x_m such that $a < m < b$. The adversary can force a mistake on x_m, because h_a and h_m agree on all previously labeled instances, but disagree on x_m. □

Claim 3.3.4 is used in the proof of the trichotomy (Theorem 3.4.1, below).

Finally, we note that for every $d \in \mathbb{N}$ there exists a hypothesis class $\mathcal{H}$ such that $d = \mathsf{TD}(\mathcal{H})$ and

$$\forall n \in \mathbb{N}: \ M(\mathcal{H}, n) = \min\{d, n\}.$$

Indeed, take $\mathcal{X} = [d]$ and $\mathcal{H} = \{0,1\}^{\mathcal{X}}$. The upper bound holds because $|\mathcal{X}| = d$, and the lower bound holds by Item 2 in Theorem 3.4.1, because $\mathsf{VC}(\mathcal{H}) = d$.

3.4 Trichotomy

Theorem 3.4.1. *Let $\mathcal{X}$ be a set, let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, and let $n \in \mathbb{N}$ such that $n \leq |\mathcal{X}|$.*

1. *If $\mathsf{VC}(\mathcal{H}) = \infty$ then $M(\mathcal{H}, n) = n$.*
2. *Otherwise, if $\mathsf{VC}(\mathcal{H}) = d < \infty$ and $\mathsf{LD}(\mathcal{H}) = \infty$ then*

$$\max\{\min\{d, n\}, \lfloor \log(n) \rfloor\} \leq M(\mathcal{H}, n) \leq O(d \log(n/d)). \tag{3.4}$$

   Furthermore, each of the bounds in Eq. (3.4) is tight for some classes. The $\Omega(\cdot)$ and $O(\cdot)$ notations hide universal constants that do not depend on $\mathcal{X}$ or $\mathcal{H}$.

3. *Otherwise, there exists an integer $C(\mathcal{H}) \leq \mathsf{LD}(\mathcal{H})$ (that depends on $\mathcal{X}$ and $\mathcal{H}$ but does not depend on n) such that $M(\mathcal{H}, n) \leq C(\mathcal{H})$.*

Proof of Theorem 3.4.1. For Item 1, assume $\mathsf{VC}(\mathcal{H}) = \infty$. Then there exists a set $X = \{x_1, \dots, x_n\} \subseteq \mathcal{X}$ of cardinality n that is shattered by $\mathcal{H}$. The adversary can force the learner to make n mistakes by selecting the sequence $(x_1, \dots, x_n)$, and then selecting labels $y_t = 1 - \hat{y}_t$ for all $t \in [n]$. This choice of labels is $\mathcal{H}$-realizable because X is a shattered set.

To obtain the upper bound in Item 2 the learner can use the *halving algorithm*, as follows. Let $x = (x_1, \dots, x_n)$ be the sequence chosen by the adversary, and let $\mathcal{H}|_x$ denote the collection of functions from elements of x to $\{0,1\}$ that are restrictions of functions in $\mathcal{H}$. For each $t \in \{0, \dots, n\}$, let

$$\mathcal{H}_t = \Big\{ f \in \mathcal{H}|_x : \ \ (\forall i \in [t] : \ f(x_i) = y_i) \Big\}$$

be a set called the *version space* at time t. At each step $t \in [n]$, the learner makes prediction

$$\hat{y}_t = \operatorname*{arg\,max}_{b \in \{0,1\}} \left| \Big\{ f \in \mathcal{H}_{t-1} : \ f(x_t) = b \Big\} \right|.$$

In words, the learner chooses $\hat{y}_t$ according to a majority vote among the functions in version space $\mathcal{H}_{t-1}$, and then any function whose vote was incorrect is excluded from the next version space, $\mathcal{H}_t$. This implies that for any $t \in [n]$, if the learner made a mistake at time t then

$$|\mathcal{H}_t| \leq \frac{1}{2} \cdot |\mathcal{H}_{t-1}|. \tag{3.5}$$

Let $M = M(\mathcal{H}, n)$. The adversary selects $\mathcal{H}$-realizable labels, so $\mathcal{H}_n$ cannot be empty. Hence, applying Eq. (3.5) recursively yields

$$1 \leq |\mathcal{H}_n| \leq 2^{-M} \cdot |\mathcal{H}_0| \leq 2^{-M} \cdot O\Big((n/d)^d\Big),$$

where the last inequality follows from $\mathsf{VC}(\mathcal{H}_0) \leq \mathsf{VC}(\mathcal{H}) = d$ and the Sauer–Shelah–Perles lemma (Theorem 3.2.6). Hence $M = O(d \log(n/d))$, as desired.

For the $\min\{d, n\}$ lower bound in Item 2, if $n \leq d$ then the adversary can force n mistakes by the same argument as in Item 1. For the logarithmic lower bound in Item 2, the assumption that $\mathsf{LD}(\mathcal{H}) = \infty$ and Theorem 3.3.3 imply that $\mathsf{TD}(\mathcal{H}) = \infty$, and in particular $\mathsf{TD}(\mathcal{H}) \geq n$, and this implies the bound by Claim 3.3.4.

For Item 3, the assumption $\mathsf{LD}(\mathcal{H}) = k < \infty$ and Theorem 3.2.5 imply that for any n, the learner will make at most k mistakes. This is because the adversary in the transductive setting is strictly weaker than the adversary in the standard online setting. So there exists some $C(\mathcal{H}) \in \{0, \dots, k\}$ as desired. □

Remark 3.4.2. *One can use Theorem 3.3.1 to obtain a lower bound for the case of Item 2 in Theorem 3.4.1. However, that yields a lower bound of* $\Omega(\log\log(n))$*, which is exponentially weaker than the bound we show.*

3.5 Multiclass Setting

The trichotomy of Theorem 3.4.1 can be generalized to the multiclass setting, in which the label set $\mathcal{Y}$ contains more than two labels. In this setting, the VC dimension is replaced by the Natarajan dimension (Natarajan, 1989), denoted ND, and the Littlestone dimension is generalized in the natural way. The result holds for *finite* sets $\mathcal{Y}$.

Theorem 3.5.1 (Informal Version of Theorem A.2.3)**.** *Let $\mathcal{X}$ be a set, let $\mathcal{Y}$ be a* finite *set, and let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$. Then $\mathcal{H}$ satisfies precisely one of the following:*

1. $M(\mathcal{H}, n) = n$*. This happens if* $\mathsf{ND}(\mathcal{H}) = \infty$*.*
2. $M(\mathcal{H}, n) = \Theta(\log(n))$*. This happens if* $\mathsf{ND}(\mathcal{H}) < \infty$ *and* $\mathsf{LD}(\mathcal{H}) = \infty$*.*
3. $M(\mathcal{H}, n) = O(1)$*. This happens if* $\mathsf{LD}(\mathcal{H}) < \infty$*.*

The proof of Theorem 3.5.1 appears in Appendix A.2, along with the necessary definitions. The main innovation in the proof involves the use of the multiclass threshold bounds developed in Appendix A.4, which in turn rely on a basic result from Ramsey theory, stated Appendix A.3.

The Case of an Infinite Label Set

It is interesting to observe that the analogy between the binary classification and multiclass classification settings breaks down when the label set $\mathcal{Y}$ is not finite.

Example 3.5.2. There exists a class $\mathcal{G} \subseteq \mathcal{Y}^{\mathcal{X}}$ such that $\mathcal{Y}$ is countable, $\mathsf{LD}(\mathcal{G})$ is infinite, but the class is learnable with a mistake bound of $M(\mathcal{G}, n) = 1$. To see this, let $\mathcal{X}$ be countable, and let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a class with $\mathsf{LD}(\mathcal{H}) = \infty$. For each $i \in \mathbb{N}$, let T_i be a Littlestone tree of depth i that is shattered by $\mathcal{H}$, and let $\{h_1^i, \dots, h_{2^{i+1}}^i\} \subseteq \mathcal{H}$ be a subset that witnesses the shattering. Let $\mathcal{G} = \{g_j^i : \ i \in \mathbb{N} \ \land \ j \in [2^{i+1}]\}$ be a set of functions such that $g_j^i(x) = (h_j^i(x), i, j)$ for all i, j. Let $\mathcal{Y} = \{0,1\} \times \mathbb{N} \times \mathbb{N}$. Observe that $\mathcal{G} \subseteq \mathcal{Y}^{\mathcal{X}}$ is a countable set of functions with a countable set of labels. Furthermore, $\mathsf{LD}(\mathcal{G}) = \infty$ because $\mathcal{G}$ shatters a sequence of suitable Littlestone trees corresponding to $T_1, T_2, \dots$. However, $\mathcal{G}$ can be learned with mistake bound 1, because a single example of the form $\big(x, (h_j^i(x), i, j)\big)$ reveals the correct labeling function h_j^i.

Recent work by Brukhim, Carmon, Dinur, Moran, and Yehudayoff (2022) has shown that multiclass PAC learning with infinite $\mathcal{Y}$ is not characterized by the Natarajan dimension, and that instead it is characterized by the DS dimension (introduced by Daniely and Shalev-Shwartz, 2014). It is therefore natural to ask whether the DS dimension might also characterize multiclass transductive online learning with infinite $\mathcal{Y}$. We show that the answer to that question is negative.

Recall the definition of the DS dimension.

Definition 3.5.3. *Let $d \in \mathbb{N}$, let $\mathcal{X}$ and $\mathcal{Y}$ be sets, and let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$. For an index $i \in [d]$ and vectors $y = (y_1, \ldots, y_d) \in \mathcal{Y}^d$, $y' = (y'_1, \ldots, y'_d) \in \mathcal{Y}^d$, we say that $\underline{y \text{ and } y' \text{ are } i\text{-neighbors}}$, denoted $y \sim_i y'$, if $\{j \in [d] : \ y_j \neq y'_j\} = \{i\}$. We say that $\mathcal{C} \subseteq \mathcal{Y}^d$ is a $\underline{d\text{-pseudocube}}$ if $\mathcal{C}$ is non-empty and finite, and*

$$\forall y \in \mathcal{C}\, \forall i \in [d]\, \exists y' \in \mathcal{C} : \ y \sim_i y'.$$

For a vector $x = (x_1, \ldots, x_d) \in \mathcal{X}^d$, we say that $\underline{\mathcal{H} \text{ DS-shatters } x}$ if the set

$$\mathcal{H}|_x := \left\{ \Big(h(x_1), \ldots, h(x_d) \Big) : \ h \in \mathcal{H} \right\} \subseteq \mathcal{Y}^d$$

contains a d-pseudocube.

Finally, the $\underline{\text{Daniely–Shalev-Shwartz (DS) dimension of } \mathcal{H}}$ is

$$\mathsf{DS}(\mathcal{H}) = \sup \left\{ d \in \mathbb{N} : \ \Big(\exists x \in \mathcal{X}^d : \ \mathcal{H} \text{ DS-shatterd } x \Big) \right\}.$$

See Brukhim et al. (2022) for figures and further discussion of the DS dimension.

The following claim shows that the DS dimension does not characterize transductive online learning, even when $\mathcal{Y}$ is finite.

Claim 3.5.4. *For every $n \in \mathbb{N}$, there exists a hypothesis class $\mathcal{H}_n$ such that $\mathsf{DS}(\mathcal{H}_n) = 1$ but the adversary in transductive online learning can force at least $M(\mathcal{H}_n, n) = n$ mistakes.*

Proof. Fix $n \in \mathbb{N}$ and let $\mathcal{X} = \{0, 1, 2, \ldots, n\}$. Consider a complete binary tree T of depth n such that for each $x \in \mathcal{X}$, all the nodes at depth x (at distance x from the root) are labeled by x, and each edge in T is labeled by a distinct label. Let $\mathcal{H}$ be a minimal hypothesis class that shatters T, namely, $\mathcal{H}$ shatters T and there does not exist a strict subset of $\mathcal{H}$ that shatters T.

Observe that $M(\mathcal{H}_n, n) = n$, because the adversary can present the sequence $0, 1, 2, \ldots, n-1$ and force a mistake at each time step. To see that $\mathsf{DS}(\mathcal{H}_n) = 1$, assume for contradiction that there exists a vector $x = (x_1, x_2) \in \mathcal{X}^2$ that is DS-shattered by $\mathcal{H}_n$, namely, there exists a 2-pseudocube $\mathcal{C} \subseteq \mathcal{H}|_x$. Note that $x_1 \neq x_2$, and without loss of generality $x_1 < x_2$ ($\mathcal{H}$ DS-shatters (x_1, x_2) if and only if it DS-shatters (x_2, x_1)).

Fix $y \in \mathcal{C}$. So $y = (h(x_1), h(x_2))$ for some $h \in \mathcal{H}$. Because $\mathcal{C}$ is a 2-pseudocube, there exists $y' \in \mathcal{C}$ that is a 1-neighbor of y. Namely, there exists $g \in \mathcal{H}$ such that $y' = (g(x_1), g(x_2)) \in \mathcal{C}$, $y'_1 \neq y_1$ and $y'_2 = y_2$. However, because each edge in T has a distinct label, and $\mathcal{H}$ is minimal, it follows that for any $x \in \mathcal{X}$,

$$g(x) = h(x) \implies \Big(\forall x' \in \{0, 1, \ldots, x\} : \ g(x') = h(x') \Big).$$

In particular, $g(x_2) = y'_2 = y_2 = h(x_2)$ implies $y'_1 = g(x_1) = h(x_1) = y_1$ which is a contradiction to the choice of y'. □

3.6 Agnostic Setting

The *agnostic* transductive online learning setting is defined analogously to the *realizable* (non-agnostic) transductive online learning setting described in Section 3.1. An early work by Cover (1965) observed that it is not possible for a learner to achieve vanishing regret in an agnostic online setting with complete information. Therefore, we consider a game with incomplete information, as follows.

First, the adversary selects an arbitrary sequence of instances, $x_1, \dots, x_n \in \mathcal{X}$, and reveals the sequence to the learner. Then, for each $t = 1, \dots, n$:

1. The adversary selects a label $y_t \in \mathcal{Y}$.

2. The learner selects a prediction $\hat{y}_t \in \mathcal{Y}$ and reveals it to the adversary.

3. The adversary reveals y_t to the learner.

At each time step $t \in [n]$, the adversary may select any $y_t \in \mathcal{Y}$, without restrictions.[2] The learner, which is typically randomized, has the objective of minimizing the *regret*, namely

$$R(A, \mathcal{H}, x, y) = \mathbb{E}[|\{t \in [n] : \ \hat{y}_t \neq y_t\}|] - \min_{h \in \mathcal{H}} |\{t \in [n] : \ h(x_t) \neq y_t\}|,$$

where the expectation is over the learner's randomness. In words, the regret is the expected excess number of mistakes the learner makes when it plays strategy A and the adversary chooses the sequence $x \in \mathcal{X}^n$ and labels $y \in \mathcal{Y}^n$, as compared to the number of mistakes made by the best fixed hypothesis $h \in \mathcal{H}$. We are interested in understanding the value of this game, namely

$$R(\mathcal{H}, n) = \inf_{A \in \mathcal{A}} \sup_{x \in \mathcal{X}^n} \sup_{y \in \mathcal{Y}^n} R(A, \mathcal{H}, x, y),$$

where $\mathcal{A}$ is the set of all learner strategies. We show the following result.

Theorem 3.6.1. *Let $\mathcal{X}$ be a set, let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, and let $n \in \mathbb{N}$ such that $n \leq |\mathcal{X}|$. Assume $0 < \mathsf{VC}(\mathcal{H}) < \infty$. Then the agnostic transductive regret for $\mathcal{H}$ on sequences of length n is*

$$\Omega\left(\sqrt{\mathsf{VC}(\mathcal{H}) \cdot n}\right) \leq R(\mathcal{H}, n) \leq O\left(\sqrt{\mathsf{VC}(\mathcal{H}) \cdot n \cdot \log\left(n/\mathsf{VC}(\mathcal{H})\right)}\right).$$

The upper bound in Theorem 3.6.1 follows directly from the Sauer–Shelah–Perles lemma (Theorem 3.2.6), together with the following well-known bound on the regret of the *Multiplicative Weights* algorithm (see, e.g., Theorem 21.10 in Shalev-Shwartz and Ben-David, 2014).

Theorem 3.6.2. *Let $\mathcal{X}$ be a set and let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be finite. There exists an algorithm for the standard (non-transductive) agnostic online learning setting that satisfies*

$$R_{\mathsf{online}}(\mathcal{H}, n) \leq \sqrt{2 \log\left(|\mathcal{H}|\right)}.$$

[2] Hence the name 'agnostic', implying that we make no assumptions concerning the choice of labels.

Theorem 3.6.2 implies the upper bound of Theorem 3.6.1, because the adversary in the transductive agnostic setting is weaker than the adversary in the standard agnostic setting.

We prove the lower bound of Theorem 3.6.1 using an anti-concentration technique from Lemma 14 of Ben-David et al. (2009). The proof appears in Appendix A.5.

Remark 3.6.3. *Additionally:*

1. *If* $\mathsf{VC}(\mathcal{H}) = 0$ *(i.e., classes with a single function) then the regret is* 0*.*

2. *If* $\mathsf{VC}(\mathcal{H}) < \infty$ *and* $\mathsf{LD}(\mathcal{H}) < \infty$ *then the regret is* $R(\mathcal{H}, n) = O\Big(\sqrt{\mathsf{LD}(\mathcal{H}) \cdot n}\Big)$*, by Alon et al. (2021) (as mentioned above). Namely, in some cases the* $\log(n)$ *factor in Theorem 3.6.1 can be removed.*

3. *If* $\mathsf{VC}(\mathcal{H}) = \infty$ *then the regret is* $\Omega(n)$*.*

3.7 Directions for Future Work

Some remaining open problems include:

1. Showing a sharper bound for the $\Theta(1)$ case in the trichotomy (Theorem 3.4.1). Currently, there is an exponential gap between the best known upper and lower bounds for Littlestone classes.

2. Showing sharper bounds for the $\Theta(\log n)$ cases in the trichotomy (Theorem 3.4.1) and multiclass trichotomy (Theorem A.2.3).

3. Showing a sharper bound for the agnostic case (Theorem 3.6.1).

4. Characterizing the number of mistakes in the multiclass setting with an infinite label set $\mathcal{Y}$ (complementing Theorem A.2.3).

Part II

PAC Verification

Chapter 4

Fundamentals: Interactive Proofs for Verifying Machine Learning

> *A simple idea underpins science: "trust, but verify". Results should always be subject to challenge from experiment. That simple but powerful idea has generated a vast body of knowledge. Since its birth in the 17th century, modern science has changed the world beyond recognition, and overwhelmingly for the better. But success can breed complacency. Modern scientists are doing too much trusting and not enough verifying – to the detriment of the whole of science, and of humanity.*
>
> The Economist, "How Science Goes Wrong" (2013)

4.1 Introduction

Data and data-driven algorithms are transforming science and society. State-of-the-art machine learning and statistical analysis algorithms use access to data at scales and granularities that would have been unimaginable even a few years ago. From medical records and genomic information to financial transactions and transportation networks, this revolution spans scientific studies, commercial applications and the operation of governments. It holds transformational promise, but also raises new concerns. If data analysis requires huge amounts of data and computational power, how can one verify the correctness and accuracy of the results? Might there be asymmetric cases, where performing the analysis is expensive, but verification is cheap?

There are many types of statistical analyses, and many ways to formalize the notion of verifying the outcome. In this work we focus on interactive proof systems Goldwasser, Micali, and Rackoff (1989) for verifying supervised learning, as defined by the PAC model of learning Valiant (1984). Our emphasis throughout is on access to the underlying data distribution as the critical resource: both quantitatively (how many samples are used for learning versus for verification), and qualitatively (what types of samples are used). We embark on tackling a series of new questions:

Suppose a learner (which we also call *prover*) claims to have arrived at a good hypothesis with regard to an unknown data distribution by analyzing random samples from the distribution. Can one verify the quality of the hypothesis with respect to the unknown distribution by using significantly fewer samples than the number needed to independently repeat the analysis? The crucial difference between this question and questions that appear in the property testing and distribution testing literature is that we allow the prover and verifier to engage in an *interactive* communication protocol (see Section 4.1 for a comparison). We are interested in the case where both the verifier and an honest prover are efficient (i.e., use polynomial runtime and sample complexity), and furthermore, a dishonest prover with unbounded computational resources cannot fool the verifier:

Question 4.1.1 (**Runtime and sample complexity of learning vs. verifying**). *Are there machine learning tasks for which the runtime and sample complexity of learning a good hypothesis is significantly larger than the complexity of verifying a hypothesis provided by someone else?*

In the learning theory literature, various types of access to training data have been considered, such as random samples, membership queries, and statistical queries. In the real world, some types of access are more costly than others. Therefore, it is interesting to consider whether it is possible to verify a hypothesis using a cheaper type of access than is necessary for learning:

Question 4.1.2 (**Sample type of learning vs. verifying**). *Are there machine learning problems where membership queries are necessary for finding a good hypothesis, but verification is possible using random samples alone?*

The answers to these fundamental questions are motivated by real-world applications. If data analysis requires huge amounts of data and computational resources while verification is a simpler task, then a natural approach for individuals and weaker entities would be to delegate the data collection and analysis to more powerful entities. Going beyond machine learning, this applies also to verifying the results of scientific studies without replicating the entire experiment. We elaborate on these motivating applications in Section 4.1 below.

PAC Verification: A Proposed Model

Our primary focus in this work is verifying the results of agnostic supervised machine learning algorithms that receive a labeled dataset, and aim to learn a classifier that predicts the labels of unseen examples. We introduce a notion of interactive proof systems for verification of PAC learning, which we call *PAC Verification* (see Definition 4.1.22). Here, the entity running the learning algorithms (which we refer to as the *prover* or the *learner*) proves the correctness of the results by engaging in an interactive communication protocol with a verifier. One special case is where the prover only sends a single message constituting an (NP-like) certificate of correctness. The honest prover should be able to convince the verifier to accept its proposed hypothesis with high probability. A dishonest prover (even an unbounded one) should not be able to convince the verifier to accept a hypothesis that is not sufficiently good (as defined below), except with small probability over the verifier's random coins and samples. The proof system is interesting if the amount of resources used for verification is significantly smaller than what is needed for performing the learning task. We are especially interested in *doubly-efficient* proof systems Goldwasser, Kalai, and Rothblum (2015), where the honest prover also runs in polynomial time.

More formally, let $\mathcal{X}$ be a set, and consider a distribution $\mathcal{D}$ over samples of the form (x, y) where $x \in \mathcal{X}$ and $y \in \{0,1\}$. Assume there is some *hypothesis class* $\mathcal{H}$, which is a set of functions $\mathcal{X} \to \{0,1\}$, and we are interested in finding a function $h \in \mathcal{H}$ that predicts the label y given a previously unseen x with high accuracy with respect to $\mathcal{D}$. To capture this we use the *loss function* $L_{\mathcal{D}}(h) = \mathbb{P}_{(x,y)\in\mathcal{D}}[h(x) \neq y]$. Our goal is to design protocols consisting of a prover and verifier that satisfy: (i) When the verifier interacts with an honest prover, with high probability the verifier outputs a hypothesis h that is ε-*good*, meaning that

$$L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(\mathcal{H}) + \varepsilon, \tag{4.1}$$

where $L_{\mathcal{D}}(\mathcal{H}) = \inf_{f\in\mathcal{H}} L_{\mathcal{D}}(f)$; (ii) For any (possibly dishonest and unbounded) prover, the verifier can choose to reject the interaction, and with high probability the verifier will not output a hypothesis that is not ε-*good*.

Observe that in the *realizable case* (or promise case), where we assume that $L_{\mathcal{D}}(\mathcal{H}) = 0$, one immediately obtains a strong result: given a hypothesis $\tilde{h}$ proposed by the prover, a natural strategy for the verifier is to take a few samples from $\mathcal{D}$, and accept if and only if $\tilde{h}$ classifies at most, say, a $\frac{9}{10}\varepsilon$-fraction of them incorrectly. From Hoeffding's inequality, taking $O\left(\frac{1}{\varepsilon^2}\right)$ samples is sufficient to ensure that with probability at least $\frac{9}{10}$ the *empirical loss*[1] of $\tilde{h}$ is $\frac{\varepsilon}{10}$-close to the true loss. Therefore, if $L_{\mathcal{D}}(\tilde{h}) \leq \frac{8}{10}\varepsilon$ then $\tilde{h}$ is accepted with probability at least $\frac{9}{10}$, and if $L_{\mathcal{D}}(\tilde{h}) > \varepsilon$ then $\tilde{h}$ is rejected with probability at least $\frac{9}{10}$. In contrast, PAC learning a hypothesis that with probability at least $\frac{9}{10}$ has loss at most ε requires $\Omega\left(\frac{d}{\varepsilon}\right)$ samples, where the parameter d, which is the VC dimension of the class, can be arbitrarily large.[2] That is, in the realizable case there is a sample complexity and time complexity separation of unbounded magnitude between learning and verifying. Furthermore, this result holds also under the weaker assumption that $L_{\mathcal{D}}(\mathcal{H}) \leq \frac{\varepsilon}{2}$.

Encouraged by this strong result, we focus on the *agnostic case*, where no assumptions are made regarding $L_{\mathcal{D}}(\mathcal{H})$. Here, things become more interesting, and deciding whether a proposed hypothesis $\tilde{h}$ is ε-good is non-trivial. Indeed, the verifier can efficiently estimate $L_{\mathcal{D}}(\tilde{h})$ using Hoeffding's inequality as before, but estimating the term $L_{\mathcal{D}}(\mathcal{H})$ on the right-hand side of (4.1) is considerably more challenging. If $\tilde{h}$ has a loss of say 15%, it could be an amazingly-good hypothesis compared to the other members of $\mathcal{H}$, or it could be very poor.

[1] I.e., the fraction of the samples that is misclassified.

[2] See preliminaries in Section 4.1 for more about VC dimension.

Distinguishing between these two cases may be difficult when $\mathcal{H}$ is a large and complicated class.

Related Models

We discuss two related models studied in prior work, and their relationship to the PAC verification model proposed in this work.

Property Testing. Goldreich, Goldwasser, and Ron (1998) initiated the study of a property testing problem that naturally accompanies proper PAC learning: Given access to samples from an unknown distribution $\mathcal{D}$, decide whether $L_{\mathcal{D}}(\mathcal{H}) = 0$ or $L_{\mathcal{D}}(\mathcal{H}) \geq \varepsilon$ for some fixed hypothesis class $\mathcal{H}$. Further developments and variations appeared in Kearns and Ron (2000) and Balcan, Blais, Blum, and Yang (2012). Blum and Hu (2018) consider *tolerant* closeness testing and a related task of distance approximation (see Parnas, Ron, and Rubinfeld, 2006), where the algorithm is required to approximate $L_{\mathcal{D}}(\mathcal{H})$ up to a small additive error. As discussed above, the main challenge faced by the verifier in PAC verification is approximating $L_{\mathcal{D}}(\mathcal{H})$. However, there is a crucial difference between testing and PAC verification: In addition to taking samples from $\mathcal{D}$, the verifier in PAC verification can also interact with a prover, and thus PAC verification can (potentially) be easier than testing. Indeed, this difference is exemplified by the *proper* testing question, where we only need to distinguish the zero-loss case from large loss. As discussed above, proper PAC verification is trivial. Proper testing, on the other hand, can be a challenging goal (and, indeed, has been the focus of a rich body of work). For the *tolerant* setting, we prove a separation between testing and PAC verification: we show a hypothesis class for which the help of the prover allows the verifier to save a (roughly) quadratic factor over the number of samples that are required for closeness testing or distance approximation. See Section 4.3 for further details.

Proofs of Proximity for Distributions. Chiesa and Gur (2018) study interactive proof systems for distribution testing. For some fixed property Π, the verifier receives samples from an unknown distribution $\mathcal{D}$, and interacts with a prover to decide whether $\mathcal{D} \in \Pi$ or whether $\mathcal{D}$ is ε-far in total variation distance from any distribution in Π. While that work does not consider machine learning, the question of verifying a lower bound ℓ on the loss of a hypothesis class can be viewed as a special case of distribution testing, where $\Pi = \{\mathcal{D} : \ L_{\mathcal{D}}(\mathcal{H}) \geq \ell\}$. Beyond our focus on PAC verification, an important distinction between the works is that in Chiesa and Gur's model and results, the honest prover's access to the distribution is unlimited – the honest prover can have complete information about the distribution. In this chapter, we focus on doubly-efficient proofs, where the verifier and the honest prover must both be efficient in the number of data samples they require. With real-world applications in mind, this focus seems quite natural.[3]

We survey further related works in Section 4.1.

[3] In Chiesa and Gur's setting, it would also be sufficient for the prover to only known the distribution up to $O(\varepsilon)$ total variation distance, and this can be achieved using random samples from the distribution. However, the number of samples necessary for the prover would be linear in the domain size, which is typically exponential, and so this approach would not work for constructing doubly-efficient PAC verification protocols.

Applications

The P vs. NP problem asks whether finding a solution ourselves is harder than verifying a solution supplied by someone else. It is natural to ask a similar question in learning theory: Are there machine learning problems for which learning a good hypothesis is harder than verifying one proposed by someone else? We find this theoretical motivation compelling in and of itself. Nevertheless, we now proceed to elaborate on a few more practical aspects of this question.

Delegation of Learning

In a commercial context, consider a scenario in which a client is interested in developing a machine learning (ML) model, and decides to outsource that task to a company P that provides ML services. For example, P promises to train a deep neural net using a big server farm. Furthermore, P claims to possess a large amount of high quality data that is not available to the client, and promises to use that data for training.

How could the client ascertain that a model provided by P is actually a good model? The client could use a general-purpose cryptographic delegation-of-computation protocol, but that would be insufficient. Indeed, a general-purpose delegation protocol can only ensure that P executed the computation as promised, but it cannot provide any guarantees about the quality of the outcome, and in particular cannot ensure that the outcome is ε-good: If P used skewed or otherwise low-quality training data (whether maliciously or inadvertently), a general-purpose delegation protocol has no way of detecting that. Moreover, even if the data and the execution of the computation were both flawless, this still provides no guarantees on the quality of the output, because an ML model might have poor performance despite being trained as prescribed.[4,5]

A different solution could be to have P provide a proof to establish that its output is indeed ε-good. In cases where the resource gap between learning and verifying is significant enough, the client could cost-effectively verify the proof, obtaining sound guarantees on the quality of the ML model it is purchasing from P.

Verification of Scientific Studies

It has been claimed that many or most published research findings are false (Ioannidis, 2005). Others refer to an ongoing *replication crisis* (Pashler and Wagenmakers, 2012; Fidler and Wilcox, 2018), where many scientific studies are hard or impossible to replicate or reproduce (e.g., Prinz, Schlange, and Asadullah, 2011; Begley and Ellis, 2012). Addressing these issues is a scientific and societal priority.

[4] E.g., a neural network might get stuck at a local minimum.

[5] Additionally, note that state-of-the-art delegation protocols are not efficient enough at present to make it practicable to delegate intensive ML computations. See the survey by Walfish and Blumberg (2015) for progress and challenges in developing such systems.

There are many factors contributing to this problem, including: structural incentives faced by researchers, scientific journals, referees, and funding bodies; the level of statistical expertise among researchers and referees; differences in the sources of data used for studies and their replication attempts; choice of standards of statistical significance; and norms pertaining to the publication of detailed replicable experimental procedures and complete datasets of experimental results.

We stress that the current chapter does not touch on the majority of these issues, and our discussion of the replication crisis (as well as our choice of quotation at the beginning of the chapter) does *not* by any means suggest that adoption of PAC verification protocols will single-handedly solve all issues pertaining to replication. Rather, the contribution of the current chapter with respect to scientific replication is very specific: we suggest that for some specific types of experiments, PAC verification can be used to design protocols that allow to verify the results of an experiment in a manner that uses a quantitatively smaller (or otherwise cheaper) set of independent experimental data than would be necessary for a traditional replication that fully repeats the original experiment. In Appendix B.1 we list four such types of experiments. We argue that devising PAC verification protocols that make scientific replication procedures even modestly cheaper for specific types of experiments is a worthwhile endeavor that could help increase the amount of scientific replication or verification that occurs, and decrease the prevalence of errors that remain undiscovered in the scientific literature.

Our Setting

In this chapter we consider the following form of interaction between a verifier and a prover.

Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a class of hypotheses, and let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \{0,1\}$. The verifier and the prover each have access to an oracle, denoted $\mathcal{O}_V$ and $\mathcal{O}_P$ respectively. In the simplest case, both oracles provide i.i.d. samples from $\mathcal{D}$. That is, each time an oracle is accessed, it returns a sample from $\mathcal{D}$ taken independently of all previous samples and events. In addition, the verifier and prover each have access to a (private) random coin value, denoted ρ_V and ρ_P respectively, which are sampled from some known distributions over $\{0,1\}^*$ independently of each other and of all other events. During the interaction, the prover and verifier take turns sending each other messages $w_1, w_2, \dots$, where $w_i \in \{0,1\}^*$ for all i. Finally, at some point during the exchange of messages, V halts and outputs either 'reject' or a hypothesis $h : \mathcal{X} \to \{0,1\}$. The goal of the verifier is to output an ε-*good* hypothesis, meaning that

$$L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(\mathcal{H}) + \varepsilon.$$

A natural special case of interest is when the prover's and verifier's oracles provide sample access to $\mathcal{D}$. The prover can learn a "good" hypothesis $\tilde{h} : \mathcal{X} \to \{0,1\}$ and send it to the verifier as its first message, as in Figure 4.1 above. The prover and verifier then exchange further messages, wherein the prover tries to convince the verifier that $\tilde{h}$ is ε-good, and the

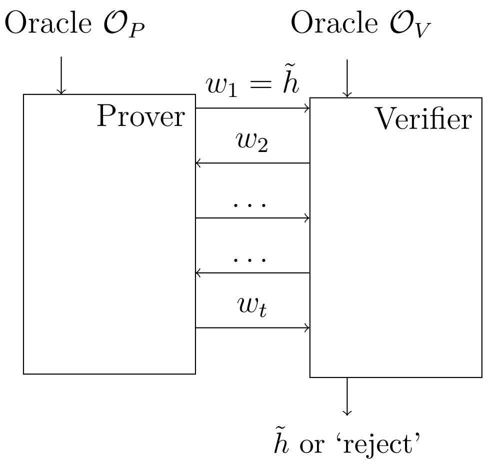


Figure 4.1: The verifier and prover each have access to an oracle, and they exchange messages with each other. Eventually, the verifier outputs a hypothesis, or rejects the interaction. One natural case is where the prover suggests a hypothesis $\tilde{h}$, and the verifier either accepts or rejects this suggestion.

verifier tries to assess the veracity of that claim. If the verifier is convinced, it outputs $\tilde{h}$, otherwise it rejects.

We proceed with an informal definition of PAC verification (see full definitions in Section 4.1). Before doing so, we first recall a relaxed variant of PAC learning, called *semi-agnostic* PAC learning, where we allow a multiplicative slack of $\alpha \geq 1$ in the error guarantee.

Definition (α-PAC Learnability – informal version of Definition 4.1.24)**.** *A class of hypothesis $\mathcal{H}$ is $\underline{\alpha\text{-PAC learnable}}$ (or $\underline{\text{semi-agnostic PAC learnable with parameter } \alpha}$) if there exists an algorithm A such that for every distribution $\mathcal{D}$ and every $\varepsilon, \delta > 0$, with probability at least $1 - \delta$, A outputs h that satisfies*

$$L_{\mathcal{D}}(h) \leq \alpha \cdot L_{\mathcal{D}}(\mathcal{H}) + \varepsilon. \tag{4.2}$$

PAC verification is the corresponding notion for interactive proof systems:

Definition (α-PAC Verifiability – informal version of Definition 4.1.22)**.** *A class of hypothesis $\mathcal{H}$ is $\underline{\alpha\text{-PAC verifiable}}$ if there exists a pair of algorithms (P, V) that satisfy the following conditions for every distribution $\mathcal{D}$ and every $\varepsilon, \delta > 0$:*

- ***Completeness.*** *After interacting with P, V outputs h such that with probability at least $1 - \delta$, $h \neq \text{reject}$ and h satisfies (4.2).*

- ***Soundness.*** *After interacting with any (possibly unbounded) prover P', V outputs h such that with probability at least $1 - \delta$, either $h = \text{reject}$ or h satisfies (4.2).*

Remark 4.1.3. *We insist on double efficiency; that is, that the sample complexity and running times of both V and P must be polynomial in $\frac{1}{\varepsilon}$, $\log\left(\frac{1}{\delta}\right)$, and perhaps also in some parameters that depend on $\mathcal{H}$, such as the VC dimension or Fourier sparsity of $\mathcal{H}$.*

Overview of Results

In this chapter, we start charting the landscape of machine learning problems with respect to Questions 4.1.1 and 4.1.2 mentioned above. First, in Section 4.2 we provide evidence for an affirmative answer to Questions 4.1.2. We show an interactive proof system that efficiently verifies the class of Fourier-sparse boolean functions, where the prover uses an oracle that provides query access, and the verifier uses an oracle that only provides random samples. In this proof system, both the verifier and prover send and receive messages.

The class of Fourier-sparse functions is very broad, and includes decision trees, bounded-depth boolean circuits and many other important classes of functions. Moreover, the result is interesting because it supplements the widely-held learning parity with noise (LPN) assumption, which entails that PAC learning this class from random samples alone without the help of a prover is hard (see Blum, Kalai, and Wasserman, 2003; Yu and Steinberger, 2016).

Theorem (Informal version of Theorem 4.2.6)**.** *Let $\mathcal{H}$ be the class of boolean functions $\{0,1\}^n \to \mathbb{R}$ that are t-sparse, as in Definition 4.1.20. Then $\mathcal{H}$ is 1-PAC verifiable with respect to the uniform distribution using a verifier that has access only to random samples of the form $(x, f(x))$, and a prover that has query access to f. The verifier in this protocol is not proper; the output is not necessarily t-sparse, but it is $\mathsf{poly}(n,t)$-sparse. The number of samples used by the verifier, the number of queries made by the prover, and their running times are all bounded by $\mathsf{poly}\Big(n, t, \log\left(\frac{1}{\delta}\right), \frac{1}{\varepsilon}\Big)$.*

Proof Idea. The proof uses two standard tools, albeit in a less-standard way. The first standard tool is the Kushilevitz–Mansour algorithm Kushilevitz and Mansour (1993), which can PAC learn any t-sparse function using random samples, but only if the set of non-zero Fourier coefficients is *known*. The second standard tool is the Goldreich–Levin algorithm Goldreich and Levin (1989); Goldreich (2007, Section 2.5.2.3), which can identify the set of non-zero Fourier coefficients, but requires *query access* in order to do so. The protocol combines the two tools in a manner that overcomes the limitations of each of them. First, the verifier executes the Goldreich–Levin algorithm, but whenever it needs to query the target function, it requests that the prover perform the query and send back the result. However, the verifier cannot trust the prover, and so the verifier engineers the queries in such a way that the answers to a certain random subset of the queries are known to the verifier based on its random sample access. This allows the verifier to detect dishonest provers. When the Goldreich–Levin algorithm terminates and outputs the set of non-zero coefficients, the verifier then feeds them as input to the Kushilevitz–Mansour algorithm to find an ε-good hypothesis using its random sample access. □

In Section 4.3 we formally answer Question 4.1.1 affirmatively by showing that a certain simple class of functions (generalized thresholds) exhibits a quadratic gap in sample complexity between learning and verifying:

Theorem (Informal version of Theorem 4.3.8)**.** *There exists a sequence of classes of functions*

$$\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \ldots \subseteq \{0,1\}^{\mathbb{R}}$$

such that for any fixed $\varepsilon, \delta \in (0, \frac{1}{2})$:

(i) The class $\mathcal{T}_d$ is proper 2-PAC verifiable, where both the verifier and prover have access to random samples, and the verifier requires only $\tilde{O}\left(\sqrt{d}\right)$ samples. Moreover, both the prover and verifier are efficient.

(ii) PAC learning the class $\mathcal{T}_d$ requires $\Omega(d)$ samples.

At this point, a perceptive reader would be justified in raising the following challenges. Perhaps 2-PAC verification requires fewer samples than 1-PAC learning simply because of the multiplicative slack factor of 2? Alternatively, perhaps the separation follows trivially from property testing results: maybe it is possible to achieve 2-PAC verification simply by having the verifier perform closeness testing using random samples, without needing the help of the prover except for finding the candidate hypothesis? The second part of the theorem dismisses both of these concerns.

Theorem (Informal version of Theorem 4.3.8 – Continued)**.** *Furthermore, for any fixed $\varepsilon, \delta \in (0, \frac{1}{2})$:*

(iii) 2-PAC learning the class $\mathcal{T}_d$ requires $\tilde{\Omega}(d)$ samples. This is true even if we assume that $L_{\mathcal{D}}(\mathcal{T}_d) > 0$, where $\mathcal{D}$ is the underlying distribution.[6]

(iv) Testing whether $L_{\mathcal{D}}(\mathcal{T}_d) \leq \alpha$ or $L_{\mathcal{D}}(\mathcal{T}_d) \geq \beta$ for any $0 < \alpha < \beta < \frac{1}{2}$ with success probability at least $1-\delta$ when $\mathcal{D}$ is an unknown distribution (without the help of a prover) requires $\tilde{\Omega}\left(d\right)$ random samples from $\mathcal{D}$.

Proof Idea. *(ii)* follows from a standard application of Theorem 4.1.15, because $\mathsf{VC}(\mathcal{T}_d) = d$. *(iii)* follows by a reduction from *(iv)*. We prove *(iv)* by showing a further reduction from the problem of approximating the support size of a distribution, and applying a lower bound for that problem (see Theorem 4.3.21).

For *(i)*, recall from the introduction that the difficulty in designing a PAC verification proof system revolves around convincing the verifier that the term $L_{\mathcal{D}}(\mathcal{H})$ in Equation (4.1) is large. Therefore, we design our class $\mathcal{T}_d$ such that it admits a simple *certificate of loss*, which is a string that helps the verifier ascertain that $L_{\mathcal{D}}(\mathcal{H}) \geq \ell$ for some value ℓ.

[6] In the case where $L_{\mathcal{D}}(\mathcal{T}_d) = 0$, 2-PAC learning is the same as PAC learning, so the stronger lower bound in *(ii)* applies.

To see how that works, first consider the simple class $\mathcal{T}$ of monotone increasing threshold functions $\mathbb{R} \to \{0,1\}$, as in Figure 4.2a on page 94 below. Observe that if there are two events $A = [0,a) \times \{1\}$ and $B = [b,1] \times \{0\}$ such that $a \leq b$ and $\mathcal{D}(A) = \mathcal{D}(B) = \ell$, then it must be the case that $L_{\mathcal{D}}(\mathcal{T}) \geq \ell$. This is true because $a \leq b$, and so if a monotone increasing threshold classifies any point in A correctly it must classify all point in B incorrectly. Furthermore, if the prover sends a description of A and B to the verifier, then the verifier can check, using a constant number of samples, that each of these events has weight approximately ℓ with high probability.

This type of certificate of loss can be generalized to the class $\mathcal{T}_d$, in which each function is a concatenation of d monotone increasing thresholds. A certificate of loss for $\mathcal{T}_d$ is simply a set of d certificates of loss $\{A_i, B_i\}_{i=1}^{d}$, one for each of the d thresholds. The question that arises at this point is how can the verifier verify d separate certificates while using only $\tilde{O}\left(\sqrt{d}\right)$ samples. This is performed using tools from distribution testing: the verifier checks whether the distribution of "errors" in the sets specified by the certificates is close to the prover's claims. I.e., whether the "weight" of 1-labels in each A_i and 0-labels in each B_i in the actual distribution, are close to the weights claimed by the prover. Using an identity tester for distributions this can be done using $O(\sqrt{d})$ samples (note that the identity tester need not be tolerant!). See Theorem B.5.1 for further details. □

In contrast, in Section 4.4 we show that verification is not always easier than learning:

Theorem (Informal version of Theorem 4.4.1)**.** *There exists a sequence of classes $\mathcal{H}_1, \mathcal{H}_2, \ldots$ such that:*

- *It is possible to PAC learn the class $\mathcal{H}_d$ using $\tilde{O}(d)$ samples.*
- *For any interactive proof system that proper 1-PAC verifies $\mathcal{H}_d$, in which the verifier uses an oracle providing random samples, the verifier must use at least $\Omega(d)$ samples.*

Remark 4.1.4. *The lower bound on the sample complexity of the verifier holds regardless of what oracle is used by the prover.*

Proof Idea. We specify a set $\mathcal{X}$ of cardinality $\Omega(d^2)$, and take $\mathcal{H}_d$ to be a randomly-chosen subset of all the balanced functions $\mathcal{X} \to \{0,1\}$ (i.e., functions f such that $|f^{-1}(0)| = |f^{-1}(1)|$). The sample complexity of PAC learning $\mathcal{H}_d$ follows from its VC dimension being $\tilde{O}(d)$. For the lower bound, consider proper PAC verifying $\mathcal{H}_d$ in the special case where the distribution $\mathcal{D}$ satisfies $\mathbb{P}_{(x,y)\in\mathcal{D}}[y=1] = 1$, but the marginal of $\mathcal{D}$ on $\mathcal{X}$ is unknown to the verifier. Because every hypothesis in the class assigns the incorrect label 0 to precisely half of the domain, a hypothesis achieves minimal loss if it assigns the 0 labels to a subset of size $\frac{|\mathcal{X}|}{2}$ that has minimal weight. Hence, the verifier must learn enough about the distribution to identify a specific subset of size $\frac{|\mathcal{X}|}{2}$ with weight close to minimal. We show that doing so requires $\Omega\left(\sqrt{|\mathcal{X}|}\right) = \Omega(d)$ samples. □

Finally, in Section 4.5, we show that in the setting of semi-supervised learning, where unlabeled samples are cheap, it is possible to perform PAC verification such that the verifier requires significantly less labeled samples than are required for learning. This verification uses a technique we call *query delegation*, and is efficient in terms of time complexity whenever there exists an efficient ERM algorithm that PAC learns the class using random samples.

Further Related Works

The growing role of data and predictive algorithms in a variety of fields has made the analysis of semi-unreliable data into a central research focus of the theoretical computer science (TCS) community. Recent research efforts that (broadly) fall into this theme include: (1) parameter estimation with greater emphasis on high dimensional data in the presence of partially unreliable data; (2) consideration of new corruption models such as list-decoding notions where some data is guaranteed to be properly sampled and the rest is subject to high error rate; (3) testing general properties of distributions beyond parameter estimation; and (4) analysis of machine learning algorithms with access to partially unreliable data. See Charikar, Steinhardt, and Valiant (2017); Diakonikolas, Kamath, Kane, Li, Moitra, and Stewart (2019, 2018); Ilyas, Jalal, Asteri, Daskalakis, and Dimakis (2017); Daskalakis, Gouleakis, Tzamos, and Zampetakis (2018). In contrast to all these directions, our focus is on interactive proof systems (or non-interactive certificates) by which an untrusted prover can convince a verifier that claimed results of a statistical analysis are correct, where the verifier is only allowed bounded access to the underlying data distribution.

A large body of work spanning the TCS and secure systems communities studies protocols for delegating computation to be performed by an untrusted prover (see e.g. Babai, Fortnow, Levin, and Szegedy, 1991; Micali, 1994; Goldwasser et al., 2015; Walfish and Blumberg, 2015). There are two significant differences between that line of work and the present chapter. First, in these protocols the input is fixed and known to the prover and the verifier. The question is whether a computation was performed correctly on this (fixed and known) input. In contrast, in our setting there is no fixed and known input: the distribution $\mathcal{D}$ is unknown to the verifier, and can only be accessed by sampling. Second, we are interested in guaranteeing that a certain statistical conclusion is valid with respect to this unknown distribution, regardless of whether any specific algorithm was executed as promised. That is, if some known learning algorithm was executed by the prover and happened to produce a poor result (e.g. a neural network got stuck in a local minimum), this result should be rejected by the verifier despite being the outcome of a correct computation. One final contrast with the literature on delegating computations is that the focus there is on verifying general computations, and this generality often results in impractical protocols. One benefit of our focus on specific and structured machine learning problems is that this focus may result in tailored protocols (for important problems) with improved efficiency.

The setting we investigate bears some similarity to sublinear proof verification (see e.g. Ergün, Kumar, and Rubinfeld, 2004; Rothblum, Vadhan, and Wigderson, 2013), where the verifier cannot read the entire input. However, in that setting the verifier enjoys *query* access to its input, whereas in our setting the verifier only gets random samples (a much more limited form of access).

Another related result, in the area of parameter estimation, is due to Diakonikolas, Kane, and Stewart (2017, Appendix C). They proved a gap between the sample complexity of estimating and verifying the center of a Gaussian. The verifier is given a parameter $\tilde{\theta} \in \mathbb{R}^n$ and access to samples from an n-dimensional Gaussian distribution $\mathcal{N}(\theta, I)$. The verifier can distinguish between the case $\tilde{\theta} = \theta$ and the case $\|\tilde{\theta} - \theta\|_2 > \varepsilon$ using $O(\sqrt{n}/\varepsilon^2)$ samples. This contrasts with estimating θ up to an ε error from samples alone (without access to $\tilde{\theta}$), which requires $\Omega(n/\varepsilon^2)$ samples. They show that the result is sharp, and also can be generalized to a setting of tolerant testing.[7]

Finally, Axelrod, Garg, Sharan, and Valiant (2020) investigates a setting somewhat resembling ours. They consider the task of "amplifying" a set of samples taken from some unknown target distribution, that is, producing an additional synthetic dataset that appears as if it was drawn from the target distribution. The authors show that generating a dataset close in total variation distance to the target distribution can be done using fewer samples from the distribution than are necessary for learning the distribution up to the same total variation distance.

Preliminaries

Probability

Notation 4.1.5. *For any probability space $(\Omega, \mathcal{F})$, let $\Delta(\Omega, \mathcal{F})$ denote the set of all probability distributions over $(\Omega, \mathcal{F})$. We will often simply write $\Delta(\Omega)$ to denote this set when the σ-algebra $\mathcal{F}$ is understood.*

Definition 4.1.6. *Let $\mathcal{P}, \mathcal{Q} \in \Delta(\Omega, \mathcal{F})$. The total variation distance between $\mathcal{P}$ and $\mathcal{Q}$ is*

$$\mathsf{TV}(\mathcal{P}, \mathcal{Q}) = \sup_{X \in \mathcal{F}} \Big|\mathcal{P}(X) - \mathcal{Q}(X)\Big| = \frac{1}{2}\Big\|\mathcal{P} - \mathcal{Q}\Big\|_1.$$

PAC Learning

We use the *Probably Approximately Correct* (PAC) definition of learning, introduced by Valiant (1984). See Shalev-Shwartz and Ben-David (2014) for a textbook on learning theory. Let $\mathcal{X}$ be a set, and let $\mathcal{H} \subseteq \mathbb{R}^{\mathcal{X}}$ be a class of functions, i.e. $\mathcal{H}$ is a subset of the functions $\mathcal{X} \to \mathbb{R}$. In this chapter, we use the ℓ_2 loss function, which is popular in machine learning.

Definition 4.1.7. *Let $h \in \mathcal{H}$, and let $\mathcal{D} \in \Delta(\mathcal{X} \times \{0,1\})$. The loss of h with respect to $\mathcal{D}$ is $L_{\mathcal{D}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[(h(x) - y)^2\Big]$. Furthermore, we denote $L_{\mathcal{D}}(\mathcal{H}) = \inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$.*

Remark 4.1.8. *In the special case of boolean labels, where $y \in \{0,1\}$ and $h : \mathcal{X} \to \{0,1\}$, the ℓ_2 loss function is the same as the 0-1 loss function: $L_{\mathcal{D}}(h) = \mathbb{P}_{(x,y)\sim\mathcal{D}}[h(x) \neq y]$.*

[7] That is, distinguishing between the case $d \geq \varepsilon$ and $d \leq \varepsilon/2$ for $d = \mathsf{TV}\big(\mathcal{N}(\tilde{\theta}, I), \mathcal{N}(\theta, I)\big)$.

Definition 4.1.9. *We say that $\mathcal{H}$ is agnostically PAC learnable if there exist an algorithm A and a function $m_{\mathcal{H}} : [0,1]^2 \to \mathbb{N}$ such that for any $\varepsilon, \delta > 0$ and any distribution $\mathcal{D} \in \Delta(\mathcal{X} \times \mathbb{R})$, if A receives as input a tuple of $m_{\mathcal{H}}(\varepsilon, \delta)$ i.i.d. samples from $\mathcal{D}$, then A outputs a function $h \in \mathcal{H}$ satisfying*

$$\mathbb{P}[L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(\mathcal{H}) + \varepsilon] \geq 1 - \delta.$$

In words, this means that h is probably (with confidence $1-\delta$) approximately correct (has loss at most ε worse than optimal). The point-wise minimal such function m is called the sample complexity of $\mathcal{H}$.

Definition 4.1.10. *Let $h \in \mathcal{H}$ and let $S = ((x_1, y_2), \dots, (x_m, y_m)) \in (\mathcal{X} \times \{0,1\})^m$. The empirical loss of h with respect to S is $L_S(h) = \frac{1}{m}\sum_{i\in[m]}(f(x_i) - y_i)^2$.*

Definition 4.1.11. *An empirical risk minimization algorithm (ERM) for class $\mathcal{H}$ is an agnostic PAC learning algorithm that takes $m = m_{\mathcal{H}}(\varepsilon, \delta)$ i.i.d. random samples from $\mathcal{D}$, denoted S, and outputs a hypothesis $h \in \arg\min_{f\in\mathcal{H}} L_S(f)$.*[8]

Definition 4.1.12. *We say that $\mathcal{H}$ has the uniform convergence property if there exists a function $m_{\mathcal{H}}^{\mathrm{UC}} : [0,1]^2 \to \mathbb{N}$ such that for any $\varepsilon, \delta > 0$ and any distribution $\mathcal{D} \in \Delta(\mathcal{X} \times \mathbb{R})$, if S is a tuple of $m_{\mathcal{H}}^{\mathrm{UC}}(\varepsilon, \delta)$ i.i.d. samples from $\mathcal{D}$, then $\mathbb{P}_S[\forall h \in \mathcal{H} : |L_S(h) - L_{\mathcal{D}}(h)| \leq \varepsilon] \geq 1-\delta$.*

The following definitions and result apply for the special case of boolean labels, where $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, and we only consider distributions $\mathcal{D} \in \Delta(\mathcal{X} \times \{0,1\})$.

Definition 4.1.13. *Let $h \in \mathcal{H}$ and $C \subseteq \mathcal{X}$. We denote by $h|_C$ the function $C \to \{0,1\}$ that agrees with h on C. The restriction of $\mathcal{H}$ to C is $\mathcal{H}|_C := \{h|_C : h \in \mathcal{H}\}$, and we say that $\mathcal{H}$ shatters C if $\mathcal{H}|_C = \{0,1\}^C$.*

Definition 4.1.14 (Vapnik and Chervonenkis, 1968, 1971)**.** *The VC dimension of $\mathcal{H}$ denoted $\mathsf{VC}(\mathcal{H})$ is the maximal size of a set $C \subseteq \mathcal{X}$ such that $\mathcal{H}$ shatters C. If $\mathcal{H}$ can shatter sets of arbitrary size, we say that the VC dimension is ∞.*

Theorem 4.1.15 (Vapnik and Chervonenkis, 1968, 1971; Blumer, Ehrenfeucht, Haussler, and Warmuth, 1989)**.** *The following are equivalent:*

1. $\mathsf{VC}(\mathcal{H}) < \infty$.
2. *$\mathcal{H}$ has the uniform convergence property.*
3. *$\mathcal{H}$ is agnostically PAC learnable.*
4. *Any ERM algorithm agnostically PAC learns $\mathcal{H}$ using $m_{\mathcal{H}}(\varepsilon, \delta)$ random samples.*

Furthermore, if $d = \mathsf{VC}(\mathcal{H}) < \infty$ then $m_{\mathcal{H}}(\varepsilon, \delta) = \Theta\left(\frac{d+\log\left(\frac{1}{\delta}\right)}{\varepsilon^2}\right)$ and $m_{\mathcal{H}}^{\mathrm{UC}}(\varepsilon, \delta) = \Theta\left(\frac{d+\log\left(\frac{1}{\delta}\right)}{\varepsilon^2}\right)$.

[8] Assuming that the minimum always exists for $\mathcal{H}$.

Fourier Analysis of Boolean Functions

To formulate and prove Theorem 4.2.6 below, we need several basic notions from the Fourier analysis of boolean functions. For a comprehensive introduction, see O'Donnell (2014).

Consider the linear space $\mathcal{F}$ of all functions of the form $f:\ \{0,1\}^n \to \mathbb{R}$.

Fact 4.1.16. *The operator $\langle \cdot,\cdot \rangle : \mathcal{F}^2 \to \mathbb{R}$ given by $\langle f, g\rangle := \mathbb{E}_{x\in\{0,1\}^n}[f(x)g(x)]$ constitutes an inner product, where $x \in \{0,1\}^n$ denotes sampling from the uniform distribution.*

Notation 4.1.17. *For any set $S \subseteq [n]$, $\chi_S :\ \{0,1\}^n \to \{0,1\}$ denotes the function $\chi_S(x) := (-1)^{\sum_i x_i}$.*

Fact 4.1.18. *The set $\{\chi_S :\ S \subseteq [n]\}$ is an orthonormal basis of $\mathcal{F}$. In particular, any $f \in \mathcal{F}$ has a unique representation $f(x) = \sum_{S\subseteq[n]} \widehat{f}(S)\chi_S(x)$, where $\widehat{f}(S) = \langle f, \chi_S\rangle$.*

Fact 4.1.19 (Parseval's identity)**.** *Let $f \in \mathcal{F}$. Then $\langle f, f\rangle = \sum_{S\subseteq[n]} \widehat{f}(S)^2$. In particular, if $f:\ \{0,1\}^n \to \{0,1\}$ then $\sum_{S\subseteq[n]} \widehat{f}(S)^2 = \mathbb{E}_x[f(x)] \le 1$.*

Definition 4.1.20. *Let $t \in \mathbb{N}$. A function $f :\ \{0,1\}^n \to \mathbb{R}$ is t-sparse if it has at most t non-zero Fourier coefficients, namely $|\{S \subseteq [n] :\ \widehat{f}(S) \neq 0\}| \le t$.*

Definition of PAC Verification

In Section 4.1 we informally described the setting of this chapter. Here, we complete that discussion by providing a formal definition of PAC verification, which is the main object of study in this chapter.

Notation 4.1.21. *We write*

$$[V^{\mathcal{O}_V}(x_V), P^{\mathcal{O}_P}(x_P)]$$

for the random variable denoting the output of the verifier V after interacting with a prover P, when V and P receive inputs x_V and x_P respectively, and have access to oracles $\mathcal{O}_V$ and $\mathcal{O}_P$ respectively. The inputs x_V and x_P can specify parameters of the interaction, such as the accuracy and confidence parameters ε and δ. This random variable takes values in $\{0,1\}^{\mathcal{X}} \cup \{\text{reject}\}$, namely, it is either a function $\mathcal{X} \to \{0,1\}$ or it is the value "reject". The random variable depends on the (possibly randomized) responses of the oracles, and on the random coins of V and P.

For a distribution $\mathcal{D}$, we write $V^{\mathcal{D}}$ (or $P^{\mathcal{D}}$) to denote use of an oracle that provides i.i.d. samples from the distributions $\mathcal{D}$. Likewise, for a function f, we write V^f (or P^f) to denote use of an oracle that provides query access to f. That is, in each access to the oracle, V (or P) sends some $x \in \mathcal{X}$ to the oracle, and receives the answer $f(x)$.

We also write

$$[V(S_V, \rho_V), P(S_P, \rho_P)] \ \in\ \{0,1\}^{\mathcal{X}} \cup \{\text{reject}\}$$

to denote the deterministic output of the verifier V after interacting with P in the case where V and P receive fixed random coin values ρ_V and ρ_P respectively, and receive fixed samples S_V and S_P from their oracles $\mathcal{O}_V$ and $\mathcal{O}_P$ respectively.

We are interested in classes $\mathcal{H}$ for which an ε-good hypothesis can always be verified with high probability via this form of interaction between an efficient prover and verifier, as formalized in the following definition. Note that the following definitions include an additional multiplicative slack parameter $\alpha \geq 1$ in the error guarantee. This parameter does not exist in the standard definition of PAC learning; the standard definition corresponds to the case $\alpha = 1$.

Definition 4.1.22 (**α-PAC Verifiability**). *Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a class of hypotheses, let $\mathbb{D} \subseteq \Delta(\mathcal{X} \times \{0,1\})$ be some family of distributions, and let $\alpha \geq 1$. We say that $\mathcal{H}$ is α-PAC verifiable with respect to $\mathbb{D}$ using oracles $\mathcal{O}_V$ and $\mathcal{O}_P$ if there exists a pair of algorithms (V, P) that satisfy the following conditions for every input $\varepsilon, \delta > 0$:*

- ***Completeness.*** *For any distribution $\mathcal{D} \in \mathbb{D}$, the random variable $h := [V^{\mathcal{O}_V}(\varepsilon, \delta), P^{\mathcal{O}_P}(\varepsilon, \delta)]$ satisfies*

$$\mathbb{P}\Big[h \neq \text{reject} \ \wedge \ \Big(L_{\mathcal{D}}(h) \leq \alpha \cdot L_{\mathcal{D}}(\mathcal{H}) + \varepsilon\Big)\Big] \geq 1 - \delta.$$

- ***Soundness.*** *For any distribution $\mathcal{D} \in \mathbb{D}$ and any (possibly unbounded) prover P', the random variable $h := [V^{\mathcal{O}_V}(\varepsilon, \delta), P'^{\mathcal{O}_P}(\varepsilon, \delta)]$ satisfies*

$$\mathbb{P}\Big[h \neq \text{reject} \ \wedge \ \Big(L_{\mathcal{D}}(h) > \alpha \cdot L_{\mathcal{D}}(\mathcal{H}) + \varepsilon\Big)\Big] \leq \delta.$$

Definition 4.1.23 (**Interactive Proof System for PAC Verification**). *A pair of algorithms (V, P) satisfying soundness and completeness as above, is called an interactive proof system that α-PAC verifies $\mathcal{H}$ with respect to $\mathbb{D}$ using oracles $\mathcal{O}_V$ and $\mathcal{O}_P$.*

Definition 4.1.24 (**α-PAC Learnability**). *Similarly, $\mathcal{H}$ is α-PAC learnable with respect to $\mathbb{D}$ using oracle $\mathcal{O}$ if there exists an algorithm A that for every input $\varepsilon, \delta > 0$ and every $\mathcal{D} \in \mathbb{D}$, outputs $h := A^{\mathcal{O}}(\varepsilon, \delta)$ such that $\mathbb{P}[L_{\mathcal{D}}(h) \leq \alpha \cdot L_{\mathcal{D}}(\mathcal{H}) + \varepsilon] \geq 1 - \delta$.*

Remark 4.1.25. *Some comments about these definitions:*

- *The behavior of the oracles $\mathcal{O}_V$ and $\mathcal{O}_P$ may depend on the specific underlying distribution $\mathcal{D} \in \mathbb{D}$, which is unknown to the prover and verifier. For example, they may provide samples from $\mathcal{D}$.*
- *We insist on double efficiency; that is, that the sample complexity and running times of both V and P must be polynomial in $\frac{1}{\varepsilon}$, $\log\left(\frac{1}{\delta}\right)$, and perhaps also in some parameters that depend on $\mathcal{H}$, such as the VC dimension or Fourier sparsity of $\mathcal{H}$.*

- *If for every $\varepsilon, \delta > 0$, and for any (possibly unbounded) prover P', the value $h := [V^{\mathcal{O}_V}(\varepsilon,\delta), P'^{\mathcal{O}_P}(\varepsilon,\delta)]$ satisfies $h \in \mathcal{H} \cup \{\text{reject}\}$ with probability 1 (i.e., V never outputs a function that is not in $\mathcal{H}$), then we say that $\underline{\mathcal{H} \text{ is proper } \alpha\text{-PAC verifiable}}$, and that the proof system $\underline{\text{proper } \alpha\text{-PAC verifies } \mathcal{H}}$.*

Remark 4.1.26. *An important type of learning (studied e.g. by Angluin, 1987 and Kushilevitz and Mansour, 1993) is* learning with membership queries with respect to the uniform distribution*. In this setting, the family $\mathbb{D}$ consists of distributions $\mathcal{D}$ such that: (1) the marginal distribution of $\mathcal{D}$ over $\mathcal{X}$ is uniform; (2) $\mathcal{D}$ has a target function $f : \mathcal{X} \to \{1, -1\}$ satisfying $\mathbb{P}_{(x,y)\sim\mathcal{D}}[y = f(x)] = 1$.*[9] *In Section 4.2, we will consider protocols for this type of learning that have the form $[V^{\mathcal{D}}, P^{f}]$, such that the verifier has access to an oracle providing random samples from a distribution $\mathcal{D} \in \mathbb{D}$, and the prover has access to an oracle providing query access to f, the target function of $\mathcal{D}$. This type of protocol models a real-world scenario where P has qualitatively more powerful access to training data than V.*

Organization of this Chapter

In Section 4.1 we formally define interactive proofs for PAC verification. In Section 4.1 we provide an overview of our results and their respective proof ideas.

Our first result appears in Section 4.2, where we answer Question 4.1.2 above affirmatively, showing that the broad and important class of Fourier-sparse boolean functions admits a doubly-efficient verification protocol in which the prover has query access, but the verifier only uses random samples. Note that according to the widely-held LPN assumption, learning this class is not possible without query access (see Section 4.1 for more about Fourier analysis, and Blum et al., 2003; Yu and Steinberger, 2016 for more about the LPN assumption).

In Section 4.3 we answer Question 4.1.1 above affirmatively by showing that a certain simple class of functions (generalized thresholds) exhibits a quadratic gap in sample complexity between learning and verifying. The verifier for this class is an NP-like verifier, in the sense that it takes as input a succinct witness string that helps it reach a decision.

Interestingly, however, verification is not always more efficient. In Section 4.4 we show a lower bound for a class of randomly-chosen functions, entailing that for this class, verification requires as many samples as learning does, up to a logarithmic factor.

Finally, in Section 4.5, where we show that, in the semi-supervised setting, PAC verification can reduce the number of labeled samples required compared to learning.

4.2 Efficient Verification for the Class of Fourier-Sparse Functions

The class $\mathcal{T}_d$ of multi-thresholds (discussed in Section 4.3 below) shows that in some cases verification is strictly easier than learning and closeness testing. The verification protocol

[9] Note that f is not necessarily a member of $\mathcal{H}$, so this is still an *agnostic* (rather than *realizable*) case.

for $\mathcal{T}_d$ has a single round, where the prover simply sends a hypothesis and a proof that it is (approximately) optimal. In this section, we describe a multi-round protocol that demonstrates that interaction is helpful for verification.

The interactive protocol we present PAC verifies the class of *Fourier-sparse functions*. This is a broad class of functions, which includes decision trees, DNF formulas with small clauses, and AC^0 circuits.[10] Every function $f: \{0,1\}^n \to \mathbb{R}$ can be written as a linear combination $f = \sum_{T \subseteq [n]} \widehat{f}(T)\chi_T$.[11] In Fourier-sparse functions, only a small number of coefficients are non-zero.

Remark 4.2.1. *According to the learning parity with noise (LPN) assumption (see Blum et al., 2003; Yu and Steinberger, 2016), it is not possible to learn the Fourier-sparse functions efficiently using random samples only. Therefore, the query delegation protocols discussed below in Section 4.5 cannot be used to obtain a doubly-efficient PAC verification protocol for this class, as we do in the current section.*

An important technicality is that throughout this section we focus solely on PAC verification with respect to families of distributions that have a uniform marginal over $\{0,1\}^n$, and have a target function $f: \{0,1\}^n \to \{1,-1\}$ such that $\mathbb{P}_{(x,y)\sim\mathcal{D}}[y = f(x)] = 1$. See further discussion in Remark 4.1.26 on page 72. One of the advantages of this setting is that in order to learn f, it is sufficient to approximate its heavy Fourier coefficients.

Notation 4.2.2. *Let $f: \{0,1\}^n \to \mathbb{R}$, and let $\tau \geq 0$. The set of τ-heavy coefficients of f is*

$$\widehat{f}^{\geq\tau} = \{T \subseteq [n] : \ |\widehat{f}(T)| \geq \tau\}.$$

Furthermore, approximating a single coefficient is easy given random samples from the uniform distribution (Claim 4.2.11). There are, however, an exponential number of coefficients, so approximating all of them is not feasible. This is where verification comes in. If the set of heavy coefficients is known, and if the function is Fourier-sparse, then one can efficiently learn the function by approximating that particular set of coefficients. The prover can provide the list of heavy coefficients, and then the verifier can learn the function by approximating these coefficients.

The challenge that remains in designing such a verification protocol is to verify that the provided list of heavy coefficients is correct. If the list contains some characters that are not actually heavy, no harm is done.[12] However, if a dishonest prover omits some of the heavy coefficients from the list, how can the verifier detect this omission? The following result provides an answer to this question.

[10] See Mansour (1994, Section 5.2.2, Theorems 5.15 and 5.16). (AC^0 is the set of functions computable by constant-depth boolean circuits with a polynomial number of AND, OR and NOT gates.)

[11] The real numbers $\widehat{f}(T)$ are called *Fourier coefficients*, and the functions χ_T are called *characters*.

[12] The verifier can approximate each coefficient in the list and discard of those that are not heavy. Alternatively, the verifier can include the additional coefficients in its approximation of the target function, because the approximation improves as the number of estimated coefficients grows (so long as the list is polynomial in n).

Lemma 4.2.3 (Interactive Goldreich–Levin). *There exists an interactive proof system* (V, P^*) *as follows. For every* $n \in \mathbb{N}$, $\delta > 0$, *every* $\tau \geq 2^{-\frac{n}{10}}$, *every function* $f : \{0,1\}^n \to \{0,1\}$, *and every prover* P, *let*

$$L_P = [V(S, n, \tau, \delta, \rho_V), P^f(n, \tau, \delta, \rho_P)]$$

be a random variable denoting the output of V *after interacting with the prover* P, *which has query access to* f, *where* $S = \Big((x_1, f(x_1)), \dots, (x_m, f(x_m))\Big)$ *is a random sample with* $x_1, \dots, x_m$ *taken independently and uniformly from* $\{0,1\}^n$, *and* ρ_V, ρ_P *are strings of private random coins.* L_P *takes values that are either a collection of subsets of* $[n]$, *or '*reject*'.*

The following properties hold:

- ***Completeness.*** $\mathbb{P}\Big[L_{P^*} \neq \text{reject} \ \wedge \ \widehat{f}^{\geq \tau} \subseteq L_{P^*}\Big] \geq 1 - \delta.$
- ***Soundness.*** *For any (possibly unbounded) prover* P,

$$\mathbb{P}\Big[L_P \neq \text{reject} \ \wedge \ \widehat{f}^{\geq \tau} \nsubseteq L_P\Big] \leq \delta.$$

- ***Double efficiency.*** *The verifier* V *uses at most* $O\left(\frac{n}{\tau}\log\left(\frac{n}{\tau}\right)\log\left(\frac{1}{\delta}\right)\right)$ *random samples from* f *and runs in time* $\mathsf{poly}\Big(n, \frac{1}{\tau}, \log\left(\frac{1}{\delta}\right)\Big)$. *The runtime of the prover* P^*, *and the number of queries it makes to* f, *are at most* $O\left(\frac{n^3}{\tau^5}\log\left(\frac{1}{\delta}\right)\right)$. *Whenever* $L_P \neq \text{reject}$, *the cardinality of* L_P *is at most* $O\left(\frac{n^2}{\tau^5}\log\left(\frac{1}{\delta}\right)\right)$.

Remark 4.2.4. *In Definition 4.1.23 we defined interactive proof systems specifically for PAC verification. The proof system in Lemma 4.2.3 is technically different, satisfying different completeness and soundness conditions. Additionally, in Definition 4.1.23 the verifier outputs a value that is either a function or '*reject*', while here the verifier outputs a value that is either a collection of subsets of* $[n]$, *or '*reject*'.*

The verifier V operates by simulating the Goldreich–Levin (GL) algorithm for finding $\widehat{f}^{\geq \tau}$. However, the GL algorithm requires query access to f, while V has access only to random samples. To overcome this limitation, V delegates the task of querying f to the prover P, who does have the necessary query access. Because P is not trusted, V engineers the set of queries it delegates to P in such a way that some random subset of them already appear in the sample S which V has received as input. This allows V to independently verify a random subset of the results sent by P, ensuring that a sufficiently dishonest prover is discovered with high probability.

As a corollary of Lemma 4.2.3, we obtain the following theorem, which is an interactive version of the Kushilevitz–Mansour algorithm (Kushilevitz and Mansour, 1993; see also Linial, Mansour, and Nisan, 1993). It says that the class of t-sparse boolean functions is efficiently PAC verifiable with respect to the uniform distribution using an interactive proof system of the form $[V^{\mathcal{D}}, P^f]$, where the prover has query access and the verifier has random samples.

Notation 4.2.5. *Let $\mathcal{X}$ be a finite set. We write $\mathbb{D}_{\mathcal{U}}^{\text{func}}(\mathcal{X})$ to denote the set of all distributions $\mathcal{D}$ over $\mathcal{X} \times \{1, -1\}$ that have the following two properties:*

- *The marginal distribution of $\mathcal{D}$ over $\mathcal{X}$ is uniform. Namely, $\sum_{y\in\{1,-1\}} \mathcal{D}\Big((x,y)\Big) = \frac{1}{|\mathcal{X}|}$ for all $x \in \mathcal{X}$.*
- *$\mathcal{D}$ has a target function $f :\ \mathcal{X} \to \{1,-1\}$ satisfying $\mathbb{P}_{(x,y)\sim\mathcal{D}}[y = f(x)] = 1$.*

Theorem 4.2.6. *Let $\mathcal{X} = \{0,1\}^n$, and let $\mathcal{H}$ be the class of functions $\mathcal{X} \to \mathbb{R}$ that are t-sparse, as in Definition 4.1.20. The class $\mathcal{H}$ is 1-PAC verifiable for any $\varepsilon \geq 4t \cdot 2^{-\frac{n}{10}}$ with respect to $\mathbb{D}_{\mathcal{U}}^{\text{func}}(\mathcal{X})$ by a proof system in which the verifier has access to random samples from a distribution $\mathcal{D} \in \mathbb{D}_{\mathcal{U}}^{\text{func}}(\mathcal{X})$, and the honest prover has oracle access to the target function $f : \mathcal{X} \to \{1,-1\}$ of $\mathcal{D}$. The running time of both parties is at most $\mathsf{poly}\Big(n, t, \frac{1}{\varepsilon}, \log\left(\frac{1}{\delta}\right)\Big)$. The verifier in this protocol is not proper; the output is not necessarily t-sparse, but it is $\mathsf{poly}\Big(n, t, \frac{1}{\varepsilon}, \log\left(\frac{1}{\delta}\right)\Big)$-sparse.*

The Interactive Goldreich–Levin Protocol

The verifier follows Protocol 4.1, which repeatedly applies Protocol 4.2 (IGL-Iteration).

```
$V$ performs the following:
  $r \leftarrow \left\lceil (\frac{4n}{\tau} + 1) \log\left(\frac{1}{\delta}\right) \right\rceil$
  for $i \in [r]$
      $L_i \leftarrow$ IGL-Iteration$(n, \tau)$
      if $L_i =$ reject
          output reject
  $L \leftarrow \bigcup_{i\in[r]} L_i$
  output $L$
```

Protocol 4.1: Interactive Goldreich–Levin: IGL(n, τ, δ)

We partition the proof of Lemma 4.2.3 into two claims. First, we show that if the prover is honest, then the output is correct.

Claim 4.2.7. *Consider an execution of* IGL-Iteration(n, τ) *for $\tau \geq 2^{-\frac{n}{10}}$. For any prover P and any randomness ρ_P, if V did not reject, and the evaluations provided by P were mostly honest, in the sense that*

$$\forall i \in [n] :\ \mathbb{P}_{x\in H}\Big[\tilde{f}(x \oplus e_i) \neq f(x \oplus e_i)\Big] \leq \frac{\tau}{4},$$

then

$$\mathbb{P}\Big[\widehat{f}^{\geq\tau} \subseteq L\Big] \geq \frac{1}{2},$$

Assumption: V receives a sample $S = \Big((x_1, f(x_1)), \ldots, (x_m, f(x_m))\Big)$ such that $m = \left\lceil \log\left(\frac{40n}{\tau^4} + 1\right)\right\rceil$, for all $i \in [m]$, $x_i \in \{0,1\}^n$ is chosen independently and uniformly, and $f(x_i) \in \{0,1\}$. "$\oplus$" denotes bitwise XOR; $\oplus$ of an empty set is 0.

1. V selects $i^* \in [n]$ uniformly at random, and then sends B to P, where

   $$B = \{b_1, \ldots, b_k\} \subseteq \{0,1\}^n$$

   is a basis chosen uniformly at random from the set of bases of the subspace

   $$H = \mathrm{span}(\{x_1 \oplus e_{i^*}, \ldots, x_m \oplus e_{i^*}\}).$$

   (For any j, e_j is a vector in which the j-th entry is 1 and all other entries are 0.)

2. P sends V the following set:

   $$\{(x \oplus e_i, \tilde{f}(x \oplus e_i)) : \; i \in [n] \;\wedge\; x \in H\},$$

   where for any z, $\tilde{f}(z)$ is purportedly the value of $f(z)$ obtained using P's query access to f.

3. V checks that for all $i \in [m]$, the evaluation $f(x_i)$ provided by V equals that which appeared in the sample S. If there are any discrepancies, V rejects and the interaction and terminates. Otherwise:

4. Let $\mathcal{K} = \{K : \; \varnothing \subsetneq K \subseteq [k]\}$. V Performs the following computation and outputs L:

```
$L \leftarrow \varnothing$
for $(y_1, \ldots, y_k) \in \{0,1\}^k$
    for $K \in \mathcal{K}$
        $x^K \leftarrow \bigoplus_{i \in K} b_i$
        $y^K \leftarrow \bigoplus_{i \in K} y_i$
    for $i \in [n]$
        $a_i \leftarrow \mathrm{majority}_{K \in \mathcal{K}} \left(\tilde{f}\left(x^K \oplus e_i\right) \oplus y^K\right)$
    add $\{i : \; a_i = 1\}$ and $\{i : \; a_i = 0\}$ to $L$
output $L$
```

Protocol 4.2: Interactive Goldreich–Levin Iteration: IGL-ITERATION(n, τ)

where the probability is over the sample S and the randomness ρ_V.

Proof of Claim 4.2.7. Let E denote the event in which the samples $\{x_1, \dots, x_m\}$ are linearly independent. From Claim B.8.2, $\mathbb{P}[E] \geq \frac{3}{4}$. We will show that

$$\forall T \in \widehat{f}^{\geq\tau} : \ \mathbb{P}[T \notin L \mid E] \leq \frac{\tau^2}{4}.$$

This is sufficient to prove the claim, because Parseval's identity entails that $|\widehat{f}^{\geq\tau}| \leq \frac{1}{\tau^2}$, and so from the union bound and the law of total probability,

$$\begin{aligned}\mathbb{P}\Big[\widehat{f}^{\geq\tau} \not\subseteq L\Big] &\leq \mathbb{P}\Big[\widehat{f}^{\geq\tau} \not\subseteq L \mid E\Big] + \mathbb{P}[\neg E] \\ &\leq |\widehat{f}^{\geq\tau}| \cdot \max_{T\in\widehat{f}^{\geq\tau}} \mathbb{P}[T \notin L \mid E] + \mathbb{P}[\neg E] \\ &\leq \frac{1}{\tau^2} \cdot \frac{\tau^2}{4} + \frac{1}{4} = \frac{1}{2}.\end{aligned}$$

Fix some $T \in \widehat{f}^{\geq\tau}$. Note that $T \in \widehat{f}^{\geq\tau}$ entails that

$$\mathbb{P}_{x\in\{0,1\}^n}[f(x) = \ell(x)] \geq \frac{1}{2} + \frac{\tau}{2} \tag{4.3}$$

where $\ell(x)$ is either $\bigoplus_{i\in T} x_i$ or $1 \oplus (\bigoplus_{i\in T} x_i)$. Now, consider the iteration of the outer loop in Step 4 in which $y_j = \ell(b_j)$ for all $j \in [k]$. For any $i \in [n]$ and any $K \in \mathcal{K}$, let

$$G_{i,K} := \mathbb{1}\Big(\tilde{f}\left(x^K \oplus e_i\right) = \ell(x^K \oplus e_i)\Big),$$

and observe that if $G_{i,K} = 1$ then from linearity of ℓ,

$$\tilde{f}\left(x^K \oplus e_i\right) \oplus y^K = \ell(x^K \oplus e_i) \oplus \ell(x^K) = \ell(e_i) = \begin{cases} \mathbb{1}(i \in T) & \ell(x) = \bigoplus_{i\in T} x_i \\ 1 \oplus \mathbb{1}(i \in T) & \ell(x) = 1 \oplus \left(\bigoplus_{i\in T} x_i\right). \end{cases}$$

Therefore, if

$$\forall i \in [n] : \ \frac{1}{|\mathcal{K}|} \sum_{K\in\mathcal{K}} G_{i,K} > \frac{1}{2}$$

then T will be added to L during the abovementioned iteration of the outer loop. Let

$$A_{i,K} := \mathbb{1}\Big(f\left(x^K \oplus e_i\right) = \ell(x^K \oplus e_i)\Big)$$

indicate cases where f agrees with ℓ, and let

$$D_{i,K} := \mathbb{1}\Big(\tilde{f}\left(x^K \oplus e_i\right) \neq f(x^K \oplus e_i)\Big)$$

indicates cases where P is dishonest about the value of f. Then for all $i \in [n]$,

$$\begin{aligned}\frac{1}{|\mathcal{K}|}\sum_K G_{i,K} &\geq \frac{1}{|\mathcal{K}|}\left(\sum_K A_{i,K} - \sum_K D_{i,K}\right)\\ &\overset{(i)}{\geq} \frac{1}{|\mathcal{K}|}\sum_K A_{i,K} - \frac{\tau}{4}\\ &\overset{(ii)}{\geq} \frac{1}{|\mathcal{K}|}\sum_K A^*_{i,K} - \frac{\tau}{4},\end{aligned}$$

where (i) follows from the assumption that P is dishonest about at most a $\frac{\tau}{4}$-fraction of the evaluations, and (ii) holds for

$$A^*_{i,K} = \begin{cases} A_{i,K} & x^K \oplus e_i \notin \{e_1, e_2, \ldots, e_n\}\\ 0 & \text{otherwise.}\end{cases}$$

Therefore, we can bound the probability that $T \notin L$ based on how well f and ℓ agree:

$$\begin{aligned}\mathbb{P}_{x_1,\ldots,x_k}[T \notin L \mid E] &\leq \mathbb{P}\left[\exists i \in [n] : \ \frac{1}{|\mathcal{K}|}\sum_{K\in\mathcal{K}} A^*_{i,K} \leq \frac{1}{2} + \frac{\tau}{4} \;\middle|\; E\right]\\ &\leq \sum_{i=1}^n \mathbb{P}\left[\frac{1}{|\mathcal{K}|}\sum_K A^*_{i,K} \leq \frac{1}{2} + \frac{\tau}{4} \;\middle|\; E\right] && \text{(union bound)}\\ &\overset{(i)}{\leq} \sum_{i=1}^n \mathbb{P}\left[\left|\frac{1}{|\mathcal{K}|}\sum_K A^*_{i,K} - \mu\right| \geq \frac{\tau}{4} - \frac{n}{2^n} \;\middle|\; E\right]\\ &\leq \sum_{i=1}^n \mathbb{P}\left[\left|\frac{1}{|\mathcal{K}|}\sum_K A^*_{i,K} - \mu\right| \geq \frac{\tau}{5} \;\middle|\; E\right] && (\tau \geq 2^{-\frac{n}{10}})\\ &\leq 25\sum_{i=1}^n \frac{\mathrm{Var}\left[\frac{1}{|\mathcal{K}|}\sum_K A^*_{i,K} \;\middle|\; E\right]}{\tau^2} && \text{(Chebyshev's inequality)}\\ &= 25\sum_{i=1}^n \frac{\mathrm{Var}\left[\sum_K A^*_{i,K} \mid E\right]}{|\mathcal{K}|^2\tau^2}\\ &\overset{(ii)}{\leq} 25\sum_{i=1}^n \frac{\sum_K \mathrm{Var}\left[A^*_{i,K} \mid E\right]}{|\mathcal{K}|^2\tau^2}\\ &\leq \frac{10n}{|\mathcal{K}|\tau^2} && \text{(variance of an indicator is } \leq \tfrac{1}{4})\\ &= \frac{10n}{(2^k-1)\tau^2}.\end{aligned}$$

Inequality (i) is justified because $\mu := \mathbb{E}\left[A^*_{i,K}\right] \geq \mathbb{E}[A_{i,K}] - \frac{n}{2^n} \geq \frac{1}{2} + \frac{\tau}{2} - \frac{n}{2^n}$, which follows from (4.3). For inequality (ii), we argue that given E, $\mathrm{Cov}[(]A^*_{i,K}, A^*_{i,K'}) \leq 0$ for any fixed $K, K' \in \mathcal{K}$, $K \neq K'$ and fixed $i \in [n]$. To see this, observe the following.

1. For any fixed sample $x_1, \ldots, x_m \in \{0,1\}^n$, the pair $(x^K, x^{K'})$ is distributed uniformly over the set $\{(u,u') : \ u,u' \in H \setminus \{0\} \ \wedge \ u \neq u'\}$. This is true because the base $\{b_1, \ldots, b_k\}$ is chosen uniformly from all bases of H, implying that $x^K = \bigoplus_{i \in K} b_i$ is a uniform point in $H \setminus \{0\}$. Furthermore, for any fixed value of x^K, $u_{\text{diff}} := x^K \oplus x^{K'} = \bigoplus_{i \in K \Delta K'} b_i$ is a uniform point in $H \setminus \{0, x^K\}$. Hence, for any fixed value of x^K, the point $x^{K'} = x^K \oplus u_{\text{diff}}$ is uniform in $H \setminus \{0, x^K\}$.
2. If $x_1, \ldots, x_m \in \{0,1\}^n$ are sampled independently and uniformly and we assume E occurs, then H is a random subspace of dimension m within $\{0,1\}^n$. Therefore, the pair $(x^K, x^{K'})$ is distributed uniformly over the set $\{(u,u') : \ u,u' \in \{0,1\}^n \setminus \{0\} \ \wedge \ u \neq u'\}$.
3. Hence, the pair $(x^K \oplus e_i, \ x^{K'} \oplus e_i)$ is distributed uniformly over the set
$$W = \{(u,u') : \ u,u' \in U \ \wedge \ u \neq u'\},$$
where $U = \{0,1\}^n \setminus \{e_i\}$.
4. Denote $A^* = \{x \in \{0,1\}^n : \ f(x) = \ell(x)\} \setminus \{e_1, e_2, \ldots, e_n\}$. Then
$$\begin{aligned}\mathrm{Cov}[(]A^*_{i,K}, A^*_{i,K'}) &= \mathbb{E}\Big[A^*_{i,K} A^*_{i,K'}\Big] - \mathbb{E}\Big[A^*_{i,K}\Big]\mathbb{E}\Big[A^*_{i,K'}\Big] \\ &= \mathbb{P}_{(x,y)\in W}[x \in A^*] \left(\mathbb{P}_{(x,y)\in W}[y \in A^* \mid x \in A^*] - \mathbb{P}_{(x,y)\in W}[x \in A^*]\right) \\ &\leq \mathbb{P}_{(x,y)\in W}[y \in A^* \mid x \in A^*] - \mathbb{P}_{(x,y)\in W}[x \in A^*] \\ &= \frac{|A^*| - 1}{|U| - 1} - \frac{|A^*|}{|U|} < 0.\end{aligned}$$

Finally, note that when E occurs (the samples $\{x_1, \ldots, x_m\}$ are linearly independent) then
$$k = m \geq \log\left(\frac{40n}{\tau^4} + 1\right),$$
and so
$$\mathbb{P}_{x_1,\ldots,x_k}[T \notin L \mid E] \leq \frac{10n}{(2^k - 1)\tau^2} \leq \frac{\tau^2}{4},$$
as desired. □

Next, we show that if the prover is dishonest, it will be rejected.

Claim 4.2.8. *Consider an execution of* IGL-ITERATION(n, τ)*. For any prover P and any randomness value ρ_P, if there exists $i \in [n]$ for which P was too dishonest in the sense that*
$$\mathbb{P}_{x \in H}\Big[\tilde{f}(x \oplus e_i) \neq f(x \oplus e_i)\Big] > \frac{\tau}{4},$$
then
$$\mathbb{P}[L = \text{reject}] \geq \frac{\tau}{4n},$$
where the probability is over the sample S and the randomness ρ_V.

Proof. Let E denote the event in which the index i^* selected by V is one for which P is too dishonest. We now focus on the case where this event occurred. Let $H^* = H \oplus e_{i^*}$, and let $X \subseteq H^*$ denote the subset of H^* that appeared in the sample S received by V. Observe that

$$1 - \frac{\tau}{4} > \mathbb{E}_{x \in H^*}\Big[\mathbb{1}(\tilde{f}(x) = f(x)) \mid E\Big] = \mathbb{E}_X\Big[\mathbb{E}_{x \in X}\Big[\mathbb{1}(\tilde{f}(x) = f(x))\Big] \mid E\Big] = \mathbb{E}_X[h_X \mid E],$$

where $h_X := \mathbb{E}_{x \in X}\Big[\mathbb{1}(\tilde{f}(x) = f(x))\Big]$, is the fraction of the sample on which P was honest. Notice that the only assumptions we have made about the distribution of X is that for every $x, x' \in H^*$, $\mathbb{P}[x \in X] = \mathbb{P}[x' \in X]$.

From Markov's inequality,

$$\mathbb{P}_X[h_X = 1 \mid E] \leq \mathbb{P}_X[h_X \geq 1 \mid E] \leq \mathbb{E}_X[h_X \mid E] < 1 - \frac{\tau}{4}.$$

This means that

$$\mathbb{P}[L = \text{reject} \mid E] = \mathbb{P}\Big[\exists x \in X : \ \tilde{f}(x) \neq f(x) \mid E\Big] \geq \frac{\tau}{4},$$

and we conclude that

$$\mathbb{P}[L = \text{reject}] \geq \mathbb{P}[L = \text{reject} \mid E]\mathbb{P}[E] \geq \frac{\tau}{4} \cdot \frac{1}{n}.$$

□

We now prove Lemma 4.2.3 using Claims 4.2.7 and 4.2.8.

Proof of Lemma 4.2.3. We show that the protocol $\mathrm{IGL}(n, \tau, \delta)$ satisfies the requirements of Lemma 4.2.3. For the completeness, consider the deterministic prover P^* that simply uses its query access to f in order to send the set

$$\{(x \oplus e_i, f(x \oplus e_i)) : \ i \in [n] \ \wedge \ x \in H\},$$

to V, and observe that P^* will never be rejected. Furthermore, for every $i \in [r]$, Claim 4.2.7 entails that $\mathbb{P}\Big[\widehat{f}^{\geq\tau} \nsubseteq L_i\Big] \leq \frac{1}{2}$. Thus, $\widehat{f}^{\geq\tau} \subseteq L_{P^*} \geq 1 - 2^{-r} \geq 1 - \delta$, as desired.

For the soundness, assume for contradiction that there exists some malicious prover $\tilde{P}$ such that

$$\mathbb{P}\Big[L_{\tilde{P}} \neq \text{reject} \ \wedge \ \widehat{f}^{\geq\tau} \nsubseteq L_{\tilde{P}}\Big] > \delta.$$

The IGL protocol consists of r executions of IGL-Iteration. We say that $\tilde{P}$ was *sufficiently honest* in a particular execution of IGL-Iteration if in that execution,

$$\forall i \in [n] : \ \mathbb{P}_{x \in H}\Big[\tilde{f}(x \oplus e_i) \neq f(x \oplus e_i)\Big] \leq \frac{\tau}{4}.$$

Let D be an indicator denoting the event that throughout the r executions, $\tilde{P}$ was too dishonest, meaning that the number of executions in which $\tilde{P}$ was sufficiently honest is strictly less than $\log(\frac{1}{\delta})$.

Consider the following two case:

- The dishonest case ($D = 1$): There were at least $r' := r - \log(\frac{1}{\delta}) \geq \frac{4n}{\tau}\log\left(\frac{1}{\delta}\right)$ executions in which $\tilde{P}$ was not sufficiently honest. From Claim 4.2.8, the probability of rejection in each of these r' repetitions is at least $\frac{\tau}{4n}$. Hence, because the rounds are independent,

$$\mathbb{P}[L_{\tilde{P}} \neq \text{reject} \mid D = 1] \leq \left(1 - \frac{\tau}{4n}\right)^{r'} \leq \left(1 - \frac{\tau}{4n}\right)^{\frac{4n}{\tau}\log\left(\frac{1}{\delta}\right)} \leq e^{-\log\left(\frac{1}{\delta}\right)} \leq \delta.$$

- The honest case ($D = 0$): Let $j_1, \dots, j_{r'} \in [r]$ be the rounds in which $\tilde{P}$ was sufficiently honest, with $r' \geq \log(\frac{1}{\delta})$. From Claim 4.2.7, with probability at least $1 - \delta$, the result L_{j_t} for each $t \in [r']$ satisfies

$$\mathbb{P}\left[\widehat{f}^{\geq\tau} \subseteq L_{j_t}\right] \geq \frac{1}{2}.$$

  Hence, because the rounds are independent,

$$\mathbb{P}\left[\widehat{f}^{\geq\tau} \nsubseteq L_{\tilde{P}} \mid D = 0\right] \leq 2^{-r'} \leq \delta.$$

Putting the two cases together, we obtain the desired contradiction:

$$\begin{aligned}\mathbb{P}\left[L_{\tilde{P}} \neq \text{reject} \ \wedge\ \widehat{f}^{\geq\tau} \nsubseteq L_{\tilde{P}}\right] &= \mathbb{P}\left[L_{\tilde{P}} \neq \text{reject} \ \wedge\ \widehat{f}^{\geq\tau} \nsubseteq L_{\tilde{P}} \mid D = 0\right]\mathbb{P}[D = 0] + \\ &\quad\ \mathbb{P}\left[L_{\tilde{P}} \neq \text{reject} \ \wedge\ \widehat{f}^{\geq\tau} \nsubseteq L_{\tilde{P}} \mid D = 1\right]\mathbb{P}[D = 1] \\ &\leq \mathbb{P}\left[\widehat{f}^{\geq\tau} \nsubseteq L_{\tilde{P}} \mid D = 0\right]\mathbb{P}[D = 0] + \\ &\quad\ \mathbb{P}[L_{\tilde{P}} \neq \text{reject} \mid D = 1]\mathbb{P}[D = 1] \\ &\leq \delta.\end{aligned}$$

This completes the proof of the soundness property. For the efficiency, observe the following:

- V performs $r = \left\lceil (\frac{4n}{\tau} + 1)\log\left(\frac{1}{\delta}\right)\right\rceil$ repetitions of the IGL-Iteration protocol, and each repetition requires $m = \left\lceil \log\left(\frac{40n}{\tau^4} + 1\right)\right\rceil$ fresh samples. Thus, V requires a total of

$$r \cdot m = O\left(\frac{n}{\tau}\log\left(\frac{n}{\tau}\right)\log\left(\frac{1}{\delta}\right)\right).$$

  random samples from f.

- P^* also performs r repetitions of the IGL-Iteration protocol, and makes at most $n2^m$ queries to f in each repetition. Thus, P^* uses at most

$$q = r \cdot n2^m = O\left(\frac{n^3}{\tau^5}\log\left(\frac{1}{\delta}\right)\right)$$

  queries to f.

- P^* runs in time $O(q)$, and V runs in time polynomial in q.
- For the bound on the cardinality of L_P, observe that V performs r repetitions of IGL-Iteration, and in each repetition, the number of items added to the list in Step 4 is at most $2^k \leq 2^m$. Thus, the total list length is at most

$$r \cdot 2^m = O\left(\frac{n^2}{\tau^5} \log\left(\frac{1}{\delta}\right)\right).$$

This completes the proof. □

Remark 4.2.9. *It is possible to run all repetitions of the IGL protocol in parallel such that only* 2 *messages are exchanged.*

Efficient Verification of Fourier-Sparse Functions

The verification protocol of Theorem 4.2.6 is described in Section 4.2. In the IGL protocol, we worked with functions $f : \{0,1\}^n \to \{0,1\}$. Now, we move to working with functions $f : \{0,1\}^n \to \{1,-1\}$. We translate data from $\{1,-1\}$ to $\{0,1\}$ as follows: $b \in \{1,-1\}$ is mapped to $\frac{1-b}{2} \in \{0,1\}$, and $b \in \{0,1\}$ is mapped to $(-1)^b \in \{1,-1\}$.

```
$V$ performs the following:
  $\tau \leftarrow \frac{\varepsilon}{4t}$
  $L \leftarrow \mathrm{IGL}(n, \tau, \frac{\delta}{2})$
  if $L = $ reject
      output reject
  else
      $\lambda \leftarrow \sqrt{\frac{\varepsilon}{8|L|}}$
      for $T \in L$
          $\alpha_T \leftarrow \textsc{EstimateCoefficient}(T, \lambda, \frac{\delta}{2|L|})$
      $h \leftarrow \sum_{T \in L} \alpha_T \chi_T$
      output $h$
```

Protocol 4.3: PAC verification of t-sparse functions: $\textsc{VerifyFourierSparse}(n, t, \varepsilon, \delta)$

Remark 4.2.10. *The output of* $\textsc{VerifyFourierSparse}$ *is a function* $h : \{0,1\}^n \to \mathbb{R}$*, not necessarily a boolean function.*

```
$m \leftarrow \left\lceil \frac{2\ln(2/\delta)}{\lambda^2} \right\rceil$
**for** $i \in [m]$
    **sample** $(x_i, y_i) \leftarrow \mathcal{D}$                    ▷ Takes i.i.d. samples from $\mathcal{D}$.
$\alpha_T \leftarrow \sum_{i=1}^{m} y_i \chi_T(x_i)$
**output** $\alpha_T$
```

Algorithm 4.1: Estimating a Fourier coefficient: EstimateCoefficient(T, λ, δ)

Proof

Theorem 4.2.6 follows from Lemma 4.2.3 via standard techniques (see exposition in Mansour, 1994). The proof is provided below for completeness. We start with the following claim.

Claim 4.2.11. *Let $\lambda, \delta > 0$, $T \subseteq [n]$, and let $\mathcal{D} \in \mathbb{D}_{\mathcal{U}}^{\text{func}}(\{0,1\}^n)$ with target function $f : \{0,1\}^n \to \{1, -1\}$. Then* EstimateCoefficient(T, λ, δ) *uses $m = \left\lceil \frac{2\ln(2/\delta)}{\lambda^2} \right\rceil$ random samples from $\mathcal{D}$ and outputs a number α_T such that*

$$\mathbb{P}\Big[|\alpha_T - \widehat{f}(T)| \geq \lambda\Big] \leq \delta,$$

where the probability is over the samples.

Proof. Let $\Big((x_1, f(x_1), \ldots, (x_m, f(x_m))\Big)$ denote the sample. Recall that

$$\widehat{f}(T) = \langle f, \chi_T \rangle := \mathbb{E}_{x \in \{0,1\}^n}[f(x)\chi_T(x)],$$

where $|f(x)\chi_T(x)| \leq 1$. Therefore, if we take

$$\alpha_T := \sum_{i=1}^{m} f(x_i)\chi_T(x_i)$$

then Hoeffding's inequality yields

$$\mathbb{P}\Big[\Big|\alpha_T - \widehat{f}(T)\Big| \geq \lambda\Big] \leq 2\exp(-m\lambda^2/2) \leq \delta.$$

□

Proof of Theorem 4.2.6. Fix $\varepsilon, \delta > 0$ and a distribution $\mathcal{D} \in \mathbb{D}_{\mathcal{U}}^{\text{func}}(\{0,1\}^n)$ with target function $f : \{0,1\}^n \to \{1, -1\}$. Consider an execution of VerifyFourierSparse$(n, t, \varepsilon, \delta)$. We show completeness, soundness, double efficiency and sparsity.

- **Completeness.** Assume that the prover P was honest. Then from Lemma 4.2.3, with probability at least $1 - \frac{\delta}{2}$, $L \neq$ reject and $\widehat{f}^{\geq \tau} \subseteq L$. Additionally, from Claim 4.2.11, with probability at least $1 - \frac{\delta}{2}$ it holds that

$$\forall T \in L : \Big|\alpha_T - \widehat{f}(T)\Big| \leq \lambda.$$

Hence, from the union bound, with probability at least $1-\delta$ all the assumptions of Claim B.7.1 hold, in which case Claim B.7.1 guarantees that $L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(\mathcal{H})+\varepsilon$, as desired.

- **Soundness.** Assume for contradiction that there exists some (possibly unbounded) prover P such that the verifier's output h satisfies

$$\mathbb{P}\Big[h \neq \text{reject} \;\wedge\; \Big(L_{\mathcal{D}}(h) > L_{\mathcal{D}}(\mathcal{H})+\varepsilon\Big)\Big] > \delta. \tag{4.4}$$

From the soundness property of the IGL protocol (Lemma 4.2.3),

$$\mathbb{P}\Big[h \neq \text{reject} \;\wedge\; \widehat{f}^{\geq\tau} \not\subseteq L\Big] \leq \frac{\delta}{2}. \tag{4.5}$$

Likewise, from Claim 4.2.11 and the union bound,

$$\mathbb{P}\Big[h \neq \text{reject} \;\wedge\; \exists T \in L: \;\Big|\alpha_T - \widehat{f}(T)\Big| > \lambda\Big] \leq \frac{\delta}{2}. \tag{4.6}$$

From Equations (4.4), (4.5) and (4.6), we obtain that

$$\mathbb{P}\Big[h \neq \text{reject} \;\wedge\; \Big(L_{\mathcal{D}}(h) > L_{\mathcal{D}}(\mathcal{H})+\varepsilon\Big) \;\wedge\; G\Big] > 0, \tag{4.7}$$

where G denotes the event in which $\widehat{f}^{\geq\tau} \subseteq L \;\wedge\; \forall T \in L: \;\Big|\alpha_T - \widehat{f}(T)\Big| \leq \lambda$. Claim B.7.1 asserts that

$$G \implies L_{\mathcal{D}}\Big(\sum_{T\in L}\alpha_T\chi_T(x)\Big) \leq L_{\mathcal{D}}(\mathcal{H})+\varepsilon. \tag{4.8}$$

Note that if $h \neq$ 'reject' then $h=\sum_{T\in L}\alpha_T\chi_T(x)$. Hence, putting together Equations (4.7) and (4.8), we conclude that

$$\mathbb{P}\Big[h \neq \text{reject} \;\wedge\; \Big(L_{\mathcal{D}}(h) > L_{\mathcal{D}}(\mathcal{H})+\varepsilon\Big) \;\wedge\; \Big(L_{\mathcal{D}}\,(h) \leq L_{\mathcal{D}}(\mathcal{H})+\varepsilon\Big)\Big] > 0,$$

which is a contradiction.

- **Double efficiency.** From Lemma 4.2.3, V uses at most

$$O\left(\frac{n}{\tau}\log\left(\frac{n}{\tau}\right)\log\left(\frac{1}{\delta}\right)\right) = O\left(\frac{nt}{\varepsilon}\log\left(\frac{nt}{\varepsilon}\right)\log\left(\frac{1}{\delta}\right)\right)$$

samples for the IGL protocol, which produces a set L of coefficients such that

$$|L| = O\left(\frac{n^2}{\tau^5}\log\left(\frac{1}{\delta}\right)\right) = O\left(\frac{n^2t^5}{\varepsilon^5}\log\left(\frac{1}{\delta}\right)\right).$$

Then, it uses

$$\left\lceil \frac{2\ln(2/\delta)}{\lambda^2} \right\rceil = O\left(\frac{\log(1/\delta)|L|}{\varepsilon}\right)$$

samples for estimating each of the coefficients. In total, V uses at most

$$O\left(\frac{nt}{\varepsilon}\log\left(\frac{nt}{\varepsilon}\right)\log\left(\frac{1}{\delta}\right)\right) + |L|\cdot O\left(\frac{2\log(1/\delta)|L|}{\varepsilon}\right) = \mathsf{poly}\Big(n, t, \frac{1}{\varepsilon}, \log\left(\frac{1}{\delta}\right)\Big)$$

random samples.

Also from Lemma 4.2.3, when executing the IGL protocol, the honest prover makes at most

$$O\left(\frac{n^3}{\tau^5}\log\left(\frac{1}{\delta}\right)\right) = O\left(\frac{n^3t^5}{\varepsilon^5}\log\left(\frac{1}{\delta}\right)\right) = \mathsf{poly}\Big(n, t, \frac{1}{\varepsilon}, \log\left(\frac{1}{\delta}\right)\Big)$$

queries.

Clearly, both parties run in time polynomial in the number of their samples or queries.

- **Sparsity.** The output $h = \sum_{T\in L}\alpha_T\chi_T$ is $|L|$-sparse, where

$$|L| = O\left(\frac{n^2}{\tau^5}\log\left(\frac{1}{\delta}\right)\right) = O\left(\frac{n^2t^5}{\varepsilon^5}\log\left(\frac{1}{\delta}\right)\right).$$

□

4.3 Separation Between Learning, Testing, and PAC Verification

In this section we present a gap in sample complexity between *learning* and *verification*. Conceptually, the result tells us that at least in some scenarios, delegating a learning task to an untrusted party is worthwhile, because verifying that their final result is correct is significantly cheaper than finding that result ourselves.

Recall from the discussion in Section 4.1 that when an untrusted prover provides a hypothesis $\tilde{h}$ which is allegedly ε-good, the straightforward approach for the verifier is to approximate each of the terms $L_{\mathcal{D}}(\tilde{h})$ and $L_{\mathcal{D}}(\mathcal{H})$, and then determine whether the inequality $L_{\mathcal{D}}(\tilde{h}) \leq L_{\mathcal{D}}(\mathcal{H}) + \varepsilon$ holds. From Hoeffding's inequality, the term $L_{\mathcal{D}}(\tilde{h})$ can easily be approximated with constant confidence up to any $O(\varepsilon)$ additive error using only $O(\frac{1}{\varepsilon^2})$ samples. However, approximating the term $L_{\mathcal{D}}(\mathcal{H})$ is more challenging, because it involves the loss values of all the hypotheses in the class $\mathcal{H}$.

In this section we show an MA-like proof system wherein the prover sends a single message $(\tilde{h}, \tilde{C}, \tilde{\ell})$ such that allegedly $\tilde{h}$ is an ε-good hypothesis with loss at most $\tilde{\ell} > 0$, and $\tilde{C} \in \{0,1\}^*$ is a string called a *certificate of loss*. The verifier operate as follows:[13]

[13] We provide a more detailed description of the verification procedure in Claim 4.3.15 below.

- Verify that $L_{\mathcal{D}}(\tilde{h}) \leq \tilde{\ell}$ with high probability. That is, estimate the loss of $\tilde{h}$ with respect to $\mathcal{D}$, and check that with high probability it is at most $\tilde{\ell}$.
- Use the certificate of loss $\tilde{C}$ to verify that with high probability, $L_{\mathcal{D}}(\mathcal{H}) \geq \tilde{\ell} - \varepsilon$. This step is called *verifying the certificate.*

That is, a certificate of loss is a string that helps the verifier ascertain that $\mathcal{H}$ has a large loss with respect to the unknown distribution $\mathcal{D}$. Whenever one defines algorithms for generating and verifying certificates of loss for a class $\mathcal{H}$, that also defines an associated single-message interactive proof system for PAC verifying $\mathcal{H}$.

Warm-Up: The Class of Thresholds

For clarity of exposition, we start with a warm-up that investigates the class $\mathcal{T}$ of threshold functions (see definition below). This class admits certificates that are easy to explain and visualize. We will show that the certificates of loss for $\mathcal{T}$ induce a proof system for PAC verifying $\mathcal{T}$ that is complete, sounds, and doubly efficient. However, verifying certificates for $\mathcal{T}$ requires as much resources as PAC learning $\mathcal{T}$ without the help of a prover, and so using this proof system to delegate learning of $\mathcal{T}$ is not worthwhile. Therefore, the next step (in Section 4.3 below) will show that $\mathcal{T}$ and its certification easily generalize to the class $\mathcal{T}_d$ of multi-thresholds. The gap between verifying and learning is demonstrated for $\mathcal{T}_d$.

Definition 4.3.1. *The class $\mathcal{T}$ is the set of all monotone increasing boolean functions on $[0, 1]$, as follows:*

$$\mathcal{T} = \{f_t : \ t \in [0, 1]\},$$

where for any $t \in [0, 1]$, the function $f_t : \ [0, 1] \to \{0, 1\}$ is given by

$$f_t(x) = \begin{cases} 0 & x < t \\ 1 & x \geq t. \end{cases}$$

Figure 4.2a illustrates an example of a function in $\mathcal{T}$.

Remark 4.3.2. *For convenience, we present the separation result with respect to thresholds defined over a continuous interval $\mathcal{X} \subseteq \mathbb{R}$. Furthermore, we assume that the marginal distribution on $\mathcal{X}$ is absolutely continuous with respect to the Lebesgue measure, and we also ignore issues relating to the representation of real numbers in computations and protocol messages. This provides for a smooth exposition of the ideas. In Appendix B.3, we show how the results can be discretized.*

Existence of Certificates of Loss for Thresholds

We want to design certificates such that for every distribution $\mathcal{D} \in \Delta([0, 1] \times \{0, 1\})$ the class $\mathcal{T}$ has large loss, $L_{\mathcal{D}}(\mathcal{T}) \geq \ell$, if and only if there exists a certificate for that fact.

The idea is straightforward. Consider two sets $A \subseteq [0,1] \times \{1\}$ and $B \subseteq [0,1] \times \{0\}$, such that all the points in A are located to the left of all the points in B, as in Figure 4.2b.

Because we only allow thresholds that are monotone increasing, a threshold that labels any point in A correctly must label all points of B incorrectly, and vice versa. Hence, any threshold must have loss at least $\min\{\mathcal{D}(A), \mathcal{D}(B)\}$. Formally:

Definition 4.3.3. *Let $\mathcal{D} \in \Delta([0,1] \times \{0,1\})$ be a distribution and $\ell, \eta \geq 0$. A certificate of loss at least ℓ for class $\mathcal{T}$ is a pair (a,b) where $0 < a \leq b < 1$.*

We say that the certificate is η-valid with respect to distribution $\mathcal{D}$ if the events

$$\begin{aligned} A &= [0,a) \times \{1\} \\ B &= [b,1] \times \{0\} \end{aligned} \tag{4.9}$$

satisfy

$$|\mathcal{D}(A) - \ell| + |\mathcal{D}(B) - \ell| \leq \eta. \tag{4.10}$$

The following claim shows the soundness of the certificate, i.e., that a valid certificate of loss does indeed entail that $L_{\mathcal{D}}(\mathcal{T})$ is large.

Claim 4.3.4. *Let $\mathcal{D} \in \Delta([0,1] \times \{0,1\})$ be a distribution and $\ell, \eta \geq 0$. If $\mathcal{D}$ has a certificate of loss at least ℓ which is η-valid with respect to $\mathcal{D}$, then $L_{\mathcal{D}}(\mathcal{T}) \geq \ell - \eta$.*

Proof. Assume $C = (a,b)$ is an η-valid certificate of loss at least ℓ for $\mathcal{T}$ with respect to $\mathcal{D}$. For any $t \in [0,1]$, we show that $L_{\mathcal{D}}(f_t) \geq \ell - \eta$.

Consider two cases:

- Case 1: $t < a$. Then for any $x \geq a$, $f_t(x) = 1$. In particular, taking B as in (4.9), we obtain that

  $$\forall (x,y) \in B: \; f_t(x) \neq y.$$

  Observe from Equation (4.10) that $\mathcal{D}(B) \geq \ell - \eta$. Therefore,

  $$L_{\mathcal{D}}(f_t) = \mathbb{P}_{(x,y)\in\mathcal{D}}[f_t(x) \neq y] \geq \mathcal{D}(B) \geq \ell - \eta.$$

- Case 2: $t \geq a$. This case is symmetric to the previous one, replacing B with $A = [0,a) \times \{1\}$.

□

Next, we show completeness, meaning that whenever $L_{\mathcal{D}}(\mathcal{T})$ is large there exists a certificate to that effect. However, the certificate is not tight, conceding a factor of 2:

Claim 4.3.5. *Let $\mathcal{D} \in \Delta([0,1] \times \{0,1\})$ be a distribution and $\ell \geq 0$. If $L_{\mathcal{D}}(\mathcal{T}) = \ell$ then there exists a 0-valid certificate of loss at least $\frac{\ell}{2}$ with respect to $\mathcal{D}$.*

Proof of Claim 4.3.5. Let f_t be an optimal threshold for $\mathcal{D}$, that is, $L_{\mathcal{D}}(f_t) = \ell$.[14] Let

$$\tilde{A} = [0,t) \times \{1\}$$

$$\tilde{B} = [t,1] \times \{0\}$$

denote the two events in which f_t misclassifies a point.[15] It follows that

$$\ell = \mathcal{D}(\tilde{A}) + \mathcal{D}(\tilde{B}).$$

If $\mathcal{D}(\tilde{A}) = \mathcal{D}(\tilde{B}) = \frac{\ell}{2}$, then (t,t) is the desired certificate. Otherwise, assume w.l.o.g. that

$$\mathcal{D}(\tilde{A}) > \frac{\ell}{2} > \mathcal{D}(\tilde{B}).$$

Because the marginal distribution of $\mathcal{D}$ on $[0,1]$ is absolutely continuous, there exists a point $a \in [0,t)$ that partitions the event $\tilde{A}$ to

$$A := [0,a) \times \{1\},$$

$$A' := [a,t) \times \{1\},$$

such that $\mathcal{D}(A) = \frac{\ell}{2}$. Considering the event $B' := [a,t) \times \{0\}$. The optimality of f_t implies that

$$\mathcal{D}(B') \geq \mathcal{D}(A')$$

because otherwise the threshold f_a would have loss strictly smaller than that of f_t.

Notice that

$$\mathcal{D}(B') \geq \mathcal{D}(A') = \mathcal{D}(\tilde{A}) - \mathcal{D}(A) = \Big(\ell - \mathcal{D}(\tilde{B})\Big) - \mathcal{D}(A) = \ell - \mathcal{D}(\tilde{B}) - \frac{\ell}{2} = \frac{\ell}{2} - \mathcal{D}(\tilde{B}).$$

Hence, again invoking absolute continuity of measure as above, there exists a point $b \in [a,t)$ such that

$$\mathcal{D}([b,t) \times \{0\}) = \frac{\ell}{2} - \mathcal{D}(\tilde{B}).$$

Therefore, taking

$$B := [b,1) \times \{0\}$$

yields

$$\mathcal{D}(B) = \mathcal{D}([b,t) \times \{0\}) + \mathcal{D}(\tilde{B}) = \frac{\ell}{2}.$$

So (a,b) is the desired certificate. □

[14] Note that an optimal threshold $t \in [0,1]$ exists because $[0,1]$ is compact, and the mapping $t \mapsto L_{\mathcal{D}}(f_t)$ is continuous.

[15] Namely, $\tilde{A}$ is the event in which a point has label 1, but f_t assigns label 0 to it, and $\tilde{B}$ is the event in which a point has label 0, but f_t assigns label 1 to it.

Efficient Generation and Verification of Certificates for Thresholds

The following two claims show that certificates of loss for $\mathcal{T}$ do not merely exist, but they can be generated and verified efficiently, making delegation feasible.

Claim 4.3.6 (Efficient Verification). *Let $\mathcal{D} \in \Delta([0,1] \times \{0,1\})$ be a distribution and $\ell, \delta, \eta \geq 0$. There exists an algorithm that, upon receiving input (a, b) such that $0 < a \leq b < 1$, takes $O\left(\frac{\log\left(\frac{1}{\delta}\right)}{\eta^2}\right)$ i.i.d. samples from $\mathcal{D}$ and satisfies the following:*

- *Completeness. If (a, b) is an η-valid certificate of loss at least ℓ with respect to $\mathcal{D}$, then the algorithm accepts with probability at least $1 - \delta$.*
- *Soundness. If (a, b) is not a 2η-valid certificate of loss at least ℓ with respect to $\mathcal{D}$, then the algorithm rejects with probability at least $1 - \delta$.*

Furthermore, the algorithm runs in time polynomial[16] in the number of samples.

Proof. Let A, B be as in Equation (4.9), and let $(x_1, y_1), \dots, (x_m, y_m)$ be the samples the algorithm received. The algorithm calculates the empirical measures of A, B by

$$\widehat{\ell}_A := \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\Big((x_i, y_i) \in A\Big)$$

$$\widehat{\ell}_B := \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\Big((x_i, y_i) \in B\Big)$$

and accepts if and only if

$$|\widehat{\ell}_A - \ell| + |\widehat{\ell}_B - \ell| < \frac{3}{2}\eta.$$

The running time is clear, and correctness follows from Hoeffding's inequality,

$$\mathbb{P}\left[\left|\widehat{\ell}_A - \mathcal{D}(A)\right| \geq \frac{\eta}{4}\right] \leq 2\exp\left(-2m\left(\frac{\eta}{4}\right)^2\right).$$

Requiring that this probability be strictly less than $\frac{\delta}{2}$ yields the bound

$$m > \frac{2\log\frac{16}{\delta}}{\eta^2}.$$

The same holds for $\widehat{\ell}_B$. The union bound entails that with probability at least $1 - \delta$ both estimates are $\frac{\eta}{4}$-close to their expectations, in which case the algorithm decides correctly. □

[16] Recall that we ignore the cost performing calculations with real numbers.

Claim 4.3.7 (**Efficient Generation**). *There exists an algorithm as follows. For any distribution $\mathcal{D} \in \Delta([0,1]\times\{0,1\})$ and any $\delta, \eta \in (0, \frac{1}{2})$, the algorithm outputs a certificate $(\widehat{a}, \widehat{b})$ for $\mathcal{T}$ that with probability at least $1-\delta$ is an η-valid certificate of loss at least $\ell = L_{\mathcal{D}}(\mathcal{T})/2$ with respect to $\mathcal{D}$. The algorithm uses*

$$O\left(\frac{1}{\eta^2}\log\frac{1}{\eta} + \frac{1}{\eta^2}\log\frac{1}{\delta}\right)$$

i.i.d. samples from $\mathcal{D}$ and runs in time polynomial in the number of samples.

Proof. The proof is a standard application of uniform convergence, VC dimension and empirical risk minimization (ERM), as covered e.g. in Shalev-Shwartz and Ben-David (2014). For completeness, we provide a self-contained proof that depends only on Theorem B.4.3, which upper bounds the number of samples necessary to obtain an ϵ-*sample* for a set system of finite VC dimension (see definitions in Appendix B.4).

We start by stating the following consequence of Theorem B.4.3. Let

$$S = ((x_1, y_1), \dots, (x_m, y_m))$$

denote the samples that the algorithm receives, and let $\mathcal{I}$ denote the following set of intervals:

$$\mathcal{I} = \{[u,v) : \ u, v \in \mathbb{R}\} \cup \{[u,v] : \ u, v \in \mathbb{R}\}.$$

Observe that the set system $\mathcal{A} = (\mathbb{R}\times\{0,1\}, \mathcal{I}\times\{0,1\})$ has VC dimension 2. Hence, from Theorem B.4.3, with probability at least $1-\delta$, we have that S is an η'-sample for $\mathcal{A}$ with respect to $\mathcal{D}$, where $\eta' := \frac{\eta}{16}$.

The algorithm operates in two steps. In the first step, the algorithm estimates ℓ. For any $t \in \mathbb{R}$, denote by $L_S(f_t)$ the empirical loss of f_t, namely

$$L_S(f_t) := L_S^{\text{left}}(f_t) + L_S^{\text{right}}(f_t)$$

for

$$L_S^{\text{left}}(f_t) := \frac{|([0,t)\times\{1\})\cap S|}{|S|}$$

and

$$L_S^{\text{right}}(f_t) := \frac{|([t,1]\times\{0\})\cap S|}{|S|}.$$

(Cardinalities are computed with S viewed as a multiset.)

The algorithm uses the sample S to find the threshold $f_{\widehat{t}} \in \mathcal{T}$ defined by

$$\widehat{t} := \operatorname*{arg\,min}_{t\in X} L_S(f_t),$$

where $X = \{x_1, \dots, x_m, 1\}$.

The algorithm estimates ℓ by taking

$$\widehat{\ell} := L_S(f_{\widehat{t}})/2 + 3\eta'.$$

We argue that $\widehat{\ell}$ is a good estimate whenever S is an η'-sample: Let $f^* = \arg\min_{f\in\mathcal{T}} L_{\mathcal{D}}(f)$. If S is an η'-sample then

$$\begin{aligned} L_{\mathcal{D}}(f_{\widehat{t}}) &\leq L_S(f_{\widehat{t}}) + 2\eta' \\ &= \min_{t\in X} L_S(f_t) + 2\eta' \\ &= \min_{t\in \mathbb{R}} L_S(f_t) + 2\eta' \\ &\leq L_S(f^*) + 2\eta' \\ &\leq L_{\mathcal{D}}(f^*) + 4\eta'. \end{aligned}$$

Therefore,

$$\begin{aligned} \left|L_S(f_{\widehat{t}}) - L_{\mathcal{D}}(f^*)\right| &\leq \left|L_S(f_{\widehat{t}}) - L_{\mathcal{D}}(f_{\widehat{t}})\right| + \left|L_{\mathcal{D}}(f_{\widehat{t}}) - L_{\mathcal{D}}(f^*)\right| \\ &\leq 2\eta' + 4\eta' = 6\eta'. \end{aligned}$$

Thus, the estimate $\widehat{\ell}$ satisfies

$$|\widehat{\ell} - \ell| = \left|\frac{L_S(f_{\widehat{t}})}{2} + 3\eta' - \frac{L_{\mathcal{D}}(f^*)}{2}\right| \leq 3\eta' + \frac{\left|L_S(f_{\widehat{t}}) - L_{\mathcal{D}}(f^*)\right|}{2} \leq 6\eta'.$$

Furthermore,

$$\begin{aligned} \widehat{\ell} &= \frac{L_S(f_{\widehat{t}})}{2} + 3\eta' \\ &\geq \frac{L_{\mathcal{D}}(f^*)}{2} - \frac{\left|L_S(f_{\widehat{t}}) - L_{\mathcal{D}}(f^*)\right|}{2} + 3\eta' \\ &\geq \frac{L_{\mathcal{D}}(f^*)}{2} = \ell. \end{aligned}$$

This completes the first step.

In the second step, the algorithm calculates

$$(\widehat{a}, \widehat{b}) := \underset{a',b'\in X:\ a'\leq b'}{\arg\min} \left|L_S^{\text{left}}(f_{a'}) - \widehat{\ell}\right| + \left|L_S^{\text{right}}(f_{b'}) - \widehat{\ell}\right|.$$

We claim that $(\widehat{a}, \widehat{b})$ is an η-valid certificate of loss $\widehat{\ell}$. From Claim 4.3.5 and the assumption that $\mathcal{D}$ is absolutely continuous, there exist (a, b) constituting a 0-valid certificate of loss exactly ℓ.

Denote

$$\widehat{A} = [0, \widehat{a}) \times \{1\}, \qquad A = [0, a) \times \{1\}$$

$$\widehat{B} = [\widehat{b}, 1] \times \{0\}, \qquad B = [b, 1] \times \{0\}.$$

Then

$$\begin{aligned}
|\mathcal{D}(\widehat{A}) - \widehat{\ell}| + |\mathcal{D}(\widehat{B}) - \widehat{\ell}| &\leq |\mathcal{D}(\widehat{A}) - L_S^{\text{left}}(f_{\widehat{a}})| + |L_S^{\text{left}}(f_{\widehat{a}}) - \widehat{\ell}| \\
&\quad + |\mathcal{D}(\widehat{B}) - L_S^{\text{right}}\left(f_{\widehat{b}}\right)| + |L_S^{\text{right}}\left(f_{\widehat{b}}\right) - \widehat{\ell}| \\
&\leq |L_S^{\text{left}}(f_{\widehat{a}}) - \widehat{\ell}| + |L_S^{\text{right}}\left(f_{\widehat{b}}\right) - \widehat{\ell}| + 2\eta' \\
&= \min_{\widehat{a}, \widehat{b} \in X:\ \widehat{a} \leq \widehat{b}} \left| L_S^{\text{left}}(f_{\widehat{a}}) - \widehat{\ell} \right| + \left| L_S^{\text{right}}\left(f_{\widehat{b}}\right) - \widehat{\ell} \right| + 2\eta' \\
&= \min_{\widehat{a}, \widehat{b} \in \mathbb{R}:\ \widehat{a} \leq \widehat{b}} \left| L_S^{\text{left}}(f_{\widehat{a}}) - \widehat{\ell} \right| + \left| L_S^{\text{right}}\left(f_{\widehat{b}}\right) - \widehat{\ell} \right| + 2\eta' \\
&\leq \left| L_S^{\text{left}}(f_a) - \widehat{\ell} \right| + \left| L_S^{\text{right}}(f_b) - \widehat{\ell} \right| + 2\eta' \\
&\leq \left| L_S^{\text{left}}(f_a) - \ell \right| + \left| L_S^{\text{right}}(f_b) - \ell \right| + 14\eta' \\
&= \left| L_S^{\text{left}}(f_a) - \mathcal{D}(A) \right| + \left| L_S^{\text{left}}(f_b) - \mathcal{D}(B) \right| + 14\eta' \\
&\leq \eta' + \eta' + 14\eta' = \eta.
\end{aligned}$$

We conclude that $(\widehat{a}, \widehat{b})$ is an η-valid certificate of loss at least ℓ, provided that S is an η'-sample with respect to $\mathcal{D}$, which happens with probability at least $1 - \delta$. Seeing as the algorithm runs in time polynomial in the number of samples, the proof is complete. □

Warm-Up Summary

We explained how certificates of loss induce a proof system for PAC verification, and described a specific instance of this for the class $\mathcal{T}$ of threshold functions. We saw that the honest prover is able to generate a message $(\tilde{h}, \tilde{C}, \tilde{\ell})$ that is accepted by the verifier. If $\tilde{h}$ has loss greater than double the true loss, no certificate can convince the verifier to accept $\tilde{h}$. Both the verifier and the honest prover are efficient. The certificate is not tight; if the true loss is $\ell = L_{\mathcal{D}}(\mathcal{T})$, the certificate of loss only proves that the loss is at least $\frac{\ell}{2}$.

However, the example of the class $\mathcal{T}$ is lacking an essential ingredient. The sample complexity used by the verifier is the same as is necessary for learning without a prover, and so delegation is not beneficial. In the next section, we present a generalization of this class, where there is a substantial gap between the resources necessary for verification and those required for learning, making it worthwhile to delegate the learning task to an untrusted prover.

Efficient PAC Verification for the Class $\mathcal{T}_d$ of Multi-Thresholds

In the warm-up we saw certificates of loss that induce a proof system for PAC verification for the class of thresholds $\mathcal{T}$. We now extend this construction to a class $\mathcal{T}_d$ of multi-thresholds, construct a PAC verification proof system for $\mathcal{T}_d$ that obtains the following sample complexity separation between PAC verification on the one hand and PAC learning and tolerant testing or distance approximation on the other hand.

Theorem 4.3.8. *There exists a sequence of classes of functions*

$$\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \ldots \subseteq \{0,1\}^{\mathbb{R}}$$

such that for any fixed $\varepsilon, \delta \in (0, \frac{1}{2})$ all of the following hold:

(i) $\mathcal{T}_d$ is proper 2-PAC verifiable, where the verifier uses[17]

$$m_V = O\left(\frac{\sqrt{d}\log(d)\log\left(\frac{1}{\delta}\right)}{\varepsilon^6}\right)$$

random samples, the honest prover uses

$$m_P = O\left(\frac{d^3\log^2(d)}{\varepsilon^4}\log\left(\frac{d}{\varepsilon}\right) + \frac{d\sqrt{d}\log(d)}{\varepsilon^2}\log\left(\frac{1}{\delta}\right)\right)$$

random samples, and each of them runs in time polynomial in its number of samples.[18]

(ii) Agnostic PAC learning $\mathcal{T}_d$ requires $\Omega\left(\frac{d+\log(\frac{1}{\delta})}{\varepsilon^2}\right)$ samples.

(iii) If $\varepsilon \le \frac{1}{32}$ then 2-PAC learning the class $\mathcal{T}_d$ requires $\Omega\left(\frac{d}{\log(d)}\right)$ samples. This is true even if we assume that $L_{\mathcal{D}}(\mathcal{T}_d) > 0$, where $\mathcal{D}$ is the underlying distribution.

(iv) Testing whether $L_{\mathcal{D}}(\mathcal{T}_d) \le \alpha$ or $L_{\mathcal{D}}(\mathcal{T}_d) \ge \beta$ for any $0 < \alpha < \beta < \frac{1}{2}$ with success probability at least $1-\delta$ when $\mathcal{D}$ is an unknown distribution (without the help of a prover) requires $\Omega\left(\frac{d}{\log(d)}\right)$ random samples from $\mathcal{D}$.

The Class $\mathcal{T}_d$

We start by defining the class of multi-thresholds.

Definition 4.3.9. *For any $d \in \mathbb{N}$, denote by $\mathcal{T}_d$ the class of functions*

$$\mathcal{T}_d = \{f_{t_1,\ldots,t_d} : \ t_1, \ldots, t_d \in \mathbb{R}\}$$

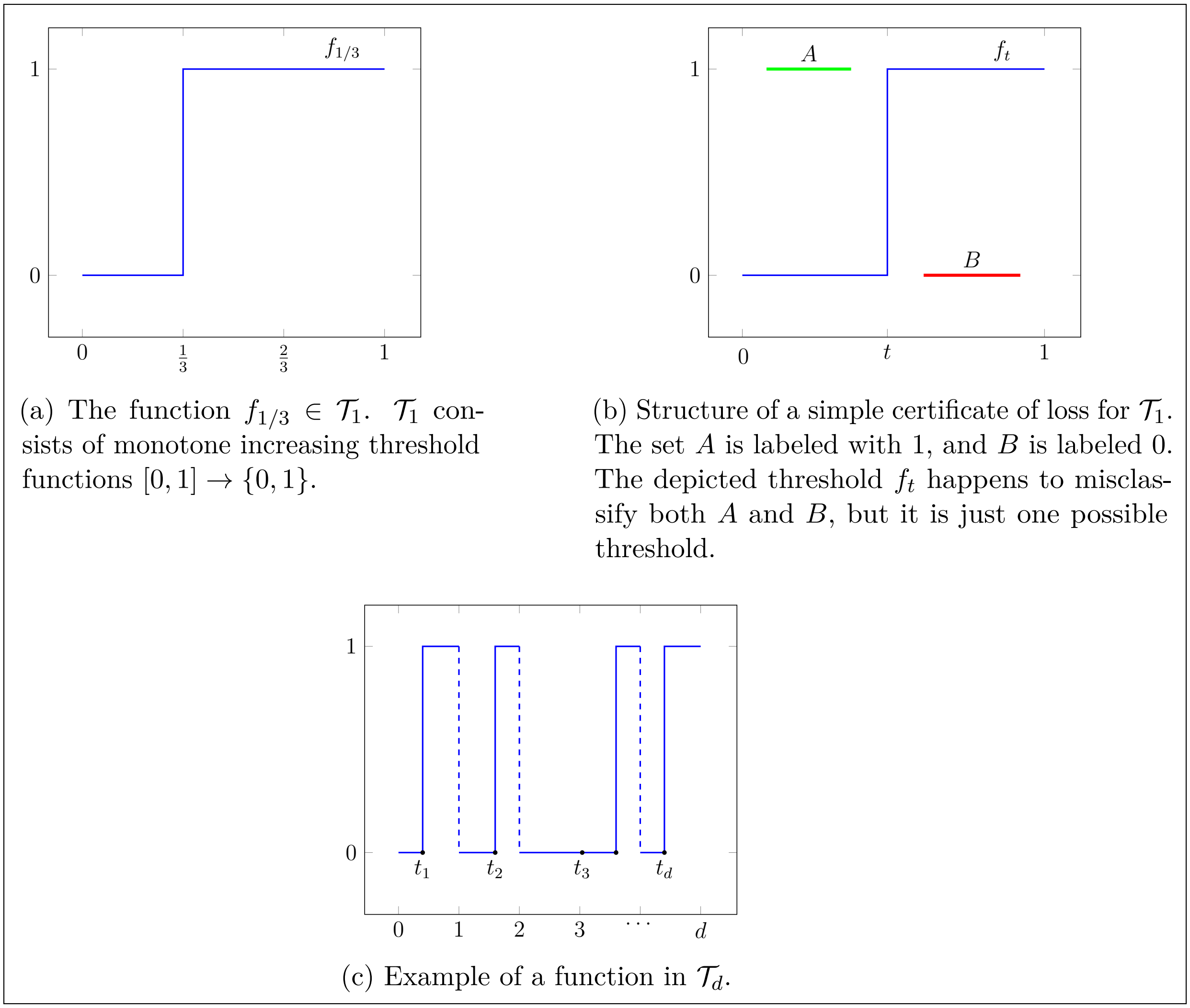
[17] We believe that the dependence of m_V on ε can be improved, see Remark 4.3.16.

[18] In Chapter 5, we strengthen this to obtain 1-PAC verification with better sample complexity bounds.

where for all $t_1, \dots, t_d \in \mathbb{R}$ and $x \in [0, d]$, the function $f_{t_1,\dots,t_d} : \mathbb{R} \to \{0, 1\}$ is given by

$$f_{t_1,\dots,t_d}(x) = \begin{cases} 0 & x < t_{\lceil x \rceil} \\ 1 & x \geq t_{\lceil x \rceil}, \end{cases}$$

and $f_{t_1,\dots,t_d}$ vanishes on the complement of $[0, d]$.



(a) The function $f_{1/3} \in \mathcal{T}_1$. $\mathcal{T}_1$ consists of monotone increasing threshold functions $[0, 1] \to \{0, 1\}$.

(b) Structure of a simple certificate of loss for $\mathcal{T}_1$. The set A is labeled with 1, and B is labeled 0. The depicted threshold f_t happens to misclassify both A and B, but it is just one possible threshold.

(c) Example of a function in $\mathcal{T}_d$.

Figure 4.2: The class $\mathcal{T}_d$ of multi-thresholds, with the special case $\mathcal{T}_1$ and its certificate structure.

Existence of Certificates of Loss for $\mathcal{T}_d$

Remark 4.3.10. *As before, we present the separation result with respect to functions defined over $\mathbb{R}$, we assume that the marginal distribution of the samples on $\mathbb{R}$ is absolutely continuous* *with respect to the Lebesgue measure, and we ignore issues relating to the representation of real numbers in computations and protocol messages. This provides for a smoother exposition of the ideas. In Appendix B.3, we show how the results can be discretized.*

For each $i \in [d]$, the class $\mathcal{T}_d$ restricted to $[i-1, i]$ is a shifted copy of the class $\mathcal{T}$. Hence, exactly as we did for $\mathcal{T}$, we can construct a certificate of loss which proves that $\mathcal{T}_d$ must have loss ℓ_i within the interval $[i-1, i]$. Therefore, we define certificates for $\mathcal{T}_d$ as collections of d certificates of loss for $\mathcal{T}$.

Definition 4.3.11. *Let $\mathcal{D} \in \Delta(\mathbb{R} \times \{0,1\})$ be a distribution and $\ell, \eta \geq 0$. A certificate of loss at least ℓ for the class $\mathcal{T}_d$ is a tuple*

$$(C_1, \ell_1, C_2, \ell_2 \ldots, C_d, \ell_d)$$

where for all $i \in [d]$:

- $C_i = (a_i, b_i)$,
- $i - 1 < a_i \leq b_i \leq i$,
- $\ell_i \geq 0$, *and*

$$\sum_{i=1}^{d} \ell_i = \ell.$$

The certificate is η-valid with respect to $\mathcal{D}$ if the events

$$A_i = [i-1, a_i) \times \{1\}$$

$$B_i = [b_i, i] \times \{0\}$$

defined for all $i \in [d]$ satisfy

$$\sum_{i=1}^{d} |\mathcal{D}(A_i) - \ell_i| + |\mathcal{D}(B_i) - \ell_i| \leq \eta.$$

The following analogs of Claims 4.3.4 and 4.3.5 follow similarly.

Claim 4.3.12. *Let $\mathcal{D} \in \Delta(\mathbb{R} \times \{0,1\})$ be a distribution and $\ell, \eta \geq 0$. If $\mathcal{D}$ has a certificate of loss at least ℓ for $\mathcal{T}_d$ that is η-valid with respect to $\mathcal{D}$, then every function in $\mathcal{T}_d$ must have loss at least $\ell - \eta$ with respect to $\mathcal{D}$.*

Claim 4.3.13. *Let $\mathcal{D} \in \Delta(\mathbb{R} \times \{0,1\})$ be a distribution and $\ell \geq 0$. If $L_{\mathcal{D}}(\mathcal{T}_d) = \ell$ then there exists a 0-valid certificate of loss at least $\frac{\ell}{2}$ for $\mathcal{T}_d$ with respect to $\mathcal{D}$.*

Efficient Generation and Verification of Certificates for $\mathcal{T}_d$

The following is a straightforward analogue of Claim 4.3.7.

Claim 4.3.14 (**Efficient Generation**). *There exists an algorithm as follows. For any distribution $\mathcal{D} \in \Delta([0,1] \times \{0,1\})$ and any $\delta, \eta \in (0, \frac{1}{2})$, the algorithm outputs a certificate of loss for $\mathcal{T}_d$ that with probability at least $1-\delta$ is an η-valid certificate of loss at least $\ell = L_{\mathcal{D}}(\mathcal{T}_d)/2$ with respect to $\mathcal{D}$. The algorithm uses*

$$O\left(\tfrac{d^2}{\eta^2}\log\tfrac{d}{\eta} + \tfrac{d^2}{\eta^2}\log\tfrac{1}{\delta}\right)$$

i.i.d. samples from $\mathcal{D}$ and runs in time polynomial in the number of samples.

Proof sketch. The proof follows the same lines as for Claim 4.3.7. Recall that in that proof, the algorithm takes a sample of size $O\left(\frac{1}{\eta^2}\log\frac{1}{\eta} + \frac{1}{\eta^2}\log\frac{1}{\delta}\right)$. Whenever the sample is an η'-sample with respect to the set system $\mathcal{A}$ defined in that proof, the algorithm is able to generate a certificate that is η-valid.

Here, the algorithm instead takes a sample that with probability at least $1-\delta$ is an $\frac{\eta'}{d}$-sample with respect to $\mathcal{A}$. This leads to the sample size mentioned in the statement. The algorithm proceeds as in the previous case, using the sample to generate d certificates of loss, one for each interval of the form $[i-1, i]$ for $i \in [d]$. Whenever the sample is an $\frac{\eta'}{d}$-sample, each of these certificates will be $\frac{\eta}{d}$-valid. Combining these certificates together yields a certificate for $\mathcal{T}_d$ that is η-valid. □

Agnostic PAC learning $\mathcal{T}_d$ requires

$$\Theta\left(\frac{d + \log(\frac{1}{\delta})}{\varepsilon^2}\right)$$

samples, because its VC dimension is d. Thus, the certificate generation procedure outlined above requires that the prover use a larger number of samples than what is necessary for learning. This may be worthwhile, because, as stated in the following claim, the verifier can verify the certificate using fewer samples than what is required for learning.

Claim 4.3.15 (**Efficient verification**). *Let $d \in \mathbb{N}$ and $\lambda \in (0,1)$. Let $C = (C_1, \ell_1, \dots, C_d, \ell_d)$ be a certificate of loss ℓ for $\mathcal{T}_d$, and let $\mathcal{D}$ be a distribution. There exists an algorithm that takes*

$$m = O\left(\log\left(\frac{1}{\delta}\right)\frac{\sqrt{d}}{\lambda^6}\log(d)\right)$$

samples from $\mathcal{D}$, and satisfies:

- ***Completeness.*** *Let*

$$\lambda' := \frac{\lambda^3}{300\sqrt{d}\log d}.$$

If C is λ'-valid with respect to $\mathcal{D}$, then the algorithm accepts with probability at least $1-\delta$.

- ***Soundness.*** *If C is not 2λ-valid with respect to $\mathcal{D}$, then the algorithm rejects with probability at least $1-\delta$.*

Remark 4.3.16. *We believe that the parameters in the above claim can be improved such that λ^6 is replaced by λ^2 in the sample complexity, and λ^3 is replaced by λ in the completeness parameter λ'. This would be achieved by using an ℓ_2-based uniformity tester, together with the reduction of Goldreich (2020).*

Proof. The proof uses ideas from distribution identity testing stated in Corollary B.5.2. For all $i \in [d]$, let

$$A_i = [i-1, a_i) \times \{1\}, \text{ and}$$
$$B_i = [b_i, i] \times \{0\}.$$

The algorithm is required to decide whether the validity v of the certificate is less than λ', i.e., whether

$$v := \sum_{i=1}^{d} |\mathcal{D}(A_i) - \ell_i| + |\mathcal{D}(B_i) - \ell_i| \leq \lambda',$$

or whether $v > 2\lambda$.

Form the partition $R := \{A_1, B_1, \dots, A_d, B_d, E\}$ of $\mathbb{R} \times \{0,1\}$, where

$$E = (\mathbb{R} \times \{0,1\}) \setminus \Big(\bigcup_{i\in[d]} A_i \cup B_i \Big).$$

Define two probability functions, $\mathcal{D}_R$ and $\mathcal{D}^*$, both over this finite set R of cardinality $2d+1$. Let $\mathcal{D}_R$ be the distribution induced on R by $\mathcal{D}$; namely, $\mathcal{D}_R(r) = \mathcal{D}(r)$ for each $r \in R$. Let $\mathcal{D}^*$ denote the distribution over R corresponding to the certificate C. Namely, $\mathcal{D}^*(A_i) = \mathcal{D}^*(B_i) = \ell_i$ for all $i \in [d]$, and $\mathcal{D}^*(E) = 1 - 2\sum_{i=1}^{d} \ell_i = 1 - 2\ell$.

Consider the mapping M_R that sends each point to the member of R it belongs to:

$$M_R(x,y) = \begin{cases} A_i & (x,y) \in A_i, \\ B_i & (x,y) \in B_i, \\ E & \text{otherwise.} \end{cases}$$

Observe that if $S = \big((x_1,y_1),\dots,(x_m,y_m)\big)$ is sampled i.i.d. from $\mathcal{D}$, then

$$M_R(S) := (M_R(x_1,y_1),\dots,M_R(x_m,y_m))$$

is an i.i.d. sample from $\mathcal{D}_R$. Observe the following connection between $\mathsf{TV}(\mathcal{D}_R,\mathcal{D}^*)$ and the validity v of the certificate:

$$\begin{aligned} v &= \sum_{i=1}^{d} |\mathcal{D}(A_i) - \ell_i| + |\mathcal{D}(B_i) - \ell_i| \\ &= \sum_{i=1}^{d} |\mathcal{D}_R(A_i) - \mathcal{D}^*(A_i)| + |\mathcal{D}_R(B_i) - \mathcal{D}^*(B_i)| \\ &= 2\mathsf{TV}(\mathcal{D}_R,\mathcal{D}^*) - |\mathcal{D}_R(E) - \mathcal{D}^*(E)|\,. \end{aligned}$$

Furthermore,

$$|\mathcal{D}_R(E) - \mathcal{D}^*(E)| \leq \mathsf{TV}(\mathcal{D}_R, \mathcal{D}^*).$$

Thus,

$$\mathsf{TV}(\mathcal{D}_R, \mathcal{D}^*) \leq v \leq 2\mathsf{TV}(\mathcal{D}_R, \mathcal{D}^*).$$

The algorithm operates as follows. It executes the distribution identity test stated in Corollary B.5.2 with respect to distribution $\mathcal{D}^*$ and the sample $M_R(S)$. Because $\mathcal{D}^*$ is a distribution over a set of size $2d+1$, taking a sample $M_R(S)$ of size m as specified in the statement is sufficient to ensure that with probability at least $1-\delta$, the test distinguishes correctly between the case $\mathsf{TV}(\mathcal{D}_R, \mathcal{D}^*) \leq \lambda'$ and the case $\mathsf{TV}(\mathcal{D}_R, \mathcal{D}^*) \geq \lambda$. The algorithm accepts the certificate if and only if the test concludes that $\mathsf{TV}(\mathcal{D}_R, \mathcal{D}^*) \leq \lambda'$.
The desired properties hold:

- Completeness. If $v \leq \lambda'$, then $\mathsf{TV}(\mathcal{D}_R, \mathcal{D}^*) \leq v \leq \lambda'$, and so with probability at least $1-\delta$ the algorithm accepts.
- Soundness. If $v > 2\lambda$, then $\lambda < \frac{v}{2} \leq \mathsf{TV}(\mathcal{D}_R, \mathcal{D}^*)$, and so with probability at least $1-\delta$ the algorithm rejects.

This concludes the proof. □

We now use the previous two claims to construct the efficient PAC verification protocol for '$\mathcal{T}_d$.

Claim 4.3.17. *$\mathcal{T}_d$ is 2-PAC verifiable with sample and runtime complexities as in part (i) of Theorem 4.3.8.*

Proof. The interactive proof system for 2-PAC verification operates as follows. Let $\mathcal{D} \in \Delta(\mathbb{R} \times \{0,1\})$, and let $\ell = L_{\mathcal{D}}(\mathcal{T}_d)$.

1. The honest prover learns a function $\tilde{h} \in \mathcal{T}_d$ that has loss at most $\ell + \frac{\varepsilon}{6}$, with probability at least $1 - \frac{\delta}{4}$. This can be done with the required sample complexity, and the computation runs in time polynomial in the number of samples, because an ERM can be computed in polynomial time (as discussed in the proof of Claim 4.3.7).

2. From Claim 4.3.13, there exists a 0-valid certificate of loss at least $\frac{\ell}{2}$ for $\mathcal{T}_d$ with respect to $\mathcal{D}$, where $\ell = L_{\mathcal{D}}(\mathcal{T}_d)$. From Claim 4.3.14, the honest prover can generate a certificate $\tilde{C} = (C_1, \ell_1, \ldots, C_d, \ell_d)$ of loss $\tilde{\ell} := \sum_i \ell_i \geq \frac{\ell}{2}$ that with probability at least $1 - \frac{\delta}{4}$ is η-valid, for

$$\eta = \frac{(\varepsilon/8)^2}{300\sqrt{d}\log(d)}.$$

The prover can do this using m_P samples as in the statement.

3. The honest prover sends $(\tilde{h}, \tilde{C}, \tilde{\ell})$ to the verifier V.

4. The verifier V uses $O\left(\log\left(\frac{1}{\delta}\right)/\varepsilon^2\right)$ samples to estimate the loss $L_{\mathcal{D}}(\tilde{h})$ up to an additive error of $\frac{\varepsilon}{6}$ with confidence at least $1-\frac{\delta}{4}$, and rejects if the estimate is greater than $2\tilde{\ell}+\frac{\varepsilon}{3}$. This ensures that V accepts only if $L_{\mathcal{D}}(\tilde{h}) \leq 2\tilde{\ell}+\frac{\varepsilon}{2}$.

5. From Claim 4.3.15, the verifier can use m_V samples to verify $\tilde{C}$, such that if $\tilde{C}$ is η-valid then V accepts with probability at least $1-\frac{\delta}{4}$, and if $\tilde{C}$ is not $\frac{\varepsilon}{4}$-valid, then V rejects with probability at least $1-\frac{\delta}{4}$.

For the completeness, observe that when interacting with the honest prover, each of the operations in Steps 1, 2, 4 and 5 succeeds with probability at least $1-\frac{\delta}{4}$, and so with probability at least $1-\delta$ they all succeed and V accepts $\tilde{h}$, which has loss at most $\ell+\frac{\varepsilon}{6}$.

For soundness, let $H \in \mathcal{T}_d \cup \{\text{reject}\}$ denote the output of V, and let

$$B = \{h \in \mathcal{T}_d : \ L_{\mathcal{D}}(h) > 2\ell + \varepsilon\}.$$

Assume towards a contradiction there exists a prover P for which $\mathbb{P}[H \in B] > \delta$. Let W denote the message $(\tilde{h}, \tilde{C}, \tilde{\ell})$ sent by P. Because

$$\delta < \mathbb{P}[H \in B] = \sum_{w} \mathbb{P}[H \in B \mid W = w]\mathbb{P}[W = w],$$

there exists some $w_0 = (\tilde{h}_0, \tilde{C}_0, \tilde{\ell}_0)$ such that

$$\mathbb{P}[H \in B \mid W = w_0] > \delta. \tag{4.11}$$

When the verifier V does not reject, V outputs the hypothesis sent by P. Thus, $\tilde{h}_0 \in B$ and yet V accepts w_0 with probability $> \delta$. We show that this is impossible, based on the following two facts:

- If $L_{\mathcal{D}}(\tilde{h}_0) > 2\tilde{\ell}+\frac{\varepsilon}{2}$, then from Step 4, the verifier V accepts w_0 with probability at most $\frac{\delta}{4}$.
- If $\tilde{C}_0$ is not an $\frac{\varepsilon}{4}$-valid certificate of loss $\tilde{\ell}$, then from Step 5, the verifier V accepts w_0 with probability at most $\frac{\delta}{4}$.

This implies that $\tilde{h}_0 \in B$, that $L_{\mathcal{D}}(\tilde{h}_0) \leq 2\tilde{\ell}+\frac{\varepsilon}{2}$ and that $\tilde{C}_0$ is an $\frac{\varepsilon}{4}$-valid certificate of loss $\tilde{\ell}$. Claim 4.3.12 yields the contradiction:

$$\ell = L_{\mathcal{D}}(\mathcal{T}_d) \geq \tilde{\ell} - \frac{\varepsilon}{4} \geq \frac{L_{\mathcal{D}}(\tilde{h}_0)}{2} - \frac{\varepsilon}{2} > \ell.$$

□

Lower Bounds for Closeness Testing and 2-PAC Learning of the Class $\mathcal{T}_d$

In this section we show near-linear lower bounds for testing closeness and 2-PAC learning of the class $\mathcal{T}_d$.

Definition 4.3.18. *Let $0 < \alpha < \beta < 1$ and $d \in \mathbb{N}$. The* <u>*(α, β, d)-threshold closeness testing problem*</u> *is the following promise problem. Given sample access to an unknown distribution $\mathcal{D} \in \Delta([0, d] \times \{0, 1\})$, distinguish between the following two cases:*

(i) $L_{\mathcal{D}}(\mathcal{T}_d) \leq \alpha$.

(ii) $L_{\mathcal{D}}(\mathcal{T}_d) \geq \beta$.

Lemma 4.3.19. *Fix $0 < \alpha < \beta < \frac{1}{2}$. Any tester that uses sample access to an unknown distribution $\mathcal{D} \in \Delta([0, d] \times \{0, 1\})$ and solves the (α, β, d)-threshold closeness testing problem correctly with probability at least $\frac{2}{3}$ for all $d \in \mathbb{N}$ must use at least $\Omega\left(\frac{d}{\log(d)}\right)$ samples from $\mathcal{D}$.*

The proof of this lemma relies on a lower bound for testing support size of a distribution.

Definition 4.3.20. *Let $0 < \alpha < \beta < 1$ and let $n \in \mathbb{N}$. The* <u>*(α, β, n)-support size testing problem*</u> *is the following promise problem. Let $\mathcal{D} \in \Delta([n])$ be an unknown distribution such that $\forall i \in \mathrm{supp}(\mathcal{D}) : \ \mathcal{D}(i) \geq \frac{1}{n}$. Given sample access to $\mathcal{D}$, distinguish between the following two cases:*

(i) $|\mathrm{supp}(D)| \leq \alpha \cdot n$.

(ii) $|\mathrm{supp}(D)| \geq \beta \cdot n$.

The following tight lower bound for this problem is due to Valiant and Valiant (2010,). The formulation we use is adapted from Canonne (2020).[19]

Theorem 4.3.21 (Valiant and Valiant, 2010,; Canonne, 2020, Theorem 3.5.3)**.** *Let $0 < \alpha < \beta < 1$. Any tester that uses sample access to an unknown distribution $\mathcal{D} \in \Delta([n])$ and solves the (α, β, n)-support size testing problem correctly with probability at least $\frac{2}{3}$ for all $n \in \mathbb{N}$ must use at least $\Omega\left(\frac{n}{\log(n)}\right)$ samples from $\mathcal{D}$.*

Proof of Lemma 4.3.19. We show the following reduction from the support size testing problem to the threshold closeness problem: Assume T' is a tester that solves the (α, β, d)-threshold closeness testing problem correctly with probability at least $\frac{2}{3}$ for all $d \in \mathbb{N}$ using

[19] See also the discussion following Theorem 3.1 in Ron and Tsur (2013), and Theorem 5.3 in Valiant (2012). Similar bounds that appear in Valiant (2011, Claim 3.10) and Raskhodnikova, Ron, Shpilka, and Smith (2009, Theorem 2.1 and Corollary 2.2) are slightly weaker, but would also suffice for separating between 2-PAC verification versus 2-PAC learning of $\mathcal{T}_d$, as in Claim 4.3.22.

$m(d)$ samples. Then there exists a tester T that solves the $(2\alpha, 2\beta, d)$-support size testing problem correctly with probability at least $\frac{2}{3}$ for all $d \in \mathbb{N}$, and uses at most $m(d)$ samples.

For any distribution $\mathcal{D} \in \Delta([d])$, define a corresponding distribution $\mathcal{D}' \in \Delta([0, d] \times \{0, 1\})$ as follows. For all $i \in [d]$, let $a_i = i - \frac{3}{4}$ and $b_i = i - \frac{1}{4}$. Then $\mathcal{D}'(a_i, 1) = \frac{\mathcal{D}(i)}{2}$ and $\mathcal{D}'(b_i, 0) = \frac{1}{2d}$ for all $i \in [d]$, and $\mathcal{D}'$ vanishes elsewhere.

Given sample access to $\mathcal{D}$, it is possible to simulate sample access to $\mathcal{D}'$: with probability $\frac{1}{2}$, sample $i \in \mathcal{D}$, and output $(a_i, 1)$; with probability $\frac{1}{2}$ select $i \in [d]$ uniformly at random, and output $(b_i, 0)$.

Because $\mathcal{T}_d$ consists of monotone increasing thresholds,

$$
\begin{aligned}
L_{\mathcal{D}'}(\mathcal{T}_d) &= \sum_{i=1}^{n} \min\{\mathcal{D}'(a_i, 1), \mathcal{D}'(b_i, 0)\} \\
&= \sum_{i=1}^{n} \min\left\{\frac{\mathcal{D}(i)}{2}, \frac{1}{2d}\right\} \\
&\overset{(*)}{=} \sum_{i \in [d] \setminus \operatorname{supp}(\mathcal{D})} 0 + \sum_{i \in \operatorname{supp}(\mathcal{D})} \frac{1}{2d} \\
&= \frac{|\operatorname{supp}(\mathcal{D})|}{2d}.
\end{aligned}
$$

Equality $(*)$ holds whenever $\mathcal{D}$ is an input for the support size testing problem, because we assume that $\mathcal{D}(i) \geq \frac{1}{d}$ for all $i \in \operatorname{supp}(\mathcal{D})$.

To solve the $(2\alpha, 2\beta, d)$-support size testing problem, T operates as follows. Given access to an unknown distribution $\mathcal{D} \in \Delta([d])$, it simulates an execution of T' with access to $\mathcal{D}'$ that solves the (α, β, d)-threshold closeness testing problem. If T' decides that $L_{\mathcal{D}'}(\mathcal{T}_d) \leq \alpha$, then T outputs that $|\operatorname{supp}(\mathcal{D})| \leq 2\alpha \cdot d$, and if T' decides that $L_{\mathcal{D}'}(\mathcal{T}_d) \geq \beta$ then T outputs that $|\operatorname{supp}(\mathcal{D})| \geq 2\beta \cdot d$. T decides correctly with probability at least $\frac{2}{3}$, because we assume that T' decides correctly with probability at least $\frac{2}{3}$, and

$$
L_{\mathcal{D}'}(\mathcal{T}_d) \leq \alpha \iff |\operatorname{supp}(\mathcal{D})| \leq 2\alpha \cdot d
$$

$$
L_{\mathcal{D}'}(\mathcal{T}_d) \geq \beta \iff |\operatorname{supp}(\mathcal{D})| \geq 2\beta \cdot d.
$$

T requires at most as many samples as T' does, because simulating one sample from $\mathcal{D}'$ requires taking at most one sample from $\mathcal{D}$.

The claim follows from this reduction and from Theorem 4.3.21. □

The previous claim also implies the following lower bound for 2-PAC learning of $\mathcal{T}_d$ without the help of a prover.

Claim 4.3.22. *2-PAC learning the class $\mathcal{T}_d$ with $\varepsilon \in (0, \frac{1}{32})$ requires at least $\Omega\left(\frac{d}{\log(d)}\right)$ random samples. This is true even if we assume that the unknown underlying distribution $\mathcal{D}$ satisfies $L_{\mathcal{D}}(\mathcal{T}_d) > 0$.*

Proof of Claim 4.3.22. Assume for contradiction that there exists an algorithm A that 2-PAC learns $\mathcal{T}_d$ using only $o\left(\frac{d}{\log(d)}\right)$ samples from $\mathcal{D}$. We construct a tester T that solves the $(\frac{1}{8}, \frac{3}{8}, d)$-threshold closeness testing problem using only $o\left(\frac{d}{\log(d)}\right)$ samples.

Let $\mathcal{D} \in \Delta([0, d] \times \{0, 1\})$ be the unknown distribution that T has access to. Fix positive $\varepsilon \leq \frac{1}{32}$, $\delta \leq \frac{1}{6}$. T operates as follows. It simulates A using samples from $\mathcal{D}$ to obtain $h \in \mathcal{T}_d$ such that with probability at least $1 - \delta$,

$$L_{\mathcal{D}}(h) \leq 2 \cdot L_{\mathcal{D}}(\mathcal{T}_d) + \varepsilon. \tag{4.12}$$

Next, it takes an additional $O(1)$ samples from $\mathcal{D}$ to obtain an estimate $\widehat{\ell}$ such that with probability at least $1 - \delta$,

$$\left|\widehat{\ell} - L_{\mathcal{D}}(h)\right| \leq \varepsilon \tag{4.13}$$

If $\widehat{\ell} \leq \frac{5}{16}$, then T outputs $L_{\mathcal{D}}(\mathcal{T}_d) \leq \frac{1}{8}$. Otherwise, if $\widehat{\ell} > \frac{5}{16}$, then T outputs $L_{\mathcal{D}}(\mathcal{T}_d) \geq \frac{3}{8}$.

From the union bound, with probability at least $1 - 2\delta \geq \frac{2}{3}$, both (4.12) and (4.13) hold. Correctness follows by considering each case separately:

- **Case 1:** $L_{\mathcal{D}}(\mathcal{T}_d) \leq \frac{1}{8}$**.** Then

$$\widehat{\ell} \leq L_{\mathcal{D}}(h) + \varepsilon \leq 2L_{\mathcal{D}}(\mathcal{T}_d) + 2\varepsilon \leq \frac{2}{8} + \frac{2}{32} = \frac{5}{16}.$$

- **Case 2:** $L_{\mathcal{D}}(\mathcal{T}_d) \geq \frac{3}{8}$**.** Then

$$\widehat{\ell} \geq L_{\mathcal{D}}(h) - \varepsilon \geq L_{\mathcal{D}}(\mathcal{T}_d) - \varepsilon \geq \frac{3}{8} - \frac{1}{32} = \frac{11}{32} > \frac{5}{16}.$$

Finally, T uses the same number of samples as A does, which is a contradiction to Lemma 4.3.19.

From an amplification argument, the claim holds for any $\delta \in (0, \frac{1}{2})$. To see that the claim is true even if we assume that $L_{\mathcal{D}}(\mathcal{T}_d) > 0$, note that the distribution $\mathcal{D}'$ constructed in the proof of Lemma 4.3.19 always satisfies $L_{\mathcal{D}'}(\mathcal{T}_d) \geq \frac{1}{2d}$, and so we may assume that the hard distributions for T in the current proof have this property. □

Finally, we have obtained the desired separation, showing that PAC verification can be more efficient than PAC learning and closeness testing.

Proof of Theorem 4.3.8.

(i) Follows from Claim 4.3.17.

(ii) Follows from Theorem 4.1.15, because $\mathsf{VC}(\mathcal{T}_d) \geq d$.

(iii) Follows from Claim 4.3.22.

(iv) Follows from Lemma 4.3.19. □

4.4 Lower Bound of $\tilde{\Omega}(d)$

We saw in Section 4.3 that for every natural number d there exists a class of VC dimension d that has a verification protocol requiring only $O\left(\sqrt{d}\right)$ samples for the verifier – a considerable saving compared to the cost of learning, which is $\Omega(d)$. A natural question to ask is, "Does every class of VC dimension d admit a verification protocol with sample complexity $O\left(\sqrt{d}\right)$?" In other words, is it always worthwhile to delegate a learning task? In this section we provide a partial negative answer to this question, presenting for every natural number d an example of a class with VC dimension $O(d\log(d))$ where the sample complexity for proper PAC verification is $\Omega(d)$. That is, for these classes the sample complexity of learning and of proper verification are equal up to a logarithmic factor. Formally:

Theorem 4.4.1. *For every $\varepsilon, \delta \in \left(0, \frac{1}{8}\right)$ there exist constants $c_0, c_1, c_2 > 0$ and a sequence of classes $\mathcal{H}_1, \mathcal{H}_2, \ldots$ such that:*

- *For all $d \in \mathbb{N}$, the class $\mathcal{H}_d$ has VC dimension at most $c_0 \cdot d\log(d)$.*
- *The sample complexity of proper 1-PAC verifying $\mathcal{H}_d$ is $\Omega(d)$. That is, if*

  $$(V_1, P_1), (V_2, P_2), \ldots$$

  is a sequence such that for all $d \in \mathbb{N}$, (V_d, P_d) is an interactive proof system that 1-PAC verifies $\mathcal{H}_d$ using oracles that provide random samples such that the output is either 'reject' or in $\mathcal{H}_d$, then for all $d \geq c_1$, V_d uses at least $c_2 \cdot d$ random samples when executed on input (ε, δ).

Remark 4.4.2 (Non-proper verification). *Theorem 4.4.1 pertains to* proper *PAC verification. Note that if the distribution is labeled by a function,*[20] *then the sample complexity for non-proper PAC verification is $O(\frac{\log(\frac{1}{\delta})}{\varepsilon^2})$: the prover sends a description of $f_{\mathcal{D}}$ to the verifier, and the verifier can easily check that the proposed function has near-zero loss by using this number of random samples. The only issues in this case pertain to the prover's ability to find the function, and send a succinct description of it, as well as the verifier's ability to evaluate the function efficiently at points of its choosing.*

In contrast, the sample complexity of non-proper PAC verification in the general case is less clear, and merits further consideration.

The Class $\mathcal{H}_d$

Notation 4.4.3. *For any $d \in \mathbb{N}$, we write $\mathcal{X}_d$ to denote some fixed set of cardinality $n_d = 2d^2$.*

[20] Formally, if the family $\mathbb{D}$ of distributions satisfies that for every $\mathcal{D} \in \mathbb{D}$ there exists a function $f_{\mathcal{D}}$ such that $\mathbb{P}_{(x,y)\sim\mathcal{D}}[y = f_{\mathcal{D}}(x)] = 1$.

Notation 4.4.4. *For any $d \in \mathbb{N}$, we write $\mathcal{F}_{d,\frac{1}{2}}$ to denote the set of balanced boolean functions over $\mathcal{X}_d$, namely,*

$$\mathcal{F}_{d,\frac{1}{2}} = \left\{ f \in \{0,1\}^{\mathcal{X}_d} : \; |f^{-1}(1)| = \frac{n_d}{2} = |f^{-1}(0)| \right\}.$$

Notation 4.4.5. *For any $f \in \mathcal{F}_{d,\frac{1}{2}}$, we write $\mathcal{D}_f$ to denote the distribution over tuples in $\mathcal{X}^t$ in which t elements are samples independently and uniformly at random from* $\mathrm{supp}(f)$*. Namely, for any $(x_1, \dots, x_t) \in \mathcal{X}^t$,*

$$\mathcal{D}_f\left((x_1, \dots, x_t)\right) = \begin{cases} \left(\frac{2}{n}\right)^t & x_1, \dots, x_t \in \mathrm{supp}(f) \\ 0 & \text{o.w.} \end{cases}$$

Furthermore, for any $F = \{f_1, \dots, f_k\} \subseteq \mathcal{F}_{d,\frac{1}{2}}$, we write $\mathcal{D}_F$ to denote the distribution over $\mathcal{X}^t$ given by

$$\mathcal{D}_F(x_1, \dots, x_t) := \frac{1}{k} \sum_{i=1}^{k} \mathcal{D}_{f_i}(x_1, \dots, x_t).$$

Lastly, $\mathcal{U}_{\mathcal{X}^t}$ denotes the uniform distribution over $\mathcal{X}^t$.

We now define the sequence of classes $\mathcal{H}_d$ for $d \in \mathbb{N}$.

Definition 4.4.6. *Fix $\delta \in (0,1)$. For any $d \in \mathbb{N}$, let $\mathcal{X}_d = [n_d]$ for $n_d = 2d^2$, and let $t_d = \left\lfloor c_2 \cdot d \right\rfloor$ where*

$$c_2 = \sqrt{\frac{\log(1 - \delta/3)}{\log(1/2e)}}.$$

The class $\mathcal{H}_d$ is a subset of $\mathcal{F}_{d,\frac{1}{2}}$ of cardinality

$$k_d = \left(\frac{3 n_d^{\sqrt{n_d}}}{\delta} \right)^3$$

which is defined as follows. For all values d in which this is possible, the subset $\mathcal{H}_d$ is chosen such that the following three properties hold:

H1. $\mathsf{TV}(\mathcal{D}_{\mathcal{H}_d}, \mathcal{U}_{\mathcal{X}^t}) \leq \delta$.

H2. *Every distinct $g_1, g_2 \in \mathcal{H}_d$ satisfy $|\mathrm{supp}(g_1) \cap \mathrm{supp}(g_2)| \leq \frac{3n_d}{8}$.*

H3. *All subsets $X \subseteq \mathcal{X}_d$ of size at most $\sqrt{n}$ satisfy*

$$|\{f \in \mathcal{H}_d : \; X \subseteq \mathrm{supp}(f)\}| \geq \frac{1}{\delta}.$$

However, if for some value of d there exists no subset of cardinality k_d that satisfies these properties, then for that d the class $\mathcal{H}_d$ is simply fixed to be some arbitrary subset of cardinality k_d.

Remark 4.4.7. *It is not obvious that a set $\mathcal{H}_d$ as in the definition above exists. In Lemma 4.4.11 below, we prove the existence of $\mathcal{H}_d$ for all d large enough.*

Notation 4.4.8. *For the remainder of this section, we often neglect to write the subscript d wherever it is readily understood from the context.*

Note that the VC dimension of $\mathcal{H}_d$ is at most $\log\left(|\mathcal{H}_d|\right) = O\left(d\log\left(d\right)\right)$, matching the requirement in the theorem.

Proof Idea

For any d large enough, we want to show that at least $t_d = \Omega(d)$ samples are necessary.

Consider PAC learning the class $\mathcal{H}_d$ in the special case where all $x \in \mathcal{X}$ are labeled 1, but the distribution over $\mathcal{X}_d$ is not known to the prover. Because every hypothesis in the class assigns incorrect labels of 0 to precisely half of the domain, a hypothesis achieves minimal loss if it assigns the 0 labels to a subset of size $\frac{n}{2}$ that has minimal weight with respect to the distribution over $\mathcal{X}_d$. Hence, to be successful the prover must learn enough about the distribution to identify a lightweight subset of size $\frac{n}{2}$ – but doing that requires $\Omega\left(\sqrt{n}\right) = \Omega(d)$ samples.

To formalize this idea we construct a stochastic process as follows. Let $P_{\mathcal{U}}$ denote a prover that causes V to accept with probability at least $1 - \delta$ when V receives samples from the uniform distribution over $\mathcal{X}$ (such a prover exists from the completeness property that V satisfies as a PAC learning verifier).

First, a set X_P of t_P samples is taken independently and uniformly from $\mathcal{X}$, where t_P is the number of samples required by $P_{\mathcal{U}}$. Next, two functions f_1 and f_2 are chosen uniformly from $\mathcal{H}_d$, and sets X_1 and X_2 each with t_d i.i.d. samples are taken from $\mathcal{D}_{f_1}$ and $\mathcal{D}_{f_2}$ respectively. A third set $X_{\mathcal{U}}$ is taken from $\mathcal{U}_{\mathcal{X}^t}$. The dependencies between these variables will be designed in such a way that with high probability $X_1 = X_2 = X_{\mathcal{U}}$. All samples are labeled with 1.

Finally, randomness values ρ_V and ρ_P are sampled for the prover and verifier, which are then executed to produce three hypotheses:

$$\begin{aligned} h_1 &:= [V(X_1, \rho_V), P_{\mathcal{U}}(X_P, \rho_P)], \\ h_2 &:= [V(X_2, \rho_V), P_{\mathcal{U}}(X_P, \rho_P)], \\ h_{\mathcal{U}} &:= [V(X_{\mathcal{U}}, \rho_V), P_{\mathcal{U}}(X_P, \rho_P)]. \end{aligned}$$

Observe that for $i = 1, 2$, because $X_i \sim \mathcal{D}_{f_i}$ and V is a PAC learner, with probability at least $1 - \delta$ either h_i is 'reject' or $L_{\mathcal{D}_{f_i}}(h_i) < \varepsilon$.

Observe further that when $X_1 = X_2 = X_{\mathcal{U}}$, the view of V (which consists of its samples, its randomness, and the transcript) is the same in all three executions, entailing that

$h_1 = h_2 = h_{\mathcal{U}}$. Additionally, by the definition of $P_{\mathcal{U}}$, with probability at least $1 - \delta$ the output $h_{\mathcal{U}}$ is not 'reject', and so $h_1 = h_2$ are not 'reject'.

However, Property H2 ensures that f_1 and f_2 have a small intersection, causing any hypothesis that has a small loss with respect to $\mathcal{D}_{f_1}$ to have a large loss with respect to $\mathcal{D}_{f_2}$, and vice versa. This is a contradiction to the above observation that $L_{\mathcal{D}_{f_i}}(h_i) < \varepsilon$ for both $i = 1$ and $i = 2$.

Remark 4.4.9. *Because we are dealing exclusively with the case of learning the constant function that assigns the label* 1 *to all $x \in \mathcal{X}$, for the remainder of this section we will neglect to mention or denote the labels, which are always* 1*.*

Proof

We now translate the above proof idea into a formal proof of Theorem 4.4.1. The main step is to construct the following joint probability space.

Lemma 4.4.10. *For every $d \in \mathbb{N}$ large enough there exists a probability space with random variables*

$$(f_1, f_2, h_1, h_2, h_{\mathcal{U}}, X_1, X_2, X_{\mathcal{U}}, X_P, \rho_P, \rho_V)$$

such that $f_1, f_2, h_1, h_2, h_{\mathcal{U}} \in \mathcal{H}_d$ and $X_1, X_2, X_{\mathcal{U}} \in \mathcal{X}^t$ and the following properties hold:

P1. *X_P is a tuple of t_P samples taken independently and uniformly from $\mathcal{X}$, and is independent of all other variables.*

P2. *The marginal distribution of $X_{\mathcal{U}}$ is uniform over $\mathcal{X}^t$.*

P3. *For $i = 1, 2$, X_i is distributed according to $\mathcal{D}_{f_i}$. Namely, for any $g \in \mathcal{H}_d$ and any $x_1, \ldots, x_t \in \mathcal{X}$,*

$$\mathbb{P}[X_i = (x_1, \ldots, x_t) \mid f_i = g] = \mathcal{D}_g((x_1, \ldots, x_t)).$$

P4. *$X_1 = X_2$ with probability* 1*.*

P5. *$X_1 = X_{\mathcal{U}}$ with probability at least $1 - \delta$.*

P6. *ρ_V and ρ_P are randomness values for V and P with suitable marginal distributions and are independent of each other and of all other random variables.*

P7. *$h_\alpha = [V(X_\alpha, \rho_V), P_{\mathcal{U}}(X_P, \rho_P)]$ for $\alpha \in \{1, 2, \mathcal{U}\}$ with probability* 1*.*

P8. *$|\operatorname{supp}(f_1) \cap \operatorname{supp}(f_2)| \leq \frac{3n}{8}$ with probability at least $1 - \delta$.*

Before constructing the probability space, we show that the existence of such a space establishes the theorem:

Proof of Theorem 4.4.1. The requirement on the VC dimension holds because a class of cardinality k_d can have VC dimension at most $\log(k_d)$, and

$$\log(k_d) = \log\left(\left(\frac{3n_d^{\sqrt{n_d}}}{\delta}\right)^3\right) \leq 6d\log\left(\frac{6d^2}{\delta}\right) = O(d\log(d)).$$

For the lower bound on the sample complexity, fix d large enough such that $\mathcal{H}_d$ enjoys Properties H1, H2 and H3, and assume for contradiction that there exists a verifier that 1-PAC verifies $\mathcal{H} = \mathcal{H}_d$ with accuracy ε and confidence $1-\delta$ using at most $t = t_d$ samples. Because $X_i \sim \mathcal{D}_{f_i}$ (Property P3), the assumption that V is a PAC learner entails that

$$\forall i \in \{1,2\}: \ \mathbb{P}\Big[h_i = \text{reject} \ \vee \ \Big(h_i \neq \text{reject} \ \wedge \ L_{\mathcal{D}_{f_i}}(h_i) < \varepsilon\Big)\Big] \geq 1-\delta. \tag{4.14}$$

Because $X_{\mathcal{U}}$ is uniform over $\mathcal{X}^t$ and $h_{\mathcal{U}} := [V(X_{\mathcal{U}}, \rho_V), P_{\mathcal{U}}(X_P, \rho_P)]$ (by P2 and P7), the definition of $P_{\mathcal{U}}$ entails that

$$\mathbb{P}[h_{\mathcal{U}} \neq \text{reject}] \geq 1-\delta. \tag{4.15}$$

Next, because $\mathbb{P}[X_1 = X_{\mathcal{U}}] \geq 1-\delta$, $\mathbb{P}[X_1 = X_2] = 1$ and $h_i := [V(X_i, \rho_V), P_{\mathcal{U}}(X_P, \rho_P)]$ for $i \in \{1, 2, \mathcal{U}\}$ (by P5, P4 and P7), it follows that with probability at least $1-\delta$ the view of V when computing h_1 and h_2 is identical to its view when computing $h_{\mathcal{U}}$, and so

$$\mathbb{P}[h_1 = h_2 = h_{\mathcal{U}}] \geq 1-\delta. \tag{4.16}$$

Combining Equations (4.15) and (4.16) yields

$$\mathbb{P}[h_1 = h_2 \neq \text{reject}] \geq 1-2\delta.$$

Together with Equation (4.14), this entails that

$$\mathbb{P}\Big[(h_1 = h_2 \neq \text{reject}) \ \wedge \ \Big(L_{\mathcal{D}_{f_1}}(h_1) < \varepsilon\Big) \ \wedge \ \Big(L_{\mathcal{D}_{f_2}}(h_2) < \varepsilon\Big)\Big] \geq 1-4\delta. \tag{4.17}$$

However, low loss of h_i with respect to $\mathcal{D}_{f_i}$ entails that the supports of h_i and f_i have a large intersection. Indeed, for all $i \in \{1,2\}$,

$$\begin{aligned}\varepsilon \geq L_{\mathcal{D}_{f_i}}(h_i) :&= \mathbb{P}_{x\sim\mathcal{D}_{f_i}}[h_i(x) \neq f_i(x)] = \sum_{x\in\mathcal{X}} \mathcal{D}_{f_i}(x) \cdot \mathbb{1}_{h_i \neq f_i}(x) \\ &= \sum_{x\in\text{supp}(f_i)} \frac{2}{n} \cdot \mathbb{1}_{h_i\neq f_i}(x) = |\operatorname{supp}(f_i) \setminus \operatorname{supp}(h_i)| \cdot \frac{2}{n}.\end{aligned}$$

Thus,

$$|\operatorname{supp}(f_i) \setminus \operatorname{supp}(h_i)| \leq \frac{\varepsilon n}{2},$$

and so,

$$|\operatorname{supp}(f_i) \cap \operatorname{supp}(h_i)| = \frac{n}{2} - |\operatorname{supp}(f_i) \setminus \operatorname{supp}(h_i)| \geq \frac{n}{2} - \frac{\varepsilon n}{2},$$

Furthermore, because $h_1 = h_2$ the identity $|A \cap B| = |A| + |B| - |A \cup B|$ shows that the supports of f_1 and f_2 also have a large intersection:

$$\begin{aligned}
|\operatorname{supp}(f_1) \cap \operatorname{supp}(f_2)| &\geq |\operatorname{supp}(f_1) \cap \operatorname{supp}(f_2) \cap \operatorname{supp}(h_1)| \\
&= \Big|\operatorname{supp}(f_1) \cap \operatorname{supp}(h_1)\Big| + \Big|\operatorname{supp}(f_2) \cap \operatorname{supp}(h_2)\Big| \\
&\qquad - \Big|(\operatorname{supp}(f_1) \cap \operatorname{supp}(h_1)) \bigcup (\operatorname{supp}(f_2) \cap \operatorname{supp}(h_2))\Big| \\
&\geq \Big|\operatorname{supp}(f_1) \cap \operatorname{supp}(h_1)\Big| + \Big|\operatorname{supp}(f_2) \cap \operatorname{supp}(h_2)\Big| - \Big|\operatorname{supp}(h_1)\Big| \\
&\geq 2\left(\frac{n}{2} - \frac{\varepsilon n}{2}\right) - \frac{n}{2} \\
&\geq \frac{n}{2} - \varepsilon n.
\end{aligned}$$

That is, Equation (4.17) entails that

$$\mathbb{P}\Big[|\operatorname{supp}(f_1) \cap \operatorname{supp}(f_2)| \geq \frac{n}{2} - \varepsilon n\Big] \geq 1 - 4\delta.$$

In contrast, Property P8 states that

$$\mathbb{P}\Big[|\operatorname{supp}(f_1) \cap \operatorname{supp}(f_2)| \leq \frac{3n}{8}\Big] \geq 1 - \delta.$$

This is a contradiction whenever $\varepsilon < \frac{1}{8}$ and $\delta < \frac{1}{5}$. □

Construction of $\mathcal{H}_d$

To complete the proof, we construct the probability space of Lemma 4.4.10. The first step is to show that for large enough values of d, a suitable class $\mathcal{H}_d$ can be constructed simply by choosing a set of k functions uniformly at random from $\mathcal{F}_{1/2}$.

Lemma 4.4.11. *Fix $\delta \in (0,1)$. The following holds for any value $d \in \mathbb{N}$ that is large enough. Let F denote a set of k_d functions chosen uniformly and independently from $\mathcal{F}_{d,\frac{1}{2}}$. Then with probability at least $1 - 3\delta$, F satisfies Properties H1, H2 and H3.*

The lemma follows immediately from Claims 4.4.12, 4.4.21 and 4.4.23 below, so the remainder of this section is devoted to stating and proving those claims.

Property H1: $\mathsf{TV}(\mathcal{U}_{\mathcal{X}^t}, \mathcal{D}_{\mathcal{H}_d}) \leq \delta$

In this subsection we prove that the distribution $\mathcal{D}_F$ defined by F is close to the uniform distribution on $\mathcal{X}$ in the following sense.

Claim 4.4.12. *Fix $\delta \in (0,1)$. The following holds for all values of n that are large enough. Let $F = \{f_1, \dots, f_k\}$ denote a set of functions chosen uniformly and independently from $\mathcal{F}_{1/2}$. If*

$$k \geq \left(\frac{3n^{\sqrt{n}}}{\delta}\right)^3$$

and $t_d = \left\lfloor c_2 \cdot d \right\rfloor$ for

$$c_2 = \sqrt{\frac{\log(1-\delta/3)}{\log(1/2e)}}$$

then

$$\mathbb{P}_F[\mathsf{TV}(\mathcal{U}_{\mathcal{X}^t}, \mathcal{D}_F) \leq \delta] \geq 1-\delta.$$

The proof is partitioned to the following claims.

Claim 4.4.13. *For any integer $0 \leq s \leq n$, any set $X \subseteq \mathcal{X}$ of size s and any $z \in [n]$,*

$$\mathbb{P}_{f \in \{0,1\}^{\mathcal{X}}}\left[X \subseteq \operatorname{supp}(f) \;\middle|\; |\operatorname{supp}(f)| = z\right] = \frac{\binom{z}{s}}{\binom{n}{s}}.$$

Proof. If $z < s$ then the probability is clearly 0. Otherwise,

$$\begin{aligned}
\mathbb{P}_{f \in \{0,1\}^{\mathcal{X}}}\left[X \subseteq \operatorname{supp}(f) \;\middle|\; |\operatorname{supp}(f)| = z\right] &= \frac{\left|\{g \in \{0,1\}^{\mathcal{X}} :\ |\operatorname{supp}(g)| = z \ \wedge\ X \subseteq \operatorname{supp}(g)\}\right|}{|\{g \in \{0,1\}^{\mathcal{X}} :\ |\operatorname{supp}(g)| = z\}|} \\
&= \frac{\binom{n-s}{z-s}}{\binom{n}{z}} \\
&= \frac{\binom{n-s}{z-s}}{\binom{n}{z}} \cdot \frac{\binom{n}{s}}{\binom{n}{s}} \\
&\overset{(*)}{=} \frac{\binom{n}{z}\binom{z}{s}}{\binom{n}{z}\binom{n}{s}} \\
&= \frac{\binom{z}{s}}{\binom{n}{s}},
\end{aligned}$$

where $(*)$ follows from the identity $\binom{n}{s}\binom{n-s}{z-s} = \binom{n}{z}\binom{z}{s}$, which holds because both expressions count the number of ways to choose a committee of size z with a subcommittee of size s from a set of n candidates. □

Corollary 4.4.14. *For any set $X \subseteq \mathcal{X}$ of size s,*

$$\mathbb{P}_{f \in \mathcal{F}_{1/2}}[X \subseteq \operatorname{supp}(f)] = \frac{\binom{\frac{n}{2}}{s}}{\binom{n}{s}}.$$

Notation 4.4.15. *For any $f \in \mathcal{F}_{1/2}$, we write $\mathcal{D}_f^{\text{distinct}}$ to denote the uniform distribution over tuples of length t that contain t distinct elements from $\operatorname{supp}(f)$. That is, for any $(x_1, \dots, x_t) \in \mathcal{X}^t$,*

$$\mathcal{D}_f^{\text{distinct}}\left((x_1, \dots, x_t)\right) = \begin{cases} \frac{1}{\binom{\frac{n}{2}}{t} \cdot t!} & x_1, \dots, x_t \in \operatorname{supp}(f) \ \wedge \ |\{x_1, \dots, x_t\}| = t \\ 0 & \text{o.w.} \end{cases}$$

Furthermore, let $\mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}$ denote the uniform distribution over the set of tuples of length t from $\mathcal{X}$ with distinct elements,

$$\left\{(x_1, \dots, x_t) \in \mathcal{X}^t : \ |\{x_1, \dots, x_t\}| = t\right\}.$$

That is,

$$\mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}\left((x_1, \dots, x_t)\right) = \begin{cases} \frac{1}{\binom{n}{t} \cdot t!} & x_1, \dots, x_t \in \mathcal{X} \ \wedge \ |\{x_1, \dots, x_t\}| = t \\ 0 & \text{o.w.} \end{cases}$$

Claim 4.4.16. *For any ordered tuple $X \in \mathcal{X}^t$ with distinct elements,*

$$\mathbb{E}_{f \in \mathcal{F}_{1/2}}\left[\mathcal{D}_f^{\text{distinct}}(X)\right] = \frac{1}{\binom{n}{t} t!}.$$

Proof. Using Corollary 4.4.14,

$$\begin{aligned} \mathbb{E}_{f \in \mathcal{F}_{1/2}}\left[\mathcal{D}_f^{\text{distinct}}(X)\right] &= \mathbb{P}[X \subseteq \operatorname{supp}(f)] \cdot \frac{1}{\binom{\frac{n}{2}}{t} t!} + \mathbb{P}_{f \in \mathcal{F}_{1/2}}[X \nsubseteq \operatorname{supp}(f)] \cdot 0 \\ &= \frac{\binom{\frac{n}{2}}{t}}{\binom{n}{t}} \cdot \frac{1}{\binom{\frac{n}{2}}{t} t!} \\ &= \frac{1}{\binom{n}{t} t!}. \end{aligned}$$

□

Claim 4.4.17. *Consider k functions $f_1, \dots, f_k$ chosen independently and uniformly at random from $\mathcal{F}_{1/2}$. For any $\delta \in (0,1)$ and ordered tuple $X \in \mathcal{X}^t$ with distinct elements, if*

$$k \geq \left(\frac{n^{\sqrt{n}}}{\delta}\right)^3$$

then

$$\mathbb{P}_{f_1,\dots,f_k \in \mathcal{F}_{1/2}}\left[\left|\mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}(X) - \frac{1}{k}\sum_{i=1}^{k}\mathcal{D}_{f_i}^{\text{distinct}}(X)\right| > \frac{\delta}{\binom{n}{t}t!}\right] \leq \frac{\delta}{\binom{n}{t}t!}.$$

Proof. Fix X. Observe that when $f_1, \dots, f_k$ are chosen independently and uniformly then $\left\{\mathcal{D}_{f_i}^{\text{distinct}}(X)\right\}_{i \in [k]}$ is a set of i.i.d. random variables each of which takes values in $[0,1]$. Furthermore, from Claim 4.4.16 the expectation of each of these random variables is $\mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}(X)$. Thus, from Hoeffding's inequality, the left-hand side in the claim is at most

$$2\exp\left(-2k\left(\frac{\delta}{\binom{n}{t}t!}\right)^2\right),$$

and so taking

$$k \geq \frac{1}{2}\left(\frac{\binom{n}{t}t!}{\delta}\right)^2 \log\left(\frac{2\binom{n}{t}t!}{\delta}\right)$$

is sufficient to obtain the desired bound. A direct calculation shows that

$$\begin{aligned}
\frac{1}{2}\left(\frac{\binom{n}{t}t!}{\delta}\right)^2 \log\left(\frac{2\binom{n}{t}t!}{\delta}\right) &\leq \left(\frac{\binom{n}{t}t!}{\delta}\right)^3 \\
&\leq \left(\frac{\binom{n}{\sqrt{n}}\sqrt{n}!}{\delta}\right)^3 \\
&= \left(\frac{n(n-1)\cdots(n-\sqrt{n}+1)}{\delta}\right)^3 \\
&\leq \left(\frac{n^{\sqrt{n}}}{\delta}\right)^3,
\end{aligned}$$

as desired. □

Notation 4.4.18. *For any $F = \{f_1, \dots, f_k\} \subseteq \mathcal{F}_{1/2}$, we write $\mathcal{D}_F^{\text{distinct}}$ to denote the distribution over $\mathcal{X}^t$ given by*

$$\mathcal{D}_F^{\text{distinct}}(x_1, \dots, x_t) := \frac{1}{k} \sum_{i=1}^{k} \mathcal{D}_{f_i}^{\text{distinct}}(x_1, \dots, x_t).$$

Claim 4.4.19. *Let $F = \{f_1, \dots, f_k\}$ denote a set of functions chosen uniformly and independently from $\mathcal{F}_{1/2}$. For any $\delta \in (0,1)$, if*

$$k \geq \left(\frac{3n^{\sqrt{n}}}{\delta}\right)^3$$

then

$$\mathbb{P}_F\left[\mathsf{TV}\Big(\mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}, \mathcal{D}_F^{\text{distinct}}\Big) \leq \frac{\delta}{3}\right] \geq 1 - \frac{\delta}{3}.$$

Proof. From Claim 4.4.17, taking k as in the statement ensures that for any particular tuple $X \in \mathcal{X}^t$ with distinct elements,

$$\mathbb{P}_{f_1,\dots,f_k \in \mathcal{F}_{1/2}}\left[\Big|\mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}(X) - \mathcal{D}_F^{\text{distinct}}(X)\Big| > \frac{\delta}{3\binom{n}{t}t!}\right] \leq \frac{\delta}{3\binom{n}{t}t!}.$$

From the union bound, we conclude that with probability at least $1 - \frac{\delta}{3}$, the inequality

$$\Big|\mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}(X) - \mathcal{D}_F^{\text{distinct}}(X)\Big| \leq \frac{\delta}{3\binom{n}{t}t!}$$

holds for all $\binom{n}{t}t!$ such tuples simultaneously. In this case,

$$\mathsf{TV}\Big(\mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}, \mathcal{D}_F^{\text{distinct}}\Big) = \frac{1}{2}\sum_{X \in \mathcal{X}^t}\Big|\mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}(X) - \mathcal{D}_F^{\text{distinct}}(X)\Big| \leq \frac{\delta}{6}.$$

□

Proof of Claim 4.4.12. From the triangle inequality

$$\mathsf{TV}(\mathcal{U}_{\mathcal{X}^t}, \mathcal{D}_F) \leq \mathsf{TV}\Big(\mathcal{U}_{\mathcal{X}^t}, \mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}\Big) + \mathsf{TV}\Big(\mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}, \mathcal{D}_F^{\text{distinct}}\Big) + \mathsf{TV}\Big(\mathcal{D}_F^{\text{distinct}}, \mathcal{D}_F\Big).$$

Therefore, it suffices to show the following three inequalities:

(*i*) $\mathsf{TV}\Big(\mathcal{U}_{\mathcal{X}^t}, \mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}\Big) \leq \frac{\delta}{3}$ for n large enough. Indeed,

$$\begin{aligned}
\mathsf{TV}\Big(\mathcal{U}_{\mathcal{X}^t}, \mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}\Big) &= \max_{A \subseteq \mathcal{X}^t} \Big(\mathcal{U}_{\mathcal{X}^t}(A) - \mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}(A)\Big) \\
&= \sum_{(x_1,\dots,x_t)\in\mathcal{X}^t:\ |\{x_1,\dots,x_t\}|<t} \left(\frac{1}{n^t} - 0\right) \\
&= \left(n^t - \binom{n}{t} t!\right) \frac{1}{n^t} \\
&= 1 - \frac{n(n-1)\cdots(n-t+1)}{n^t} \\
&\leq 1 - \left(1 - \frac{t}{n}\right)^t \\
&\leq 1 - \left(1 - \frac{(c_2\sqrt{n})}{n}\right)^{c_2\sqrt{n}} \\
&= 1 - \left(1 - \frac{c_2}{\sqrt{n}}\right)^{\frac{\sqrt{n}}{c_2}\cdot c_2^2} \\
&\overset{(*)}{\leq} 1 - \left(\frac{1}{2e}\right)^{c_2^2} \\
&\overset{(**)}{\leq} \frac{\delta}{3},
\end{aligned}$$

where $(*)$ holds for all n large enough because $\left(1 - \frac{c_2}{\sqrt{n}}\right)^{\frac{\sqrt{n}}{c_2}} \xrightarrow{n\to\infty} \frac{1}{e}$ from below, and $(**)$ holds whenever

$$c_2 \leq \sqrt{\frac{\log(1-\delta/3)}{\log(1/2e)}}.$$

(*ii*) $\mathbb{P}_F\Big[\mathsf{TV}\Big(\mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}, \mathcal{D}_F^{\text{distinct}}\Big) > \frac{\delta}{3}\Big] \leq \delta$. This is true by Claim 4.4.19.

(*iii*) $\mathsf{TV}\Big(\mathcal{D}_F^{\text{distinct}}, \mathcal{D}_F\Big) \leq \frac{\delta}{3}$ or n large enough. This follows from a calculation very similar to (*i*).

We conclude that for n large enough, with probability at least $1-\delta$ over the choice of F,

$$\mathsf{TV}(\mathcal{U}_{\mathcal{X}^t}, \mathcal{D}_F) \leq \delta,$$

as desired. □

Property H2: $\forall i \neq j: \quad |\operatorname{supp}(f_i) \cap \operatorname{supp}(f_j)| \leq \frac{3n}{8}$

In this section we show that random sets typically form a code.

Claim 4.4.20. $\mathbb{P}_{f_1,f_2\in\mathcal{F}_{1/2}}\Big[|\operatorname{supp}(f_1)\cap\operatorname{supp}(f_2)| > \frac{3n}{8}\Big] \le \frac{\delta}{k^2}$.

Proof. Let $\operatorname{supp}(f_2) = \{x_1,\dots,x_{n/2}\}$. We think of this experiment as if f_1 is chosen first, and then we count how many members of $\operatorname{supp}(f_2)$ fall inside $\operatorname{supp}(f_1)$. The expected number of hits is $\frac{n}{4}$, and they are independent, so we can use Hoeffding's bound to prove the claim.

$$\begin{aligned}\mathbb{P}_{f_1,f_2\in\mathcal{F}_{1/2}}\left[|\operatorname{supp}(f_1)\cap\operatorname{supp}(f_2)| > \frac{3n}{8}\right] &\le \mathbb{P}_{f_1,f_2\in\mathcal{F}_{1/2}}\left[\sum_{i=1}^{n/2}\mathbb{1}(x_i\in\operatorname{supp}(f_1)) > \frac{3n}{8}\right]\\ &= \mathbb{P}_{f_1,f_2\in\mathcal{F}_{1/2}}\left[\frac{2}{n}\sum_{i=1}^{n/2}\mathbb{1}(x_i\in\operatorname{supp}(f_1)) > \frac{3}{4}\right]\\ &\le \mathbb{P}_{f_1,f_2\in\mathcal{F}_{1/2}}\left[\left|\frac{2}{n}\sum_{i=1}^{n/2}\mathbb{1}(x_i\in\operatorname{supp}(f_1)) - \frac{1}{2}\right| > \frac{1}{4}\right]\\ &\le 2\exp\left(-2\cdot\frac{n}{2}\cdot\left(\frac{1}{4}\right)^2\right) = 2^{\Theta(-n)}.\end{aligned}$$

In contrast, considering δ to be a constant, it holds that

$$\frac{\delta}{k^2} = 2^{\Theta\left(-\log(n)\sqrt{n}\right)},$$

and so for n large enough we obtain $\mathbb{P}_{f_1,f_2\in\mathcal{F}_{1/2}}\Big[|\operatorname{supp}(f_1)\cap\operatorname{supp}(f_2)| > \frac{3n}{8}\Big] \le \frac{\delta}{k^2}$, as desired. □

Claim 4.4.21. $\mathbb{P}_{f_1,\dots,f_k\in\mathcal{F}_{1/2}}\Big[\forall i\neq j\in[k]:\ |\operatorname{supp}(f_i)\cap\operatorname{supp}(f_j)| \le \frac{3n}{8}\Big] \ge 1-\delta$.

Proof.

$$\begin{aligned}&\mathbb{P}_{f_1,\dots,f_k\in\mathcal{F}_{1/2}}\left[\forall i\neq j\in[k]:\ |\operatorname{supp}(f_i)\cap\operatorname{supp}(f_j)| \le \frac{3n}{8}\right]\\ &= 1-\mathbb{P}\left[\bigcup_{i\neq j}\left\{|\operatorname{supp}(f_i)\cap\operatorname{supp}(f_j)| > \frac{3n}{8}\right\}\right]\\ &\ge 1-\sum_{i\neq j}\mathbb{P}\left[|\operatorname{supp}(f_i)\cap\operatorname{supp}(f_j)| > \frac{3n}{8}\right]\\ &\ge 1-k^2\cdot\frac{\delta}{k^2} = 1-\delta,\end{aligned}$$

where the last inequality follows from Claim 4.4.20. □

Property H3: $|F_X| \geq \frac{1}{\delta}$

In this section we show that there are typically many sets that contain a given subset of size order $\sqrt{n}$.

Notation 4.4.22. *Let $F \subseteq \mathcal{F}_{1/2}$, and let $X \subseteq \mathcal{X}$. We write F_X to denote the set*

$$\{f \in F : \; X \subseteq \mathrm{supp}(f)\}.$$

Claim 4.4.23. *Fix $\delta \in (0,1)$. Let $F = \{f_1, \dots, f_k\}$ denote a set of functions chosen uniformly and independently from $\mathcal{F}_{1/2}$. There exists N_0 such that for all $n \geq N_0$, if*

$$k \geq \left(\frac{n^{\sqrt{n}}}{\delta}\right)^3$$

then with probability at least $1 - \delta$ over the choice of F, all subsets $X \subseteq \mathcal{X}$ of size at most $\sqrt{n}$ satisfy

$$|F_X| \geq \frac{1}{\delta}.$$

Proof of Claim 4.4.23. Let $X \subseteq \mathcal{X}$ such that $|X| = t$. From Corollary 4.4.14,

$$\begin{aligned}
\mathbb{P}_{f \in \mathcal{F}_{1/2}}[X \subseteq \mathrm{supp}(f)] &= \frac{\binom{\frac{n}{2}}{t}}{\binom{n}{t}} = \frac{\frac{n}{2}!}{(\frac{n}{2}-t)!t!} \cdot \frac{(n-t)!t!}{n!} \\
&= \frac{n-t}{n} \cdot \frac{n-t-1}{(n-1)} \cdots \frac{\frac{n}{2}-t+1}{\frac{n}{2}+1} \\
&= \frac{\frac{n}{2}}{n} \cdot \frac{\frac{n}{2}-1}{(n-1)} \cdots \frac{\frac{n}{2}-t+1}{n-t+1} \\
&\geq \left(\frac{\frac{n}{2}-t}{n}\right)^t \\
&\geq \left(\frac{\frac{n}{2}-\sqrt{n}}{n}\right)^{\sqrt{n}} \\
&= \left(\frac{1}{2} - \frac{1}{\sqrt{n}}\right)^{\sqrt{n}} \geq 4^{-\sqrt{n}},
\end{aligned}$$

where the last inequality holds for $n \geq 16$. Observe that

$$\mu := \mathbb{E}_{f_1,\dots,f_k \in \mathcal{F}_{1/2}}[|F_X|] \geq k \cdot 4^{-\sqrt{n}} \geq 2^{\log(n)\sqrt{n} - 2\sqrt{n}} \xrightarrow{n\to\infty} \infty,$$

and choose N_0 large enough such that for all $n \geq N_0$, $\mathbb{E}[|F_X|] \geq \frac{2}{\delta}$.

Now, for any $n \geq N_0$ and any set X of size t, Hoeffding's inequality entails

$$\begin{aligned}\mathbb{P}_{f_1,\ldots,f_k \in \mathcal{F}_{1/2}}\left[|F_X| \leq \frac{1}{\delta}\right] &\leq \mathbb{P}\left[\Big||F_X| - \mu\Big| \geq \frac{k \cdot 4^{-\sqrt{n}}}{2}\right] \\ &= \mathbb{P}\left[\left|\frac{1}{k}\sum_{i=1}^{k} \mathbb{1}(X \subseteq \operatorname{supp}(f_i)) - \frac{\mu}{k}\right| \geq \frac{4^{-\sqrt{n}}}{2}\right] \\ &\leq 2\exp\left(-2k\left(\frac{4^{-\sqrt{n}}}{2}\right)^2\right).\end{aligned}$$

Hence, taking

$$k \geq \frac{1}{2} \cdot 4^{2\sqrt{n}+1} \cdot \log\left(\frac{2n^{\sqrt{n}}}{\delta}\right)$$

is sufficient to ensure that

$$\forall X \in \binom{\mathcal{X}}{t}: \ \mathbb{P}_{f_1,\ldots,f_k \in \mathcal{F}_{1/2}}\left[|F_X| \leq \frac{1}{\delta}\right] \leq \frac{\delta}{n^{\sqrt{n}}}$$

Taking k as in the claim is therefore more than sufficient to this end. Seeing as there exist less than $n^{\sqrt{n}}$ such sets, the union bound yields that

$$\mathbb{P}_{f_1,\ldots,f_k \in \mathcal{F}_{1/2}}\left[\forall X \subseteq \mathcal{X} \text{ s.t. } |X| \leq t: \ |F_X| \geq \frac{1}{\delta}\right] \geq 1 - \delta.$$

Note that for the case $|X| < t$ in the previous line, we have used the facts that X is contained in some set of size precisely t, and that $|F_X|$ is monotone decreasing with the cardinality of X. □

Construction of the Joint Probability Space

Assume $\mathcal{H}_d$ is a class that satisfies Properties H1, H2, and H3. We show how to use these properties to construct a joint probability space that satisfies Properties P1–8, proving Lemma 4.4.10.

The construction is as follows:

1. X_P is sampled uniformly from $\mathcal{X}^{t_P}$.
2. A function f_1 is chosen uniformly from $\mathcal{H}_d$.
3. $X_1 = (x_1, \ldots, x_t)$ is sampled i.i.d. from D_{f_1}.
4. X_2 is set to be equal to X_1.
5. A function f_2 is chosen uniformly from $\{f \in \mathcal{H}_d : \ X_2 \subseteq \operatorname{supp}(f)\}$.

6. $X_{\mathcal{U}} = (x_1^{\mathcal{U}}, \dots, x_t^{\mathcal{U}})$ is sampled such that its marginal distribution is uniform over $(\mathcal{X}_d)^t$, and also $\mathbb{P}[X_{\mathcal{U}} = X_1] \geq 1 - \delta$. This is possible due to Property H1 of the class $\mathcal{H}_d$.

7. ρ_V and ρ_P are sampled from the distributions of randomness used by V and $P_{\mathcal{U}}$ respectively, independently of each other and of everything else.

8. For $\alpha \in \{1, 2, \mathcal{U}\}$, compute $h_\alpha := [V(X_\alpha, \rho_V), P_{\mathcal{U}}(X_P, \rho_P)]$.

Note that Properties P1, P2, P4, P5, P6 and P7 are satisfied immediately by the construction, as is Property P3 for the case of $i = 1$. Property P8 is immediate from the construction together with H2 and H3. Hence, to prove the correctness of the construction, it suffices to prove that Property P3 holds also for the case $i = 2$, as in the following claim.

Claim 4.4.24. *The constriction in Section 4.4 satisfies that $X_2 \sim D_{f_2}$. More formally, for any $g \in \mathcal{H}_d$ and $x_1, \dots, x_t \in \mathcal{X}$,*

$$\mathbb{P}[X_2 = (x_1, \dots, x_t) \mid f_2 = g] = \mathcal{D}_g((x_1, \dots, x_t)).$$

Proof. By construction, $X_1 \sim D_{f_1}$. Hence, it is sufficient to show that

$$(X_1, f_1) \overset{d}{=} (X_2, f_2),$$

where $\overset{d}{=}$ denotes equality in distribution. Indeed, conditioned on $X_1 = X_2 = x$, both f_1 and f_2 are chosen i.i.d. uniformly in

$$F_x := \{f \in \mathcal{H}_d : \; x \subseteq \operatorname{supp}(f)\}.$$

More formally, for any $g \in \mathcal{H}_d$ and $x \in \mathcal{X}^t$,

- If $x \subseteq \operatorname{supp}(g)$ then

$$\begin{aligned}
\mathbb{P}[f_1 = g \mid X_1 = x] &= \frac{\mathbb{P}[X_1 = x \mid f_1 = g]\mathbb{P}[f_1 = g]}{\mathbb{P}[X_1 = x]} \\
&= \frac{\mathbb{P}[X_1 = x \mid f_1 = g]\mathbb{P}[f_1 = g]}{\sum_{g' \in F_x} \mathbb{P}[X_1 = x \mid f_1 = g']\mathbb{P}[f_1 = g']} \\
&= \frac{\mathbb{P}[X_1 = x \mid f_1 = g]}{\sum_{g' \in F_x} \mathbb{P}[X_1 = x \mid f_1 = g']} \\
&= \frac{1}{|F_x|} = \mathbb{P}[f_2 = g \mid X_2 = x].
\end{aligned}$$

- Otherwise, if $x \nsubseteq \operatorname{supp}(g)$ then

$$\mathbb{P}[f_1 = g \mid X_1 = x] = 0 = \mathbb{P}[f_2 = g \mid X_2 = x].$$

That is, for any $g \in \mathcal{H}_d$ and $x \in \mathcal{X}^t$,

$$\begin{aligned}\mathbb{P}[f_1 = g \ \wedge \ X_1 = x] = \mathbb{P}[f_1 = g \mid X_1 = x]\mathbb{P}[X_1 = x] = \\ = \mathbb{P}[f_2 = g \mid X_2 = x]\mathbb{P}[X_2 = x] = \mathbb{P}[f_2 = g \ \wedge \ X_2 = x].\end{aligned}$$

□

This proves Lemma 4.4.10, thereby concluding our proof of Theorem 4.4.1.

4.5 Efficient Verification via Query Delegation

In this section we present some simple results for the case in which the following two assumptions hold:[21]

1. Taking unlabeled samples from $\mathcal{X}$ is cheap, while obtaining labeled samples is costly. This is the assumption in the semi-supervised learning literature. It also holds when learning with respect to the uniform distribution on $\mathcal{X}$ (or some other known distribution on $\mathcal{X}$).

2. The distribution is labeled according to some function $f : \ \mathcal{X} \to \{0, 1\}$, and the prover has query access to f.

The basic idea of the results in this section is *query delegation*: The verifier simulates a learning algorithm that uses random samples, but for the majority of the samples the verifier can avoid accessing the distribution directly. Instead, it delegates the task of collecting the data to the prover. Consider the following illustration. First, the verifier chooses some $x_1, \ldots, x_m \in \mathcal{X}$ and sends them to the prover. For instance, the verifier may choose the x_i's by taking (cheap) unlabeled samples from the distribution. Secondly, the prover replies by sending a value $\tilde{y}_1, \ldots, \tilde{y}_m \in \{0, 1\}$ to the verifier that purportedly are the correct labels of the x_i's. Thirdly, the verifier independently takes a small amount of labeled samples directly from the distribution in order to decide whether to accept or reject the labels proposed by the prover. Finally, if the verifier does not detect any dishonesty, it will use the proposed labels to simulate the learning algorithm and output the resulting hypothesis.

The benefit of using query delegation is that the verifier requires much fewer labeled samples than are necessary for learning, with only a mild increase in time complexity.

[21] See more formal definitions in Conditions 4.5.1.

Following are a number of variations on this idea:

	V labeled samples	V queries	Messages	Assumption
Claim 4.5.2	$O\left(\frac{\log(\frac{1}{\delta})}{\varepsilon}\right)$	-	2	-
Claim 4.5.3	$O\left(\frac{\log(\frac{1}{\delta})}{\varepsilon}\right)$	-	2 (shorter)	PRG
Claim 4.5.4	-	$O\left(\frac{\log(\frac{1}{\delta})}{\varepsilon}\right)$	1	CRS

Table 4.1: Query delegation results.

Conditions 4.5.1 (Conditions for Query Delegation). Let $\mathcal{H}$ be a class of functions from $\mathcal{X}$ to $\{0,1\}$. Let $\mathbb{D}$ be a family of distributions over $\mathcal{X} \times \{0,1\}$ such that:

1. There exists a distribution $\mathcal{D}_{\mathcal{X}}$ over $\mathcal{X}$ such that for every $\mathcal{D} \in \mathbb{D}$, the marginal distribution of $\mathcal{D}$ on $\mathcal{X}$ is $\mathcal{D}_{\mathcal{X}}$.
2. For every $\mathcal{D} \in \mathbb{D}$, there exists a function $f_{\mathcal{D}} : \mathcal{X} \to \{0,1\}$ such that $\mathbb{P}_{(x,y)\sim\mathcal{D}}[f_{\mathcal{D}}(x) = y] = 1$.

Assume further that $\mathcal{H}$ has finite VC dimension. From Theorem 4.1.15, there exists an ERM algorithm that 1-PAC learns $\mathcal{H}$ using $m = m_{\mathcal{H}}(\varepsilon, \delta)$ random labeled samples. Let A be such an algorithm and assume that for any $\mathcal{D} \in \mathbb{D}$, A runs in time at most $t = t(\varepsilon, \delta)$.

Claim 4.5.2 (Simple Query Delegation). *Under Conditions 4.5.1, $\mathcal{H}$ is 1-PAC verifiable using verifier V and prover P such that:*

- *V has random sample access to the unknown distribution $\mathcal{D}$ and to the marginal distribution $\mathcal{D}_{\mathcal{X}}$. V uses only $k = O\left(\frac{\log(\frac{1}{\delta})}{\varepsilon}\right)$ labeled samples from $\mathcal{D}$, and uses $O(m)$ unlabeled samples from $\mathcal{D}_{\mathcal{X}}$.*
- *P has query access to $f_{\mathcal{D}}$, and uses $O(m)$ queries to this function.*
- *V runs in time $O(t(\frac{\varepsilon}{4}, \frac{\delta}{2}))$, and P runs in time $O(m)$.*
- *The protocol consists of two messages. First, V sends a message of length $O(m \log |\mathcal{X}|)$ to P, and then P sends back a message of length $O(m)$.*

Observe that 1-PAC learning requires $\Theta\left(\frac{d+\log(\frac{1}{\delta})}{\varepsilon^2}\right)$ where the VC dimension d can be any natural number. Hence, the above result implies that under Conditions 4.5.1, there exists a sample complexity separation of unbounded magnitude between PAC learning and PAC verifying for any family $\{\mathcal{H}_d\}_{d\in\mathbb{N}}$ where $\mathsf{VC}(\mathcal{H}_d) = d$ for all d.

If we assume the distribution $\mathcal{D}_{\mathcal{X}}$ has a pseudorandom generator (PRG) with respect to the ERM algorithm A, then we can also use slightly less communication.

Claim 4.5.3 (Compressed Query Delegation). *Under Conditions 4.5.1, assume that there exists a pseudorandom generator that generates samples from a distribution $\tilde{\mathcal{D}}_{\mathcal{X}}$ over $\mathcal{X}$, such that the algorithm A successfully 1-PAC learns $\mathcal{H}$ with respect to $\mathbb{D}$ as above when receiving labeled examples in which the marginal distribution over $\mathcal{X}$ is $\tilde{\mathcal{D}}_{\mathcal{X}}$ (instead of $\mathcal{D}_{\mathcal{X}}$). Then $\mathcal{H}$ is 1-PAC verifiable using a verifier V and prover P that satisfy the same conditions as in Claim 4.5.2, except that V sends a shorter message of length $O\left(\frac{\log(\frac{1}{\delta})}{\varepsilon}\log|\mathcal{X}|\right)$ to P. The security of the protocol is information-theoretic, and does not depend on any cryptographic assumptions. That is, soundness holds also with respect to an unbounded adversary that has full information about the pseudorandom generator mechanism and can distinguish whether a sample was taken from $\tilde{\mathcal{D}}_{\mathcal{X}}$ or from $\mathcal{D}_{\mathcal{X}}$.*

Finally, if we work in the common random string model (CRS) and we assume that the verifier also has query access to $f_{\mathcal{D}}$, then there exists a non-interactive protocol consisting of a single message sent from the prover to the verifier. This could be useful in cases where the prover wants to publish a claim in a manner that allows any interested third party to verify the claim at a later time, without interacting with the prover.

Claim 4.5.4 (Noninteractive Query Delegation). *In the common random string model, under Conditions 4.5.1, $\mathcal{H}$ is 1-PAC verifiable using a verifier V and prover P such that:*

- *V and P both have access to $f_{\mathcal{D}}$ and to a CRS that provides random samples from $\mathcal{D}_{\mathcal{X}}$.*
- *V uses $O\left(\frac{\log(\frac{1}{\delta})}{\varepsilon}\right)$ queries from $f_{\mathcal{D}}$.*
- *P uses m queries from $f_{\mathcal{D}}$.*
- *V runs in time $O(t(\frac{\varepsilon}{4},\frac{\delta}{2}))$, and P runs in time $O(m)$.*
- *The protocol consists of a single messages of m bits sent from P to V.*

Remark 4.5.5. *In the above claims, we have reduced the sample or query complexity of the verifier compared to PAC learning, but the time complexity is modestly increased. In some cases, it might be possible to combine query delegation with existing general-purpose delegation of computation protocols, to reduce the time complexity as well.*

Protocols and proofs for these claims appear in Appendix B.2. The main issue to notice is that the prover can always be a little bit dishonest, and therefore the verifier must be able to PAC learn in the presence of a small amount of *adversarial* noise. This difficulty is overcome by using the fact that any ERM algorithm is robust with respect to a small amount of adversarial noise.

4.6 Directions for Future Work

This work initializes the study of verification in the context of machine learning. We have seen separations between the sample complexity of verification versus learning and testing, a protocol that uses interaction to efficiently learn sparse boolean functions, and have seen that in some cases the sample complexities of verification and learning are the same.

Building a theory that can help guide verification procedures is a main objective for future research. A specific approach is to identify dimension-like quantities that describe the sample complexity of verification, similarly to role VC dimension plays in characterizing learnability. A different approach is to understand the trade-offs between the various resources in the system – the amount of time, space and samples used by the prover and the verifier, as well as the amount of interaction between the parties.

From a practical perspective, we described potential applications for delegation of machine learning, and for verification of experimental data. It seems beneficial to build efficient verification protocols for machine learning problems that are commonly used in practice, and for the types of scientific experiments mentioned in Appendix B.1. This would have commercial and scientific applications.

There are also some technical improvements that we find interesting. For example, is there a simple way to improve the MA-like protocol for the multi-thresholds class $\mathcal{T}_d$ to achieve 1-PAC verification (instead of 2-PAC verification)?

Finally, seeing as learning verification is still a new concept, it would be good to consider alternative formal definitions, investigate how robust our definition is, and discuss what the "right" definition should be. One case has $\mathcal{O}_V$ and $\mathcal{O}_P$ providing i.i.d. sample access to different distributions, $\mathcal{D}_V$ and $\mathcal{D}_P$ respectively, where $\mathcal{D}_P$ has better quality data in some sense. For instance, for some target function f it might be the case that

$$\mathbb{P}_{(x,y)\sim\mathcal{D}_V}[y = f(x)] < \mathbb{P}_{(x,y)\sim\mathcal{D}_P}[y = f(x)].$$

Can a prover who has access to $\mathcal{D}_P$ efficiently provide an advantage to the verifier? Alternatively, it might be the case that $\mathcal{D}_P$ provides data with "higher resolution" than $\mathcal{D}_V$ (i.e., the σ-algebra of $\mathcal{D}_V$ is a sub-σ-algebra of that of $\mathcal{D}_P$). One can also consider verification in other settings of learning, such as the statistical queries model, clustering, parameter estimation and reinforcement learning.

Chapter 5

Further Results: Verification of Statistical Algorithms

5.1 Introduction

Comparing what can be computed in a given model of computation versus what can be verified in that model is a recurring theme throughout the fields of computability and computational complexity. The most notorious example is of course the P vs. NP problem, which asks whether the set of decision problems that can be solved in polynomial time equals the set of decision problems whose solution can be verified in polynomial time given a suitable proof string. But the same question has been studied for many other settings and models of computation as well, with prominent examples including L vs. NL (for logspace computation), P vs. $\mathsf{IP} = \mathsf{PSPACE}$ (polytime computation, with an interactive proof) and $\mathsf{MIP}^* = \mathsf{RE}$ (ditto, with multiple quantum provers). The existence of a gap between computing and verifying is sometimes interpreted as capturing the notion of *creativity*, in the sense that finding a solution to a problem might require discovery or inventiveness, while verifying a formal proof for the same is merely rote work.

While this theme has deep roots in the literature and an appealing interpretation, its parallels for *learning* have only recently been explored for the first time. In the context of PAC[1] learning, Goldwasser et al. (2021) (Chapter 4 in this dissertation) introduced the setting of *PAC verification*, in which an untrusted prover attempts to convince a verifier that a certain classifier has nearly-optimal loss with respect to a fixed unknown distribution from which the verifier can take random samples. Specifically, they work in the agnostic PAC setting, where the objective is to find a hypothesis h that has nearly-optimal loss in the sense

$$L_{\mathcal{D}}^{0\text{-}1}(h) \leq \inf_{h' \in \mathcal{H}} L_{\mathcal{D}}^{0\text{-}1}(h') + \varepsilon, \tag{5.1}$$

[1] Probably Approximately Correct (PAC) is the standard theoretical model for supervised learning, introduced by Vapnik and Chervonenkis (1968, 1971) and Valiant (1984). Agnostic PAC learning is a generalization to the non-realizable case, introduced by Haussler (1992). See also Shalev-Shwartz and Ben-David (2014).

where $L_{\mathcal{D}}^{0\text{-}1}$ denotes 0-1 population loss and $\mathcal{H}$ is some fixed and known hypothesis class (formal definitions appear in Sections 5.1 and 5.2 below).

Seeing as computational gaps are already well-studied, the main novelty in this setting concerns sample complexity gaps. They show that for some hypothesis classes (but not for others) the number of i.i.d. samples necessary to find a hypothesis with nearly-optimal loss is strictly greater than the number of i.i.d. samples necessary for verifying, with the help of an untrusted prover, that a proposed hypothesis has nearly-optimal loss.

Beyond the (substantial) theoretical motivation, this setting could have meaningful (and timely) real-world applications. First, if a sample complexity gap exists then "verifiable data collection + ML as a service" becomes a viable business model. The provider would collect suitable training data from the desired population distribution, execute a chosen ML algorithm, and subsequently prove to the client that the end result is good with respect to the population distribution. The client would only need a small amount of independent data from the population distribution to determine the veracity of the claim. Beyond this, Goldwasser et al. (2021) envision a variety of other applications, such as more efficient schemes for replicating scientific results in the empirical sciences.

Our Contributions

PAC verification is novel territory, and very little is currently known. The current chapter aims to make some modest steps towards charting this landscape. We focus on studying sample complexity gaps between learning and verifying specifically in terms of the dependence on the VC (Vapnik–Chervonenkis) dimension. We start with showing a lower bound for the sample complexity gap. Prior to our work, one could imagine that some classes would give rise to very large gaps, e.g., $O(\log(d))$ i.i.d. samples for verifying vs. the $\Theta(d)$ samples that are known to be necessary and sufficient for learning, where $d = \mathsf{VC}(\mathcal{H})$. Our first result shows that the gap can be at most quadratic. Namely, for any hypothesis class, PAC verification requires that the verifier use at least $\Omega\left(\sqrt{d}\right)$ i.i.d. random samples.

Second, we show that our lower bound's dependence on the VC dimension is tight in some cases, by improving upon a result of Goldwasser et al. (2021) to obtain a PAC verifier for the class of unions of intervals on $\mathbb{R}$ that uses $O\left(\sqrt{d}\right)$ i.i.d. random samples. The previous result was an upper bound for a weaker notion of verification, that guarantees only that $L_{\mathcal{D}}^{0\text{-}1}(h) \leq 2 \cdot \mathsf{Opt} + \varepsilon$, where $\mathsf{Opt} = \inf_{h' \in \mathcal{H}} L_{\mathcal{D}}^{0\text{-}1}(h')$ (instead of $\mathsf{Opt} + \varepsilon$ as in Eq. (5.1)). Their result applied only to a specific restriction of the class of unions of intervals, while our technique works for the restricted and for the unrestricted versions of the class.

Third, we take a step towards making the notion of PAC verification more applicable in practical settings. Many ML and data science algorithms that people use in practice, and might like to delegate to an untrusted service, do not obtain (or at least do not provably obtain) the objective of agnostic PAC learning as in Eq. (5.1). Instead, they obtain some quantity of loss which is typically good enough in practice. With this reality in mind, we introduce a generalization of PAC verification that guarantees that the outcome is competitive with a specific algorithm. Namely, the verifier guarantees that with high probability, the hypothesis h satisfies $L_{\mathcal{D}}^{0\text{-}1}(h) \leq \mathbb{E}[L_{\mathcal{D}}^{0\text{-}1}(h_A)] + \varepsilon$, where h_A is the (possibly randomized) output of the algorithm (see Definition 5.2.3).

Fourth, we study PAC verification of statistical query algorithms. For a batch $\mathbf{q}$ of statistical queries, we define a notion of *partition size*, denoted $\mathsf{PS}(\mathbf{q})$, which is the number of atoms in the σ-algebra generated by $\mathbf{q}$. We show that whenever this quantity is sufficiently small, there is a sample complexity gap between execution and verification of the statistical query algorithm.

Lastly, we show that there exists a sample complexity gap for a natural example we present, of optimizing a portfolio with advice. Both our lower bound and our upper bound apply to this example.

Related Works

The study of interactive proofs for properties of distributions was initiated by Chiesa and Gur (2018). They showed general bounds in terms of the support size. However, they did not consider tighter bounds that depend on combinatorial characterizations of the distribution testing property of interest (e.g., bounds that depend on the VC dimension).

The study of PAC verification of a hypothesis class was introduced by Goldwasser et al. (2021), who considered interactive proofs for properties of distributions in the specific context of machine learning. In particular, they also considered the relationship between the VC dimension of the class and the sample complexity of verification. They showed a lower bound that is incomparable with our lower bound, and they showed an upper bound for unions of intervals which is weaker than our upper bound. Our definition of PAC verification of an algorithm is closely modeled on their definition.

Recently, there have been a number of works on the general theme of distribution testing and interactive proofs for properties of distributions in the context of machine learning. These include Canetti and Karchmer (2021), Anil, Zhang, Wu, and Grosse (2021), Rubinfeld and Vasilyan (2023) and Herman and Rothblum (2022), among others. Caro, Hinsche, Ioannou, Nietner, and Sweke (2023) studied PAC verification with a quantum prover. Seshia, Sadigh, and Sastry (2022) survey the use of formal methods for verification of AI systems.

Preliminaries

Notation 5.1.1. $\mathbb{N} = \{1, 2, 3, \dots\}$, *i.e.,* $0 \notin \mathbb{N}$. *For any* $n \in \mathbb{N}$, *we denote* $[n] = \{1, 2, 3, \dots, n\}$.

Notation 5.1.2. *For a set* Ω, *we write* $\Delta(\Omega)$ *to denote the set of all probability measures defined on the measurable space* $(\Omega, \mathcal{F})$, *where* $\mathcal{F}$ *is some fixed* σ*-algebra that is implicitly understood.*

Definition 5.1.3. *Let* $\mathcal{P}, \mathcal{Q}$ *be probability measures defined on a measurable space* $(\Omega, \mathcal{F})$. *The total variation distance between* $\mathcal{P}$ *and* $\mathcal{Q}$ *is* $\mathsf{TV}(\mathcal{P}, \mathcal{Q}) = \sup_{A \in \mathcal{F}} |\mathcal{P}(A) - \mathcal{Q}(A)|$.

PAC Learning

Definition 5.1.4. *Let $\mathcal{X}$ be a set, and let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a set of functions. Let $k \in \mathbb{N}$, $X = \{x_1, x_2, \dots, x_k\} \subseteq \mathcal{X}$. We say that $\mathcal{H}$ shatters X if for any $y_1, y_2, \dots, y_k \in \{0,1\}$ there exists $h \in \mathcal{H}$ such that $h(x_i) = y_i$ for all $i \in [k]$. The Vapnik–Chervonenkis (VC) dimension of $\mathcal{H}$, denoted $\mathsf{VC}(\mathcal{H})$, is the largest $d \in \mathbb{N}$ for which there exist a set $X \subseteq \mathcal{X}$ of cardinality d that is shattered by $\mathcal{H}$. If $\mathcal{H}$ shatters sets of cardinality arbitrarily large, we say that $\mathsf{VC}(\mathcal{H}) = \infty$.*

Throughout most of this chapter we use loss functions of the type common in PAC learning, where the loss of a hypothesis with respect to a distribution is defined as the expected loss of that hypothesis on a randomly drawn sample form the distribution, as follows.

Definition 5.1.5. *Let Ω and $\mathcal{H}$ be sets. A loss function is a function $L: \ \Omega \times \mathcal{H} \to [0,1]$. Let $h \in \mathcal{H}$, and let $S = (z_1, \dots, z_m) \in \Omega^m$ be a vector. The empirical loss of h with respect to S is $L_S(h) = \frac{1}{m}\sum_{i \in [m]} L(z_i, h)$. For any distribution $\mathcal{D} \in \Delta(\Omega)$, the loss of h with respect to $\mathcal{D}$ is $L_{\mathcal{D}}(h) = \mathbb{E}_{Z \sim \mathcal{D}}[L(Z, h)]$. The loss of $\mathcal{H}$ with respect to $\mathcal{D}$ is $L_{\mathcal{D}}(\mathcal{H}) = \inf_{h' \in \mathcal{H}} L_{\mathcal{D}}(h')$.*

The 0-1 loss, denoted $L^{\text{0-1}}$, is the special case in which $\mathcal{X}$ is a set, $\Omega = \mathcal{X} \times \{0,1\}$, $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, and $L((x,y),h) = \mathbb{1}(h(x) \neq y)$.

However, in Definition 5.2.3 below we also consider more general types of loss.

Definition 5.1.6. *Let $\mathcal{X}$ be a set, and let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a class of hypotheses. We say that $\mathcal{H}$ is agnostically PAC learnable if there exist an algorithm A and a function $m_A: \ [0,1]^2 \to \mathbb{N}$ such that for any $\varepsilon, \delta \in (0,1)$ and any distribution $\mathcal{D} \in \Delta(\mathcal{X} \times \{0,1\})$, if A receives as input a tuple of $m_A(\varepsilon, \delta)$ i.i.d. samples from $\mathcal{D}$, then A outputs a function $h \in \mathcal{H}$ satisfying*

$$\mathbb{P}\Big[L_{\mathcal{D}}^{\text{0-1}}(h) \leq L_{\mathcal{D}}^{\text{0-1}}(\mathcal{H}) + \varepsilon\Big] \geq 1 - \delta.$$

In words, this means that h is probably (with confidence $1-\delta$) approximately correct (has loss at most ε worse than optimal). The point-wise minimal such function m_A is called the sample complexity of $\mathcal{H}$.

PAC Verification of a Hypothesis Class

Definition 5.1.7 (PAC Verification of a Hypothesis Class; a special case of Definition 4.1.22)**.** *Let $\mathcal{X}$ be a set, let $\mathbb{D} \subseteq \Delta(\mathcal{X} \times \{0,1\})$ be a set of distributions, and let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a class of hypotheses. We say that $\mathcal{H}$ is PAC verifiable with respect to $\mathbb{D}$ using random samples if there exist an interactive proof system consisting of a verifier V and an honest prover P such that for any $\varepsilon, \delta \in (0,1)$ there exist $m_V, m_P \in \mathbb{N}$ such that for any $\mathcal{D} \in \mathbb{D}$, the following conditions are satisfied:*

- ***Completeness.*** *Let the random variable*

$$h_V = [V(S_V, \varepsilon, \delta), P(S_P, \varepsilon, \delta)] \in \mathcal{H} \cup \{\mathsf{reject}\}$$

denote the output of V after receiving input $(S_V, \varepsilon, \delta)$ and interacting with P, which received input $(S_P, \varepsilon, \delta)$. Then

$$\mathbb{P}_{S_V \sim \mathcal{D}^{m_V}, S_P \sim \mathcal{D}^{m_P}}\left[h_V \neq \mathsf{reject} \;\wedge\; \left(L_{\mathcal{D}}^{0\text{-}1}(h_V) \leq L_{\mathcal{D}}^{0\text{-}1}(\mathcal{H}) + \varepsilon\right)\right] \geq 1 - \delta.$$

- ***Soundness.*** *For any (possibly malicious and computationally unbounded) prover P' (which may depend on $\mathcal{D}$, ε, and δ), the verifier's output $h_V = [V(S_V, \varepsilon, \delta), P']$ satisfies*

$$\mathbb{P}_{S_V \sim \mathcal{D}^{m_V}, S_P \sim \mathcal{D}^{m_P}}\left[h_V = \mathsf{reject} \;\vee\; \left(L_{\mathcal{D}}^{0\text{-}1}(h_V) \leq L_{\mathcal{D}}^{0\text{-}1}(\mathcal{H}) + \varepsilon\right)\right] \geq 1 - \delta.$$

In both conditions, the probability is over the randomness of the samples S_V and S_P, as well as the randomness of V, P and P'.

5.2 Technical Overview

Bounds for Verification of VC Classes

Our first result is a lower bound for the number of i.i.d. random samples the verifier requires to successfully PAC verify a class.

Theorem 5.2.1. *There exist constants $C, c > 0$ as follows. Let $\varepsilon \in (0,1)$, $\delta = 1/3$, let $\mathcal{X}$ be a set, and let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class with $\mathsf{VC}(\mathcal{H}) = d \in \mathbb{N}$. Assume that (V, P) is an interactive proof system that PAC verifies $\mathcal{H}$ with parameters (ε, δ) with respect to the set of all distributions $\mathbb{D} = \Delta(\mathcal{X} \times \{0,1\})$, and the verifier V uses $m_V = m_V(d, \varepsilon)$ i.i.d. labeled samples. Then $m_V(d, \varepsilon) \geq (C \cdot \sqrt{d} - c)/\varepsilon^2$.*

Proof Idea. This is an application of Le Cam's 'point vs. mixture' method (see Yu, 1997), together with a reduction from distribution testing to PAC verification. Consider distributions where the marginal over the domain is uniform on a fixed $\mathcal{H}$-shattered set of size d. PAC verification requires distinguishing the case of truly random labels (where the loss of the class is $1/2$), from the case where the labels are ε-biased (and the loss of the class is $1/2 - \varepsilon$). An $\Omega\left(\sqrt{d}/\varepsilon^2\right)$ lower bound for distinguishing these two cases is due to Paninski (2008). □

Our second result shows that the lower bound's dependence on d is tight for a specific class.

Theorem 5.2.2. *Let $d \in \mathbb{N}$, and let*

$$\mathcal{H}_d = \left\{\mathbb{1}_X : \; X = \bigcup_{i \in [d]} [a_i, b_i] \;\wedge\; (\forall i \in [d] : \; 0 \leq a_i \leq b_i \leq 1)\right\} \subseteq \{0,1\}^{[0,1]}$$

be the class of boolean-valued functions over the domain $[0,1]$ that are indicator functions for a union of d intervals. There exists an interactive proof system that PAC verifies the class

$\mathcal{H}_d$ with respect to the set of all distributions over $[0,1] \times \{0,1\}$, such that the verifier uses $m_V = O\Big(\sqrt{d}\log(1/\delta)\varepsilon^{-2.5}\Big)$ random samples, the honest prover uses

$$m_P = O\Big((d^2 \log(d/\varepsilon) + \log(1/\delta))\varepsilon^{-4}\Big)$$

random samples, and both the verifier and the honest prover run in time polynomial in their numbers of samples.

Proof Idea. A discretization of the population distribution is induced by partitioning the domain $[0,1]$ into d/ε intervals, each of which has weight ε/d according to the population distribution. In the discretized distribution, the probability mass from each interval is lumped together into a single arbitrary point in that interval. We show that to find an ε-sub-optimal union of intervals, it suffices to know this discretized distribution. The prover sends the (purported) discretized distribution to the verifier. The verifier uses a distribution identity tester to verify that the provided distribution is a correct discretization of the population distribution. This is possible using $O\Big(\sqrt{d}\Big)$ samples, because the support of the discretized distribution is of size $O(d)$. □

Verification of Statistical Algorithms

Many popular algorithms do not come with provable PAC-like guarantees, but tend to work well in practice. Such heuristics are common in machine learning, data science, optimization, operations research, finance, etc. People might like to delegate the task of collecting data and executing an algorithm on that data to an untrusted party. To capture this notion, our next contribution is a new definition of PAC verification of an *algorithm*.[2] This generalizes the definition of PAC verification of a *hypothesis class* (Definition 5.1.7, introduced by Goldwasser et al., 2021), which corresponds to the special case of PAC verifying an algorithm that is an agnostic PAC learner for the class.

Definition 5.2.3 (PAC Verification of an Algorithm)**.** *Let Ω be a set, let $\mathbb{D} \subseteq \Delta(\Omega)$ be a set of distributions, let $\mathcal{H}$ be a set (called the set of* possible outputs*), and for each $\mathcal{D} \in \mathbb{D}$ let $\mathcal{O}_\mathcal{D}$ be an oracle. Let A be a (possibly randomized) algorithm that takes no inputs, has query access to $\mathcal{O}_\mathcal{D}$, and outputs a value $h_A = A^{\mathcal{O}_\mathcal{D}} \in \mathcal{H}$. Let $L: \mathbb{D} \times (\mathcal{H} \cup \{\mathsf{reject}\}) \to [0,1]$ be an arbitrary function[3], let $L_\mathcal{D}(\cdot)$ denote $L(\mathcal{D}, \cdot)$, and let $L_\mathcal{D}(A) = \mathbb{E}[L_\mathcal{D}(h_A)]$, where the expectation is over the randomness of A and of the oracle $\mathcal{O}_\mathcal{D}$. We say that the algorithm A with access to oracles $\{\mathcal{O}_\mathcal{D}\}_{\mathcal{D}\in\mathbb{D}}$ is PAC verifiable with respect to $\mathbb{D}$ by a verification protocol that uses random samples if there exist an interactive proof system consisting of a verifier V and an honest prover P such that for any $\varepsilon, \delta \in (0,1)$ there exist $m_V, m_P \in \mathbb{N}$ such that for any $\mathcal{D} \in \mathbb{D}$, the following conditions are satisfied:*

[2] This notion differs from delegation of computation, in that the data (the input to the algorithm) is collected by the untrusted prover.

[3] Note that this is more general than in Definition 5.1.5.

- ***Completeness.*** *Let the random variable*

$$h_V = [V(S_V, \varepsilon, \delta), P(S_P, \varepsilon, \delta)] \in \mathcal{H} \cup \{\mathsf{reject}\}$$

denote the output of V after receiving input $(S_V, \varepsilon, \delta)$ and interacting with P, which received input $(S_P, \varepsilon, \delta)$. Then

$$\mathbb{P}_{S_V \sim \mathcal{D}^{m_V}, S_P \sim \mathcal{D}^{m_P}} [h_V \neq \mathsf{reject} \ \wedge \ L_{\mathcal{D}}(h_V) \leq L_{\mathcal{D}}(A) + \varepsilon] \geq 1 - \delta.$$

- ***Soundness.*** *For any deterministic or randomized (possibly malicious and computationally unbounded) prover P' (which may depend on $\mathcal{D}$, ε, δ and $\{\mathcal{O}_{\mathcal{D}}\}_{\mathcal{D} \in \mathbb{D}}$), the verifier's output $h = [V(S_V, \varepsilon, \delta), P']$ satisfies*

$$\mathbb{P}_{S_V \sim \mathcal{D}^{m_V}} [h_V = \mathsf{reject} \ \vee \ L_{\mathcal{D}}(h_V) \leq L_{\mathcal{D}}(A) + \varepsilon] \geq 1 - \delta.$$

The probabilities are over the randomness of V, P and P' and of the samples S_V and S_P.

In other words, whereas the definition of Goldwasser et al. (2021) required that the interactive proof system guarantee that a hypothesis is competitive with respect to any hypothesis in $\mathcal{H}$, our definition requires that it be competitive with respect to a specific algorithm.

Remark 5.2.4. *PAC verification of an algorithm A requires that $L_{\mathcal{D}}(h_V) \leq \mathsf{Opt}_A + \varepsilon$ with high probability. Two natural candidate definitions for Opt_A include (1) $\mathsf{Opt}_A = L_{\mathcal{D}}(h_A)$, and (2) $\mathsf{Opt}_A = \mathbb{E}[L_{\mathcal{D}}(h_A)]$. Candidate (1) requires that with high probability the verifier's output be at most ε worse than the output of executing algorithm A, while (2) requires that it be at most ε worse than the expected loss of A.*

The loss $L_{\mathcal{D}}(h_A)$ is a random variable that depends, inter alia, on the random samples used by A (more generally: on the randomness of the oracle used by A). A crucial aspect of PAC verification is that the verifier use less random samples than are necessary for executing A, and in particular it cannot access the random samples used by A. So the verifier cannot know what loss was obtained in any particular execution of A. Therefore, we reject candidate (1) and adopt candidate (2).

As an application of this new definition, we show that some statistical query algorithms (see Definitions C.2.1 and C.2.3) can be PAC verified via a protocol in which the verifier uses less i.i.d. samples than would be required for simulating the statistical query oracle used by the algorithm. Specifically, for a batch $\mathbf{q}$ of statistical queries, the *partition size* $\mathsf{PS}(\mathbf{q})$ is the number of atoms in the σ-algebra generated by $\mathbf{q}$. If the algorithm uses only batches with small partition size then verification is cheap, as in the following theorem.

Theorem (Informal version of Theorem C.2.8)**.** *Let A be a statistical query algorithm that adaptively generates at most b batches of queries with precision τ such that each batch $\mathbf{q}$*

satisfies $\mathsf{PS}(\mathbf{q}) \leq s$*. Then* A *is PAC verifiable by an interactive proof system where the verifier uses*

$$m_V = \Theta\left(\frac{\sqrt{s}\log(b/\varepsilon\delta)}{\tau^2} + \frac{\log(1/\varepsilon\delta)}{\varepsilon^2}\right)$$

i.i.d. samples.

Proof Idea. The verifier simulates algorithm A. Each time A sends a batch of queries to be evaluated by the statistical query oracle, the verifier sends the queries to the prover, and the prover sends back a vector of purported evaluations. The verifier uses $O(\sqrt{s}/\tau^2)$ i.i.d. random samples to execute a distribution identity tester (Theorem 5.4.1) to verify that the prover's evaluations are correct up to the desired accuracy τ. □

In particular, Theorem C.2.8 implies the following separation:

Corollary (Informal version of Corollary C.2.9)**.** *Let* $d \in \mathbb{N}$ *and let* A *be a statistical query algorithm such that each batch of queries generated by* A *corresponds precisely to a* σ*-algebra with* d *atoms. Then simulating* A *using random samples requires* $\Omega(d/\tau^2)$ *random samples, but there exists a PAC verification protocol for* A *where the verifier uses* $O\left(\sqrt{d}/\tau^2\right)$ *random samples.*

Examples

Example 5.2.5 (Optimizing a portfolio with advice)**.** Consider a task in which an agent selects a subset S consisting of n items from the set $\Omega = [2n]$. Subsequently, an item $i \in \Omega$ is chosen at random according to a distribution $\mathcal{D} \in \Delta(\Omega)$ that is unknown to the agent, and the agent experiences loss $L(i, S) = \mathbb{1}(i \notin S)$.

To help make an optimal decision, the agent has access to an i.i.d. sample $Z = (z_1, \ldots, z_m) \sim \mathcal{D}^m$. Let $\mathcal{H} = \binom{\Omega}{n}$ denote the collection of subsets of size n that the agent could select. $\mathsf{VC}(\mathcal{H}) = n$, and therefore estimating the expected loss $L_{\mathcal{D}}(S)$ of each possible choice $S \in \mathcal{H}$ up to precision $\varepsilon > 0$ requires $m_A = \Omega((n + \log(1/\delta))/\varepsilon^2)$ samples.

By Corollary C.2.9, if the agent can receive advice from an untrusted prover, it can make an ε-optimal choice using $m_V = O(\sqrt{n}\log(1/\delta\varepsilon)/\varepsilon^2)$ i.i.d. samples. Note that $m_V \ll m_A$ for large n. Furthermore, our expression for m_V is tight in the sense that, by Theorem 5.2.1, $\Omega(\sqrt{n})$ samples are necessary for verifying the advice of an untrusted prover. □

Note that the above example is an instance of verification in our generalized setting (Definition 5.2.3), but it is technically not an instance of PAC verification as previously defined by Goldwasser et al. (2021), e.g., because the distribution has no labels. More generally, Definition 5.2.3 includes verification of *distribution learning*, as follows.

Example 5.2.6 (Verification of distribution learning)**.** Let $\Omega = [n]$. Consider a task in which an agent has access to an i.i.d. sample $Z = (z_1, \ldots, z_m) \sim \mathcal{D}^m$ from some distribution

$\mathcal{D} \in \Delta(\Omega)$ that is unknown to the agent. The agent selects a distribution $\hat{\mathcal{D}} \in \Delta(\Omega)$, and experience loss $L_{\mathcal{D}}\left(\hat{\mathcal{D}}\right) = \mathsf{TV}\left(\hat{\mathcal{D}}, \mathcal{D}\right)$.

It is well known that to achieve loss at most ε with probability at least $1 - \delta$, it is necessary and sufficient to take $m_A = \Theta((n + \log(1/\delta))/\varepsilon^2)$ samples Canonne (2020, Theorem 1). In contrast, if the agent has access to advice from an untrusted prover then $m_V = O(\sqrt{n}\log(1/\delta)\varepsilon^{-2})$ i.i.d. samples are sufficient. The honest prover simply sends the verifier a description of a distribution $\tilde{\mathcal{D}} \in \Delta(\Omega)$ that has loss at most $\varepsilon/\sqrt{n}$. The verifier uses distribution testing (Theorem 5.4.1) to decide whether $L_{\mathcal{D}}\left(\tilde{\mathcal{D}}\right) \leq \varepsilon/\sqrt{n}$ or $L_{\mathcal{D}}\left(\tilde{\mathcal{D}}\right) \geq \varepsilon$, and accepts if and only if the former case holds. □

A large collection of concrete tasks that might be of interest and that fall within the setting of Definition 5.2.3 involve solving various problems on graphs given random samples that convey information about the graph, as follows.

Example 5.2.7 (Verification in graphs)**.** Fix $n \in \mathbb{N}$. For any graph $G = (V, E)$ with $V = [n]$, let $\mathcal{D}_G$ be the uniform distribution on E. The agent does not know G, but it knows n and it has access to an i.i.d. sample $Z = (z_1, \ldots, z_m) \sim \mathcal{D}_G^m$. Consider some standard tasks, such as:

- Maximum matching. The agent selects a subset $M \subseteq \binom{V}{2}$ and experiences loss

$$L_{\mathcal{D}_G}(M) = \min_{M' \in \mathcal{M}} \frac{|M \Delta M'|}{n},$$

where $\mathcal{M}$ is the set of all matchings in G of maximal size.

- Coloring. The agent selects a function $f : V \to \mathbb{N}$ and experiences loss

$$L_{\mathcal{D}_G}(f) = \min_{f' \in \mathcal{F}} \frac{\sum_{v \in V} \mathbb{1}\left(f(v) \neq f'(v)\right)}{n}$$

where $\mathcal{F}$ is the set of all valid colorings of G that use a minimal number of colors.

For these tasks, there is an easy lower bound of $m = \Omega(n)$ on the number of samples the agent needs to guarantee loss at most ε with probability at least $1 - \delta$ for $\varepsilon = \delta = 0.1$. To see this, consider the family of graphs that consist of a disjoint union of triplets (sets of three vertices), such that each triplet contains a single edge. Because the agent does not know in advance where the edge is in each triplet, finding an approximately maximum matching and an approximate 2-coloring require seeing nearly all the edges in the graph.

However, if we assume that G has maximum degree bounded by a constant (as in the lower bound), then $\mathcal{D}_G$ is a uniform distribution with support size $O(n)$. Hence, given access to advice from an untrusted prover, the agent can solve these tasks using $O(\sqrt{n})$ samples using the verification procedure of Example 5.2.6.

To see that $\Omega(\sqrt{n})$ samples are necessary for verification with the help of a prover, consider a family of graphs consisting of a disjoint union of triplets as above, but where only half the triplets contain an edge. Distinguishing between this family and the previous family requires observing a collision (receiving a sample that contains the same edge twice), which requires $\Omega(\sqrt{n})$ samples by the 'birthday paradox'. □

So far, all our examples involved a quadratic gap between learning and verifying. However, larger gaps are possible if we make strong assumptions on the unknown distribution. One example of this, pointed out by Goldwasser et al. (2021), is that the gap between learning and verifying for *realizable* PAC learning is unbounded. Unbounded gaps can exist also for other tasks as well, as in the following example.

Example 5.2.8 (Unbounded gap in a graph task)**.** Let n, $G = (V, E)$, and $\mathcal{D}_G$ be as in Example 5.2.7. Consider the maximal matching tasks under the assumption that E is a perfect matching. Again, there is an easy lower bound of $\Omega(n)$ random samples to guarantee loss at most ε with probability at least $1-\delta$ for $\varepsilon = \delta = 0.1$ without the help of a prover. To see this, consider a graph that is a disjoint union of sets of four vertices, where each such set contains two disjoint edges. Finding a perfect matching requires seeing an edge from each set.

In contrast, $m_V = O(\log(1/\delta)/\varepsilon)$ samples are sufficient given advice from an untrusted prover. The protocol is as follows. The prover sends $\tilde{E}$, which purportedly equals E. If $\tilde{E}$ is not a perfect matching then the verifier rejects. Then, the verifier takes m_V samples from $\mathcal{D}_G$, and accepts if and only if all the edges in the sample appear in $\tilde{E}$. For completeness, if $\tilde{E} = E$ then the verifier always accepts. For soundness, if $\left(|\tilde{E}\Delta E|\right)/n \geq \varepsilon$, then $\mathcal{D}_G$ has weight $\Omega(\varepsilon)$ on edges that are not in $\tilde{E}$, and so taking m_V samples is sufficient to ensure that the verifier rejects with probability at least $1-\delta$. □

For the maximum matching task, we have seen that under the assumption that G has maximum degree bounded by a constant the sample complexity gap is quadratic, but that the gap is unbounded under the stronger assumption that G is a perfect matching. We view this as a demonstration of the richness of this setting.

5.3 A Lower Bound for PAC Verification of VC Classes

Theorem 5.2.1 is proved via a reduction from the following distribution testing lower bound.

Theorem 5.3.1 (Reformulation of Theorem 4 in Paninski, 2008)**.** *Let $d, t \in \mathbb{N}$ and let $\varepsilon \in (0,1)$. For every $\sigma \in \Sigma = \{\pm 1\}^d$, let $\mathcal{D}_{\sigma,\varepsilon} \in \Delta([2d])$ be a distribution such that for all $i \in [d]$,*

$$\mathcal{D}_{\sigma,\varepsilon}(2i-1) = \frac{1+\sigma_i \cdot \varepsilon}{2d}, \quad \textit{and} \quad \mathcal{D}_{\sigma,\varepsilon}(2i) = \frac{1-\sigma_i \cdot \varepsilon}{2d}.$$

Let $\mathcal{D}_{\Sigma,\varepsilon,t}$ be the distribution over $[2d]^t$ generated by selecting a vector $\sigma \in \Sigma$ uniformly at random, and then taking t i.i.d. samples from $\mathcal{D}_{\sigma,\varepsilon}$. Let $\mathcal{D}_{U,t} = \mathrm{U}([2d])^t$ be the distribution over $[2d]^t$ generated by selecting t i.i.d. uniform samples from $[2d]$. Then $\mathsf{TV}(\mathcal{D}_{U,t}, \mathcal{D}_{\Sigma,\varepsilon,t}) \leq$

$f_{\mathsf{Paninski}}(t, \varepsilon, d)$ for

$$f_{\mathsf{Paninski}}(t, \varepsilon, d) = \frac{1}{2} \cdot \left(\exp\left(\frac{t^2 \varepsilon^4}{d} \right) - 1 \right)^{1/2}.$$

The proof also uses the following well-known fact about maximal couplings (see e.g. Lemma 4.1.13 in Roch, 2023).

Theorem 5.3.2. *Let Ω be a set, and let $p_X, p_Y \in \Delta(\Omega)$ be distributions. Then*

$$\mathsf{TV}(p_X, p_Y) = \\ \inf \Big\{ \mathbb{P}[X \neq Y] : \ (X, Y) \textit{ is a joint distribution with marginals } X \sim p_X \textit{ and } Y \sim p_Y \Big\}.$$

Proof of Theorem 5.2.1. Let $X = \{x_1, \dots, x_d\} \subseteq \mathcal{X}$ be a set of size d that is shattered by $\mathcal{H}$ (such a set exists because $\mathsf{VC}(\mathcal{H}) = d$). Let $\mathcal{D}_U = \mathrm{U}(X \times \{0,1\})$.

For every $h \in \mathcal{H}_X = \{0,1\}^X$, let $\mathcal{D}_{h,4\varepsilon} \in \Delta(X \times \{0,1\})$ be a distribution such that

$$\forall\, (x, y) \in X \times \{0,1\} : \ \mathcal{D}_{h,4\varepsilon}\big((x,y)\big) = \begin{cases} (1+4\varepsilon)/2d & h(x) = y \\ (1-4\varepsilon)/2d & h(x) \neq y \end{cases}.$$

Consider a (possibly randomized) testing algorithm T that takes t i.i.d. samples from an unknown distribution $\mathcal{D}$ and decides correctly with probability at least $1-\beta$ whether $\mathcal{D} = \mathcal{D}_U$ or whether $\mathcal{D} \in \{\mathcal{D}_{h,4\varepsilon} : \ h \in \mathcal{H}_X\}$ (if $\mathcal{D}$ is not one of these $|\mathcal{H}_X| + 1$ options then we make no assumptions regarding the behavior of T).

Let $\mathcal{D}_{U,t} = (\mathcal{D}_U)^t$ and let $\mathcal{D}_{\mathcal{H}_X,4\varepsilon,t}$ be the distribution generated by selecting $h \in \mathcal{H}_X$ uniformly at random and then taking t i.i.d. samples from $\mathcal{D}_{h,4\varepsilon}$. By Theorem 5.3.1, $\mathsf{TV}(\mathcal{D}_{U,t}, \mathcal{D}_{\mathcal{H}_X,4\varepsilon,t}) \leq f_{\mathsf{Paninski}}(t, 4\varepsilon, d)$. By Theorem 5.3.2, for every $\alpha > 0$ there exists a joint distribution $(S_U, S_{\mathcal{H}})$ such that $S_U \sim \mathcal{D}_{U,t}$, $S_{\mathcal{H}} \sim \mathcal{D}_{\mathcal{H}_X,4\varepsilon,t}$, and $\mathbb{P}[S_U \neq S_{\mathcal{H}}] \leq f_{\mathsf{Paninski}}(t, 4\varepsilon, d) + \alpha$.

For any such α and $(S_U, S_{\mathcal{H}})$, no tester can distinguish with probability strictly greater than $1/2$ between S_U and $S_{\mathcal{H}}$ in the event where $S_U = S_{\mathcal{H}}$. Hence,

$$\beta \geq 1/2 \cdot \mathbb{P}[S_U = S_{\mathcal{H}}] = 1/2 \cdot (1 - \mathbb{P}[S_U \neq S_{\mathcal{H}}]) \geq 1/2 \cdot (1 - f_{\mathsf{Paninski}}(t, 4\varepsilon, d) - \alpha).$$

Taking $\alpha \to 0$ and rearranging yields

$$t \geq \frac{\sqrt{d \cdot \ln(1 + (4\beta - 2)^2)}}{\varepsilon^2}. \tag{5.2}$$

This establishes a lower bound on the sample complexity for the $\mathcal{D}_U$ vs. $\{\mathcal{D}_{h,4\varepsilon} : \ h \in \mathcal{H}_X\}$ distribution testing problem.

Next, we show a reduction from the distribution testing problem to PAC verification of $\mathcal{H}$. Let (V, P) be an interactive proof system that PAC verifies $\mathcal{H}$ such that the verifier V and honest prover P use m_V and m_P i.i.d. samples from the unknown distribution respectively, and satisfy Definition 5.1.7 with parameters ε and δ, as in the statement of Theorem 5.2.1. Using (V,P), we construct a tester T for the $\mathcal{D}_U$ vs. $\{\mathcal{D}_{h,4\varepsilon} : h \in \mathcal{H}_X\}$ testing problem. Given sample access to an unknown distribution $\mathcal{D}$ for the testing problem, T operates as follows:

1. Compute $h_V = [V(\mathcal{D}), P(\mathcal{D}_U)]$. Namely, simulate an execution of the PAC verification protocol as follows. Take a sample $S_V \sim \mathcal{D}^{m_V}$ of m_V i.i.d. samples from $\mathcal{D}$, and take a sample $S_P \sim (\mathcal{D}_U)^{m_P}$ of m_P i.i.d. samples from $\mathcal{D}_U$ (seeing as the specification of $\mathcal{D}_U$ is completely known to T, T can generate as many samples from $\mathcal{D}_U$ as necessary using uniform random coins). Execute the PAC verification protocol such that V receives input S_V, P receives input S_P, and the output of the verifier at the end of the protocol is $h_V \in \mathcal{H} \cup \{\mathsf{reject}\}$.

2. Take a sample $S_{\mathsf{test}} \sim \mathcal{D}^\ell$ of $\ell = \lceil \ln(24)/2\varepsilon^2 \rceil < 3/\varepsilon^2$ i.i.d. samples from $\mathcal{D}$.

3. If $(h_V = \mathsf{reject}) \ \vee \ (h_V \neq \mathsf{reject} \ \wedge \ L^{0\text{-}1}_{S_{\mathsf{test}}}(h_V) \leq 1/2 - 2\varepsilon)$ then output "$\mathcal{D} \in \{\mathcal{D}_{h,4\varepsilon} : h \in \mathcal{H}_X\}$". Otherwise, output "$\mathcal{D} = \mathcal{D}_U$".

We argue that the tester T defined in this manner solves the testing problem correctly with probability at least $7/12$. If $\mathcal{D} = \mathcal{D}_U$, then $L^{0\text{-}1}_{\mathcal{D}}(h) = 1/2$ for any $h \in \mathcal{H}$. In particular, if $h_V \neq \mathsf{reject}$ then $L^{0\text{-}1}_{S_{\mathsf{test}}}(h_V) \geq 1/2 - \varepsilon$ with probability at least $11/12$ (by Hoeffding's inequality and the choice of ℓ). Thus, if $\mathcal{D} = \mathcal{D}_U$ then T outputs "$\mathcal{D} = \mathcal{D}_U$" with probability at least $11/12$.

Conversely, if $\mathcal{D} = \mathcal{D}_{h',4\varepsilon}$ for some $h' \in \mathcal{H}_X$, then $L^{0\text{-}1}_{\mathcal{D}}(h) = 1/2 - 4\varepsilon$ for $h \in \mathcal{H}$ such that $h|_X = h'$. From the correctness of the PAC verification protocol, with probability at least $2/3$, either $h_V = \mathsf{reject}$, or $L^{0\text{-}1}_{\mathcal{D}}(h_V) \leq 1/2 - 3\varepsilon$, and in that case with probability at least $11/12$, $L^{0\text{-}1}_{S_{\mathsf{test}}}(h) \leq 1/2 - 2\varepsilon$ (again by Hoeffding's inequality and choice of ℓ). A union bound implies that if $\mathcal{D} = \mathcal{D}_{h',4\varepsilon}$ for some $h' \in \mathcal{H}_X$ then T outputs "$\mathcal{D} \in \{\mathcal{D}_{h,4\varepsilon} : h \in \mathcal{H}_X\}$" with probability at least $1 - 1/3 - 1/12 = 7/12$.

We conclude that T correctly solves the $\mathcal{D}_U$ vs. $\{\mathcal{D}_{h,4\varepsilon} : h \in \mathcal{H}_X\}$ testing problem with probability at least $7/12$ using $t = m_V + \ell$ i.i.d. samples from the unknown distribution $\mathcal{D}$. Plugging $\beta = 5/12$ in Eq. (5.2), this implies that $m_V \geq (0.3 \cdot \sqrt{d} - 3)/\varepsilon^2$, as desired. □

Remark 5.3.3. *A previous version of this chapter (Mutreja and Shafer, 2022) presented a proof of an $\Omega\left(\sqrt{d}\right)$ lower bound, without the dependence on ε. That proof uses a reduction to a simpler distribution testing lower bound based on the 'birthday paradox' (instead of the Paninski bound), and it may be better suited for pedagogical expositions.*

5.4 Verification of Unions of Intervals

We use the following theorem.[4]

[4] See also Goldreich and Ron (2011) and the discussion following Theorem 5.4 in Canonne (2020).

Theorem 5.4.1 (Theorem 1 in Canonne, Jain, Kamath, and Li, 2022)**.** *Let $\varepsilon, \delta \in (0,1)$, let $n \in \mathbb{N}$, and let $\mathcal{P}, \tilde{\mathcal{P}} \in \Delta([n])$ be distributions. There exists a tolerant distribution identity tester that, given a complete description of $\tilde{\mathcal{P}}$ and $m = O(\sqrt{n}\log(1/\delta)\varepsilon^{-2})$ i.i.d. samples from $\mathcal{P}$, satisfies the following:*

- ***Completeness.*** *If $\mathsf{TV}\Big(\mathcal{P}, \tilde{\mathcal{P}}\Big) \leq \varepsilon/\sqrt{n}$ then the tester accepts with probability at least $1-\delta$.*
- ***Soundness.*** *If $\mathsf{TV}\Big(\mathcal{P}, \tilde{\mathcal{P}}\Big) > \varepsilon$ then the tester rejects with probability at least $1-\delta$.*

Definition 5.4.2. *Let $\varepsilon \in [0,1]$, let $\mathcal{X}$ be a set and let $\mathcal{F} \subseteq \{0,1\}^{\mathcal{X}}$ be a set of functions. Let $\mathcal{D} \in \Delta(\mathcal{X})$, and let $S \in \mathcal{X}^m$ for some $m \in \mathbb{N}$. We say that S is an ε-sample for $\mathcal{D}$ with respect to $\mathcal{F}$ if*

$$\forall f \in \mathcal{F}: \quad \left|\frac{|\{x \in S: \ f(x) = 1\}|}{m} - \mathbb{P}_{x\sim\mathcal{D}}[f(x) = 1]\right| \leq \varepsilon.$$

We also use the fundamental uniform convergence result from VC theory.[5]

Theorem 5.4.3 (Vapnik and Chervonenkis, 1968, 1971)**.** *Let $d \in \mathbb{N}$ and $\varepsilon, \delta \in (0,1)$. Let $\mathcal{X}$ be a set and let $\mathcal{F} \subseteq \{0,1\}^{\mathcal{X}}$ be a set of functions with $\mathsf{VC}(\mathcal{F}) = d$. Let $\mathcal{D} \in \Delta(\mathcal{X})$, and let $S \sim \mathcal{D}^m$, where*

$$m = \Omega\left(\frac{d\log({}^{d}/_{\varepsilon}) + \log({}^{1}/_{\delta})}{\varepsilon^2}\right).$$

Then with probability at least $1-\delta$, S is an ε-sample for $\mathcal{D}$ with respect to $\mathcal{F}$.

Proof of Theorem 5.2.2. We show that Protocol C.1 (in Appendix C.1) satisfies the requirements of the theorem. For completeness, note that if the prover follows the protocol then $\tilde{P}_{j,0} + \tilde{P}_{j,1} = {}^{1}/_{k}$ for all j, so the verifier will never reject at the first 'if' statement. Let $\mathcal{B} = \{I_j \times \{y\} : \ j \in [k] \ \wedge \ y \in \{0,1\}\}$, and let $\mathcal{F} = \{\mathbb{1}_E : \ E \in \sigma(\mathcal{B})\} \subseteq \{0,1\}^{[0,1]\times\{0,1\}}$. In words, $\mathcal{F}$ is the set of indicator functions for events in the σ-algebra generated by $\mathcal{B}$. $\mathsf{VC}(\mathcal{F}) = 2k = O(d/\varepsilon)$, so Theorem 5.4.3 and the choice of m_P imply that with probability at least $1 - \delta/2$, S_P is an $\varepsilon/(6\sqrt{2k})$-sample for $\mathcal{D}$ with respect to $\mathcal{F}$. By the definitions of total variation distance and of an ε-sample, this implies that $\mathbb{P}\Big[\mathsf{TV}\Big(\mathcal{P}, \tilde{\mathcal{P}}\Big) \leq \varepsilon/(6\sqrt{2k})\Big] \geq 1 - \delta/2$. From the completeness of the tester of Theorem 5.4.1 and a union bound we conclude that with probability at least $1-\delta$, the verifier does not reject. This establishes completeness.

For soundness, consider two cases.

- The prover is too dishonest, such that $\mathsf{TV}\Big(\mathcal{P}, \tilde{\mathcal{P}}\Big) > \varepsilon/6$. Then by the soundness of the tester of Theorem 5.4.1, the verifier rejects with probability at least $1-\delta/2$.

[5] See also Alon and Spencer (2000), Theorem 13.4.4.

- The prover is sufficiently honest, such that $\mathsf{TV}\Big(\mathcal{P},\tilde{\mathcal{P}}\Big) \leq \varepsilon/6$. Then for any $h' \in \mathcal{H}_d$,

  $$\begin{aligned}\Big|L^{0\text{-}1}_{\mathcal{D}}(h') - L^{0\text{-}1}_{\tilde{\mathcal{P}}}(h')\Big| &\leq \Big|L^{0\text{-}1}_{\mathcal{D}}(h') - L^{0\text{-}1}_{\mathcal{P}}(h')\Big| + \Big|L^{0\text{-}1}_{\mathcal{P}}(h') - L^{0\text{-}1}_{\tilde{\mathcal{P}}}(h')\Big| \\ &\leq \Big|L^{0\text{-}1}_{\mathcal{D}}(h') - L^{0\text{-}1}_{\mathcal{P}}(h')\Big| + \varepsilon/6, \qquad (5.3)\end{aligned}$$

  where the last inequality follows from $\mathsf{TV}\Big(\mathcal{P},\tilde{\mathcal{P}}\Big) \leq \varepsilon/6$.

  Fix $h' \in \mathcal{H}_d$. We argue that $|L^{0\text{-}1}_{\mathcal{D}}(h') - L^{0\text{-}1}_{\mathcal{P}}(h')| \leq \varepsilon/3$. Let $Q = \{x \in [0,1] : \; h'(x) \neq h'(x^*)\}$, where for each $x \in [0,1]$, we define $x^* = x^*_j$ such that $x \in I_j$. Namely, Q is the set of points for which applying the discretization procedure alters the output of h'. Then

  $$\begin{aligned}\Big|L^{0\text{-}1}_{\mathcal{D}}(h') - L^{0\text{-}1}_{\mathcal{P}}(h')\Big| &= \Big|\mathbb{P}_{(x,y)\sim\mathcal{D}}[h'(x) \neq y] - \mathbb{P}_{(x,y)\sim\mathcal{D}}[h'(x^*) \neq y]\Big| \\ &= \Bigg|\mathbb{P}_{(x,y)\sim\mathcal{D}}[h'(x) \neq y \;\wedge\; x \in Q] \\ &\qquad - \mathbb{P}_{(x,y)\sim\mathcal{D}}[h'(x^*) \neq y \;\wedge\; x \in Q]\Bigg| \qquad (5.4) \\ &\leq \mathcal{D}(Q') \qquad (Q' = Q \times \{0,1\}) \\ &\leq \sum_{j\in[k]:\; I_j\cap Q\neq\varnothing} \mathcal{D}(I'_j) \qquad (I'_j = I_j \times \{0,1\}) \\ &= \sum_{j\in[k]:\; I_j\cap Q\neq\varnothing} \mathcal{P}(I'_j) \qquad (\mathcal{D}(I'_j) = \mathcal{P}(I'_j)) \\ &= \mathcal{P}\left(\bigcup\Big\{I'_j : \; I_j \cap Q \neq \varnothing\Big\}\right) \\ &\leq \tilde{\mathcal{P}}\left(\bigcup\Big\{I'_j : \; I_j \cap Q \neq \varnothing\Big\}\right) + \mathsf{TV}\Big(\mathcal{P},\tilde{\mathcal{P}}\Big) \\ &\leq 2d/k + \mathsf{TV}\Big(\mathcal{P},\tilde{\mathcal{P}}\Big) \qquad (5.5) \\ &\leq 2d/k + \varepsilon/6 = \varepsilon/3, \qquad (5.6)\end{aligned}$$

  where Eq. (5.4) holds since the loss of h' can differ between $\mathcal{D}$ and $\mathcal{P}$ only for points in Q; Eq. (5.5) holds because h' consists of d intervals, which together have $2d$ endpoints, $I_j \cap Q \neq \varnothing$ only if I_j contains one of these endpoints, and if the verifier did not reject then $\tilde{\mathcal{P}}(I'_j) = {}^1\!/\!{}_k$ for all j; finally Eq. (5.6) holds by the assumption (in the current case) that the prover is sufficiently honest.

  Combining Eq. (5.6) with Eq. (5.3) yields $\forall h' \in \mathcal{H}_d : \; \Big|L^{0\text{-}1}_{\mathcal{D}}(h') - L^{0\text{-}1}_{\tilde{\mathcal{P}}}(h')\Big| \leq \varepsilon/2$. This implies that a hypothesis h that has minimum loss with respect to $\tilde{\mathcal{P}}$ satisfies $L^{0\text{-}1}_{\mathcal{D}}(h) \leq L^{0\text{-}1}_{\mathcal{D}}(\mathcal{H}) + \varepsilon$.

We conclude that regardless of the prover's behavior, with probability at least $1 - \delta/2$ the verifier either rejects or outputs a hypothesis with excess loss at most ε, as desired. □

Remark 5.4.4. *The dependence of the tolerance parameter in Theorem 5.4.1 on the domain size is quadratic, namely the verifier accepts if* $\mathsf{TV}\Big(\mathcal{P}, \tilde{\mathcal{P}}\Big) \leq \varepsilon/\sqrt{n}$*. Notice that this affects the sample complexity of the honest prover but not of the verifier. For instance, if the tolerance was* ε/e^n *instead of* $\varepsilon/\sqrt{n}$*, the verifier's sample complexity would remain unchanged.*

5.5 Discussion and Future Work

In this chapter, we have shown that $\Omega\Big(\sqrt{d}\Big)$ samples are necessary for PAC verifying a class of VC dimension d, and furthermore, for some classes $O\Big(\sqrt{d}\Big)$ samples are sufficient. In contrast, Theorem 4.4.1 states that there also exist VC classes where the sample complexity for verification is $\tilde{\Omega}(d)$ under the assumption that the verifier is proper (outputs a hypothesis from the class), and we believe it is likely that there exist VC classes for which an $\tilde{\Omega}(d)$ lower bound holds for any verifier.

Hence, it appears likely that the VC dimension does not characterize the sample complexity of PAC verification. In that case, finding an alternative combinatorial quantity that does characterize that sample complexity is an exciting open problem.

A potentially easier problem is to devise upper bounds (PAC verification protocols) for specific classes of interest. For example, the main property of the thresholds class utilized in the proof of Theorem 5.2.2 is that it has low 'surface area' or noise sensitivity (cf. Balcan et al., 2012). Perhaps a similar proof technique could apply to other classes as well.

Additionally, we introduced a notion of PAC verification of an algorithm. We believe this is very natural definition, because many of the algorithms that people might like to delegate in practice are not PAC learners, including unsupervised learning algorithms (e.g., clustering and dimensionality reduction algorithms), and supervised algorithms that are not provably PAC learners (e.g., neural networks trained via SGD). Devising PAC verification protocols for specific algorithms of interest could be a rewarding endeavor.

Part III

Stability

Chapter 6

The Bayesian Stability Zoo

6.1 Introduction

Algorithmic stability is a major theme in learning theory, where seminal results have firmly established its close relationship with generalization. Recent research has further highlighted the intricate interplay between stability and additional properties of interest beyond statistical generalization. These properties encompass privacy (Dwork, McSherry, Nissim, and Smith, 2006), fairness (Hébert-Johnson, Kim, Reingold, and Rothblum, 2018), replicability (Bun, Gaboardi, Hopkins, Impagliazzo, Lei, Pitassi, Sivakumar, and Sorrell, 2023; Impagliazzo, Lei, Pitassi, and Sorrell, 2022), adaptive data analysis (Dwork, Feldman, Hardt, Pitassi, Reingold, and Roth, 2015,), and mistake bounds in online learning (Alon, Livni, Malliaris, and Moran, 2019; Bun, Livni, and Moran, 2020).

This progress has come with a proliferation of formal definitions of stability, including pure and approximate Differential Privacy (Dwork et al., 2006; Dwork, Kenthapadi, McSherry, Mironov, and Naor, 2006), Perfect Generalization (Cummings, Ligett, Nissim, Roth, and Wu, 2016), Global Stability (Bun et al., 2020), KL-Stability (McAllester, 1999), TV-Stability (Kalavasis, Karbasi, Moran, and Velegkas, 2023), f-Divergence Stability (Esposito, Gastpar, and Issa, 2020), Rényi Divergence Stability (Esposito, Gastpar, and Issa, 2020), and Mutual Information Stability (Xu and Raginsky, 2017; Bassily, Moran, Nachum, Shafer, and Yehudayoff, 2018), as well as related combinatorial quantities such as the Littlestone dimension (Littlestone, 1988) and the clique dimension (Alon et al., 2023).

It is natural to wonder to what extent these various and sundry notions of stability actually differ from one another. The type of equivalence we consider between definitions of stability is as follows.

Definition A and Definition B are ***weakly equivalent*** *if for every hypothesis class $\mathcal{H}$ the following holds:*

$$\text{$\mathcal{H}$ has a PAC learning rule that is stable according to Definition A} \iff \text{$\mathcal{H}$ has a PAC learning rule that is stable according to Definition B}$$

This type of equivalence is weak because it does *not* imply that a learning rule satisfying one definition also satisfies the other.

Recent results show that many stability notions appearing in the literature are in fact weakly equivalent. The work of Bun et al. (2023) has shown sample efficient reductions between approximate differential privacy, replicability, and perfect generalization. Combined with the work of Alon, Bun, Livni, Malliaris, and Moran (2022); Impagliazzo et al. (2022); Kalavasis et al. (2023); Malliaris and Moran (2022), a rich web of equivalences is being uncovered between approximate differential privacy and other definitions of algorithmic stability (see Figure 6.1).

In this chapter we extend the study of equivalences between notions of stability, and make it more systematic. Our starting point is the following observation: many of the definitions mentioned above belong to a broad family of definitions of stability, which we informally call *Bayesian definitions of stability.* Definitions in this family roughly take the following form: a learning rule A is considered stable if the quantity

$$d\Big(A(S), \mathcal{P}\Big)$$

is small enough, where:

- d is a measure of dissimilarity between distributions.
- $\mathcal{P}$ is a specific *prior distribution* over hypotheses;
- $A(S)$ is the *posterior distribution*, i.e., the distribution of hypotheses generated by the learning rule A when applied to the input sample S.

Namely, a Bayesian definition of stability is parameterized by a choice of d, a choice of $\mathcal{P}$, and a specification of how small the dissimilarity is required to be.[1]

Remark 6.1.1. *To understand our choice of the name* Bayesian *stability, recall that the terms* prior *and* posterior *come from Bayesian statistics. In Bayesian statistics the analyst has*

[1] An example for an application in the context of generalization is the classic PAC Bayes Theorem. The theorem assures that for every population distribution and any given prior $\mathcal{P}$, the difference between the population error of an algorithm A and the empirical error of A is bounded by $\tilde{O}\left(\frac{\sqrt{\mathrm{KL}(A(S),\mathcal{P})}}{m}\right)$, where m is the size of the input sample S, and the KL divergence is the "measure of dissimilarity" between the prior and the posterior. See e.g. Theorem 6.3.2.

some prior distribution over possible hypothesis before conducting the analysis, and chooses a posterior distribution over hypotheses when the analysis is complete. Bayesian stability is defined in terms of the dissimilarity between these two distributions.

A central insight of this chapter is that there exists a meaningful distinction between two types of Bayesian definitions, based on whether the choice of the prior $\mathcal{P}$ depends on the population distribution $\mathcal{D}$:

- Distribution-*independent* (DI) stability. These are Bayesian definitions of stability in which $\mathcal{P}$ is some fixed prior that depends only on the class $\mathcal{H}$ and the learning rule A, and does not depend on the population distribution $\mathcal{D}$. Namely, they take the form:

  $$\exists \text{ prior } \mathcal{P} \; \forall \text{ population } \mathcal{D} \; \forall m \in \mathbb{N}: \; d(A(S), \mathcal{P}) \text{ is small,}$$

  where $S \sim \mathcal{D}^m$.

- Distribution-*dependent* (DD) stability. Here, the prior may depend also on $\mathcal{D}$, so each population distribution $\mathcal{D}$ might have a different prior. Namely:

  $$\forall \text{ population } \mathcal{D} \; \exists \text{ prior } \mathcal{P}_{\mathcal{D}} \; \forall m \in \mathbb{N}: \; d(A(S), \mathcal{P}_{\mathcal{D}}) \text{ is small.}$$

A substantial body of literature has investigated the interconnections among distribution-dependent definitions. In Theorem 6.1.4, we provide a comprehensive summary of the established equivalences. A natural question arises as to whether a similar web of equivalences exists for distribution-independent definitions. Our principal contribution is to affirm that, indeed, such a network exists. Identifying such equivalences is a step towards creating a comprehensive taxonomy of stability definitions.

Our Contribution

Our first main contribution is an equivalence between distribution-independent definitions of stability.

Theorem (**Informal Version of Theorem 6.2.1**)**.** The following definitions of stability are weakly equivalent:

1. Pure Differential Privacy; (Definition 6.3.5)
2. Distribution-Independent KL-Stability; (Definition 6.3.6)
3. Distribution-Independent One-Way Pure Perfect Generalization; (Definition 6.3.7)
4. Distribution-Independent D_α-Stability for $\alpha \in [1, \infty]$. (Definition 6.3.6)

Where D_α is the Rényi divergence of order α. Furthermore, a hypothesis class $\mathcal{H}$ has a PAC learning rule that is stable according to one of these definitions if and only if $\mathcal{H}$ has finite fractional clique dimension (See Section 6.5).

Remark 6.1.2. *Observe that DI KL-stability is equivalent to DI D_1-stability, and DI one-way pure perfect generalization is equivalent to DI D_∞-stability. Therefore, The above theorem can be viewed as stating a weak equivalence between pure differential privacy and D_α-stability for $\alpha \in [1, \infty]$.*

Remark 6.1.3. *In this chapter we focus purely on the information-theoretic aspects of learning under stability constraints, and therefore we consider learning rules that are mathematical functions, and disregard considerations of computability and computational complexity.*

Table 6.1 summarizes the distribution-independent definitions discussed in Theorem 6.2.1. All the definitions in each row are weakly equivalent.

Name	Dissimilarity	Definition
KL-Stability	$\mathbb{P}_S[\mathsf{KL}(A(S) \parallel \mathcal{P}) \leq o(m)] \geq 1 - o(1)$	6.3.6
D_α-Stability	$\mathbb{P}_S[\mathsf{D}_\alpha(A(S) \parallel \mathcal{P}) \leq o(m)] \geq 1 - o(1)$	6.3.6
Pure Perfect Generalization	$\mathbb{P}_S\Big[\forall \mathcal{O} :\ A(S)(\mathcal{O}) \leq e^{o(m)}\mathcal{P}(\mathcal{O})\Big] \geq 1 - o(1)$	6.3.7

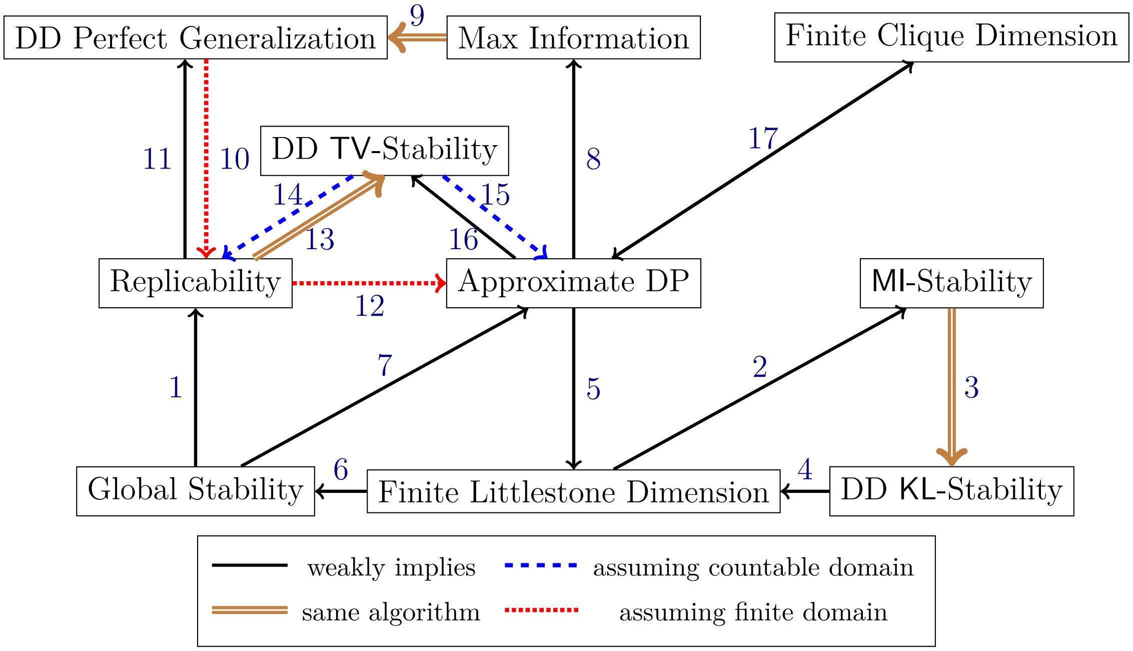
Table 6.1: Distribution-independent Bayesian definitions of stability.

One example for how the equivalence results can help build bridges between different stability notions in the literature is the connection between pure differential privacy and the PAC-Bayes theorem. Both of these are fundamental ideas that have been extensively studied. Theorem 6.2.1 states that a hypothesis class admits a pure differentially private PAC learner if and only if it admits a distribution independent KL-stable PAC learner. This is an interesting and non-trivial connection between two well studied notions. As a concrete example of this connection, recall that thresholds over the real line cannot be learned by a differentially private learner (Alon et al., 2019). Hence, by Theorem 6.2.1, there does not exist a PAC learner for thresholds that is KL-stable. Another example is half-spaces with margins in $\mathbb{R}^d$. Half-spaces with margins are differentially private learnable (Blum, Dwork, McSherry, and Nissim, 2005), therefore there exists a PAC learner for half-spaces with margins that is KL-stable.

Our second main contribution is a boosting result for weak learners that have bounded KL-divergence with respect to a distribution-independent prior. Our result demonstrates that distribution-independent KL-stability is boostable. It is interesting to see that one can simultaneously boost both the stability and the learning parameters of an algorithm.

Theorem (Informal Version of Theorem 6.2.2). Let $\mathcal{H}$ be a hypothesis class. If there exists a weak learner A for $\mathcal{H}$, and there exists a prior distribution $\mathcal{P}$ such that the expectation of $\mathsf{KL}(A(S) \,\|\, \mathcal{P})$ is bounded, then there exists a KL-stable PAC learner that admits a logarithmic divergence bound.

The proof of Theorem 6.2.2 relies on connections between boosting of PAC learners and online learning with expert advice.



1. Lemma 6.6.6
2. Lemma 6.6.13, Pradeep, Nachum, and Gastpar (2022)
3. Lemma 6.6.11
4. Lemma 6.6.8, Livni and Moran (2020)
5. Corollary 2, Alon et al. (2019)
6. Theorem 10, Bun et al. (2020)
7. Theorem 17, Bun et al. (2020)
8. Corollary 3.13, Bun et al. (2023)
9. Lemma 3.14, Bun et al. (2023)
10. Theorem 3.17, Bun et al. (2023)
11. Theorem 3.19, Bun et al. (2023)
12. Theorem 3.1, Bun et al. (2023)
13. Theorem 1, Kalavasis et al. (2023)
14. Theorem 2, Kalavasis et al. (2023)
15. Theorem 4, Kalavasis et al. (2023)
16. Theorem 3, Kalavasis et al. (2023)
17. Theorems 2.2, 2.7 Alon et al. (2023)

Figure 6.1: A summary of equivalences between distribution-dependent definitions of stability (Theorem 6.1.4). A solid black arrow from A to B means that definition A weakly implies definition B. A dashed blue arrow from A to B means that A weakly implies B only if the domain $\mathcal{X}$ is countable. A dotted red arrow from A to B means that A weakly implies B only if the domain $\mathcal{X}$ is finite. A double brown arrow from A to B means that every learning rule that satisfies definition A also satisfies definition B.

Lastly, after conducting an extensive review of the literature, we have compiled a comprehensive network of equivalence results for distribution-dependent definitions of stability. This network is presented in Theorem 6.1.4, Figure 6.1, and Table 6.2.

Theorem 6.1.4 (Distribution-Dependent Equivalences; Alon et al., 2022; Impagliazzo et al., 2022; Malliaris and Moran, 2022; Pradeep et al., 2022; Bun et al., 2023; Kalavasis et al., 2023)**.** *The following definitions of stability are weakly equivalent with respect to an arbitrary hypothesis class $\mathcal{H}$:*

1. *Approximate Differential Privacy;* *(Definition 6.3.5)*
2. *Distribution-Dependent* KL*-Stability;* *(Definition 6.3.6)*
3. *Mutual-Information Stability;* *(Definition 6.3.12)*
4. *Global Stability.* *(Definition 6.3.11)*

If the domain is countable then the following are also weakly equivalent to the above:

5. *Distribution-Dependent* TV*-Stability;* *(Definition 6.3.13)*
6. *Replicability.* *(Definition 6.3.8)*

If the domain is finite then the following are also weakly equivalent to the above:

7. *One-Way Perfect Generalization;* *(Definition 6.3.7)*
8. *Max Information.* *(Definition 6.3.14)*

Furthermore, for any hypothesis class $\mathcal{H}$, the following conditions are equivalent:

- *$\mathcal{H}$ has a PAC learning rule that is stable according to one of the definitions 1 to 6 (and the cardinality of the domain is as described above);*
- *$\mathcal{H}$ has finite Littlestone dimension;* *(Definition 6.6.3)*
- *$\mathcal{H}$ has finite clique dimension.* *(Definition 6.6.5)*

We emphasize that Theorem 6.1.4 is a summary of existing results, and is not a new result. We believe that our compilation serves as a valuable resource, and that stating these results here in a unified framework helps to convey the conceptual message of this chapter. Namely, the fact that a large number of disparate results can neatly be organized based on our notions of distribution-dependent and distribution-independent definitions of stability is a valuable observation that can help researchers make sense of the stability landscape.

Name	Dissimilarity	Definition	References
KL-Stability	$\mathbb{P}_S[\mathsf{KL}(A(S) \parallel \mathcal{P}_{\mathcal{D}}) \leq o(m)] \geq 1 - o(1)$	6.3.6	McAllester (1999)
TV-Stability	$\mathbb{E}_S[\mathsf{TV}(A(S), \mathcal{P}_{\mathcal{D}})] \leq o(1)$	6.3.13	Kalavasis et al. (2023)
MI-Stability	$\mathbb{E}_S[\mathsf{KL}(A(S) \parallel \mathcal{P}_{\mathcal{D}})] \leq o(m)$	6.3.12	Xu and Raginsky (2017) & Bassily et al. (2018)
Perfect Generalization	$\mathbb{P}_S[\forall \mathcal{O} : \ A(S)(\mathcal{O}) \leq e^{\varepsilon}\mathcal{P}_{\mathcal{D}}(\mathcal{O}) + \delta] \geq 1 - o(1)$	6.3.7	Cummings et al. (2016)
Global Stability	$\mathbb{P}_{S,h\sim\mathcal{P}_{\mathcal{D}}}[A(S) = h] \geq \eta$	6.3.11	Bun et al. (2020)
Replicability	$\mathbb{P}_{r\sim\mathcal{R}}\left[\mathbb{P}_{S,h_r\sim\mathcal{P}_{\mathcal{D},r}}[A(S;r) = h_r] \geq \eta\right] \geq \nu$	6.3.10	Bun et al. (2023) & Impagliazzo et al. (2022)

Table 6.2: Distribution-dependent Bayesian definitions of stability.

Related Works

The literature on stability is vast. Stability has been studied in the context of optimization, statistical estimation, regularization (e.g., Tikhonov, 1943 and Phillips, 1962), the bias-variance trade-off, algorithmic stability (e.g., Bousquet and Elisseeff, 2002; see bibliography in Section 13.6 of Shalev-Shwartz and Ben-David, 2014), bagging (Breiman, 1996), online learning and optimization and bandit algorithms (e.g., Hannan, 1958; see bibliography in Section 28.6 of Lattimore and Szepesvári, 2020), and other topics.

There are numerous definitions of stability, including pure and approximate Differential Privacy (Dwork et al., 2006,), Perfect Generalization (Cummings et al., 2016), Global Stability (Bun et al., 2020), KL-Stability (McAllester, 1999), TV-Stability (Kalavasis et al., 2023), f-Divergence Stability (Esposito et al., 2020), Rényi Divergence Stability (Esposito et al., 2020), and Mutual Information Stability (Xu and Raginsky, 2017; Bassily et al., 2018).

Our work is most directly related to the recent publication by Bun et al. (2023). They established connections and separations between replicability, approximate differential privacy, max-information and perfect generalization for a broad class of statistical tasks. The reductions they present are sample-efficient, and nearly all are computationally efficient and apply to a general outcome space. Their results are central to the understanding of equivalences between notions of stability as laid out in the current chapter.

A concurrent work by Kalavasis et al. (2023) showed that TV-stability, replicability and approximate differential privacy are equivalent; this holds for general statistical tasks on countable domains, and for PAC learning on any domain. They also provide a statistical amplification and TV-stability boosting algorithm for PAC learning on countable domains.

Additionally, recent works (Asi, Ullman, and Zakynthinou, 2023; Hopkins, Kamath, Majid, and Narayanan, 2023) have shown an equivalence between differential privacy and robustness for estimation tasks.

Theorem 6.2.2 is a boosting result. Boosting has been a central topic of study in computational learning theory since its inception in the 1990s by Schapire (1990) and Freund (1995). The best-known boosting algorithm is AdaBoost (Freund and Schapire, 1997), which has been extensively studied. Boosting also has rich connections with other topics such as game theory, online learning, and convex optimization (see Schapire and Freund, 2012, Chapter 10 in Shalev-Shwartz and Ben-David, 2014, and Chapter 7 in Mohri, Rostamizadeh, and Talwalkar, 2018).

6.2 Technical Overview

This section presents the complete versions of Theorems 6.1.4 and 6.2.2. We provide a concise overview of the key ideas and techniques employed in the proofs. Please refer to Section 6.3 for a complete overview of preliminaries, including all technical terms and definitions.

Equivalences between DI Bayesian Notions of Stability

The following theorem, which is one of the main results of this chapter, shows the equivalence between different distribution-independent definitions. The content of Theorem 6.2.1 is summarized in Table 6.1.

Theorem 6.2.1 (Distribution-Independent Equivalences). *Let $\mathcal{H}$ be a hypothesis class. The following is equivalent.*

1. *There exists a learning rule that PAC learns $\mathcal{H}$ and satisfied pure differential privacy (Definition 6.3.5).*

2. *$\mathcal{H}$ has finite fractional clique dimension.*

3. *For every $\alpha \in [1, \infty]$, there exists a learning rule that PAC learns $\mathcal{H}$ and satisfied distribution-independent D_α-stability (Definition 6.3.6).*

4. *For every $\alpha \in [1, \infty]$, there exists a distribution-independent D_α-stable PAC learner A for $\mathcal{H}$, that satisfies the following:*

   (i) A is interpolating almost surely. Namely, for every $\mathcal{H}$-realizable distribution $\mathcal{D}$, $\mathbb{P}_{S\sim\mathcal{D}^m}\left[\mathrm{L}_S^{0\text{-}1}(A(S)) = 0\right] = 1$.

   (ii) A admits a divergence bound of $f(m) = O(\log m)$, with confidence $\beta(m) \equiv 0$. I.e., for every $\mathcal{H}$-realizable distribution $\mathcal{D}$, $\mathsf{D}_\alpha(A(S) \,\|\, \mathcal{P}) \leq O(\log m)$ with probability 1, where $S \sim \mathcal{D}^m$ and $\mathcal{P}$ is a prior distribution independent of $\mathcal{D}$.

   (iii) For every $\mathcal{H}$-realizable distribution $\mathcal{D}$, the expected population loss of A with respect to $\mathcal{D}$ satisfies $\mathbb{E}_{S\sim\mathcal{D}^m}\left[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}(A(S))\right] \leq O\left(\sqrt{m^{-1}\log m}\right)$.

In particular, plugging $\alpha = 1$ in Item (ii) implies KL-stability with divergence bound of $f(m) = O(\log m)$ and confidence $\beta(m) \equiv 0$. Plugging $\alpha = \infty$ implies distribution-independent one-way ε-pure perfect generalization, with $\varepsilon(m) \leq O(\log m)$ and confidence $\beta(m) \equiv 0$.

Proof Idea for Theorem 6.2.1

We prove the following chain of implications:

$$\text{Pure DP} \overset{(1)}{\Longrightarrow} \mathsf{D}_\infty\text{-Stability} \overset{(2)}{\Longrightarrow} \mathsf{D}_\alpha\text{-Stability } \forall\alpha \in [1,\infty] \overset{(3)}{\Longrightarrow} \text{Pure DP}.$$

Pure DP $\overset{(1)}{\Longrightarrow}$ D_∞-Stability. The first step towards proving implication (1) is to define a suitable prior distribution $\mathcal{P}$ over hypotheses. The key tool we used in order to define $\mathcal{P}$ is the characterization of pure DP via the fractional clique dimension of Alon et al. (2023). In a nutshell, Alon et al. (2023) proved that (i) a class $\mathcal{H}$ is pure DP learnable if and only if the fractional clique dimension of $\mathcal{H}$ is finite; (ii) the fractional clique dimension is finite if and only if there exists a polynomial $q(m)$ and a distribution over hypothesis $\mathcal{P}_m$, such that for every realizable sample S of size m, we have

$$\mathbb{P}_{h\sim\mathcal{P}_m}\left[\mathrm{L}_S^{0\text{-}1}(h) = 0\right] \geq \frac{1}{q(m)}. \tag{6.1}$$

(For more details please refer to Section 6.5.) Now, the desired prior distribution $\mathcal{P}$ is defined to be a mixture of all the $\mathcal{P}_m$'s.

The next step in the proof is to define a learning rule A: (i) sample hypotheses from the prior $\mathcal{P}$; (ii) return the first hypothesis h that is consistent with the input sample S (i.e. $\mathrm{L}_S^{0\text{-}1}(h) = 0$). A is well-defined since with high probability it will stop and return a hypothesis after $\approx q(m)$ re-samples from $\mathcal{P}$. Since the posterior $A(S)$ is supported on $\{h : \mathrm{L}_S^{0\text{-}1}(h) = 0\}$, a simple calculation which follows from Equation (6.1) shows that for every realizable distribution $\mathcal{D}$, $\mathsf{D}_\infty(A(S) \,\|\, \mathcal{P}) \leq \log(q(m))$ almost surly where $S \sim \mathcal{D}^m$.

D_∞-Stability $\overset{(2)}{\Longrightarrow}$ D_α-Stability $\forall\alpha \in [1,\infty]$. This implication is immediate since the Rényi divergence $\mathsf{D}_\alpha(\mathcal{Q}_1 \,\|\, \mathcal{Q}_2)$ is non-decreasing in α.

D_α-Stability $\forall\alpha \in [1,\infty]$ $\overset{(3)}{\Longrightarrow}$ Pure DP. In fact, it suffices to assume KL-stability. We prove that the promised prior $\mathcal{P}$ satisfies that for every realizable sample S of size m, we have $\mathbb{P}_{h\sim\mathcal{P}}\left[\mathrm{L}_S^{0\text{-}1}(h) = 0\right] \geq \frac{1}{\mathsf{poly}(m)}$, and conclude that $\mathcal{H}$ is pure DP learnable. Given a realizable sample S of size m, we uniformly sample $\approx m\log m$ examples from S and feed the new sample S' to the promised KL-stable learner A. By noting that if $\mathsf{KL}(A(S') \,\|\, \mathcal{P})$ is small, one can lower bound the probability of an event according to $\mathcal{P}$ by its probability according to $A(S')$. The proof then follows by applying a standard concentration argument.

Stability Boosting

We prove a boosting result for weak learners with bounded KL with respect to a distribution-independent prior. We show that every learner with bounded KL that slightly beats random guessing can be amplified to a learner with logarithmic KL and expected loss of $O(\sqrt{m^{-1}\log m})$.

Theorem 6.2.2 (Boosting Weak Learners with Bounded KL)**.** *Let $\mathcal{X}$ be a set, let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class, and let A be a learning rule. Assume there exists $k \in \mathbb{N}$ and $\gamma > 0$ such that*

$$\forall \mathcal{D} \in \mathsf{Realizable}(\mathcal{H}) : \ \mathbb{E}_{S\sim\mathcal{D}^k}\Big[\mathrm{L}^{0\text{-}1}_{\mathcal{D}}(A(S))\Big] \leq \frac{1}{2} - \gamma, \tag{6.2}$$

and there exists $\mathcal{P} \in \Delta\Big(\{0,1\}^{\mathcal{X}}\Big)$ and $b \geq 0$ such that

$$\forall \mathcal{D} \in \mathsf{Realizable}(\mathcal{H}) : \ \mathbb{E}_{S\sim\mathcal{D}^k}[\mathsf{KL}(A(S) \parallel \mathcal{P})] \leq b. \tag{6.3}$$

Then, there exists an interpolating learning rule $A^\star$ that PAC learns $\mathcal{H}$ with logarithmic KL-stability. More explicitly, there exists a prior distribution $\mathcal{P}^\star \in \Delta\Big(\{0,1\}^{\mathcal{X}}\Big)$ and function $b^\star$ and $\varepsilon^\star$ that depend on γ and b such that

$$\forall \mathcal{D} \in \mathsf{Realizable}(\mathcal{H}) \ \forall m \in \mathbb{N} :$$

$$\mathbb{P}_{S\sim\mathcal{D}^m}[\mathsf{KL}(A^\star(S) \parallel \mathcal{P}^\star) \leq b^\star(m) = O(\log(m))] = 1, \tag{6.4}$$

and

$$\mathbb{E}_{S\sim\mathcal{D}^m}\Big[\mathrm{L}^{0\text{-}1}_{\mathcal{D}}(A^\star(S))\Big] \leq \varepsilon^\star(m) = O\left(\sqrt{\frac{\log(m)}{m}}\right). \tag{6.5}$$

Proof Idea for Theorem 6.2.2

The strong learning rule $A^\star$ is obtained by simulating the weak learner A on $O(\log m/\gamma^2)$ samples of constant size k (which are carefully sampled from the original input sample S). Then, $A^\star$ returns an aggregated hypothesis – the majority vote of the outputs of A. As it turns out, $A^\star$ satisfies logarithmic KL-stability with respect to the prior $\mathcal{P}^\star$ that is a mixture of majority votes of the original prior $\mathcal{P}$. The analysis involves a reduction to regret analysis of online learning using expert advice, and also uses properties of the KL-divergence.

6.3 Preliminaries

Divergences

The Rényi α-divergence is a measure of dissimilarity between distributions that generalizes many common dissimilarity measures, including the Bhattacharyya coefficient ($\alpha = 1/2$), the Kullback–Leibler divergence ($\alpha = 1$), the log of the expected ratio ($\alpha = 2$), and the log of the maximum ratio ($\alpha = \infty$).

Definition 6.3.1 (Rényi divergence; Rényi, 1961; van Erven and Harremoës, 2014)**.** *Let $\alpha \in (1,\infty)$. The Rényi divergence of order α of the distribution $\mathcal{P}$ from the distribution $\mathcal{Q}$ is*

$$\mathsf{D}_\alpha(\mathcal{P} \parallel \mathcal{Q}) = \frac{1}{\alpha - 1} \log\left(\mathbb{E}_{x\sim\mathcal{P}}\left[\left(\frac{\mathcal{P}(x)}{\mathcal{Q}(x)}\right)^{\alpha-1}\right]\right).$$

For $\alpha = 1$ and $\alpha = \infty$ the Rényi divergence is extended by taking a limit. In particular, the limit $\alpha \to 1$ gives the Kullback–Leibler divergence,

$$\mathsf{D}_1(\mathcal{P} \parallel \mathcal{Q}) = \mathbb{E}_{x\sim\mathcal{P}}\left[\log \frac{\mathcal{P}(x)}{\mathcal{Q}(x)}\right] = \mathsf{KL}(\mathcal{P} \parallel \mathcal{Q}),$$

and

$$\mathsf{D}_\infty(\mathcal{P} \parallel \mathcal{Q}) = \log\left(\operatorname*{ess\,sup}_{\mathcal{P}} \frac{\mathcal{P}(x)}{\mathcal{Q}(x)}\right),$$

with the conventions that $0/0 = 0$ and $x/0 = \infty$ for $x > 0$.

Learning Theory

We use standard notation from statistical learning (e.g., Shalev-Shwartz and Ben-David, 2014). Given a hypothesis $h : \mathcal{X} \to \{0,1\}$, the *empirical loss* of h with respect to a sample $S = \{(x_1,y_1),\ldots,(x_m,y_m)\}$ is defined as $\mathrm{L}_S^{0\text{-}1}(h) = \frac{1}{m}\sum_{i=1}^m \mathbb{1}[h(x_i) \neq y_i]$. A learning rule A is *interpolating* if for every input sample S, $\mathbb{P}_{h\sim A(S)}\left[\mathrm{L}_S^{0\text{-}1}(h) = 0\right] = 1$. The *population loss* of h with respect to a population distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ is defined as $\mathrm{L}_\mathcal{D}^{0\text{-}1}(h) = \mathbb{P}_{(x,y)\sim\mathcal{D}}[h(x) \neq y]$. A population $\mathcal{D}$ over labeled examples is *realizable* with respect to a class $\mathcal{H}$ if $\inf_{h\in\mathcal{H}} \mathrm{L}_\mathcal{D}^{0\text{-}1}(h) = 0$. We denote the set of all realizable population distributions of a class $\mathcal{H}$ by $\mathsf{Realizable}(\mathcal{H})$. Given a learning rule A and an input sample S of size m, the *population loss* of $A(S)$ with respect to a population $\mathcal{D}$ is defined as $\mathbb{E}_{h\sim A(S)}\left[\mathrm{L}_\mathcal{D}^{0\text{-}1}(h)\right]$.

A hypothesis class $\mathcal{H}$ is *Probably Approximately Correct (PAC) learnable* if there exists a learning rule A such that for all $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$ and for all $m \in \mathbb{N}$, we have $\mathbb{E}_{S\sim\mathcal{D}^m}\left[\mathrm{L}_\mathcal{D}^{0\text{-}1}(A(S))\right] \leq \varepsilon(m)$, where $\lim_{m\to\infty} \varepsilon(m) = 0$.

Theorem 6.3.2 (PAC-Bayes Bound; McAllester, 1999; Langford, Seeger, and Megiddo, 2001; McAllester, 2003; Theorem 31.1 in Shalev-Shwartz and Ben-David, 2014)**.** *Let $\mathcal{X}$ be a set, let $\mathcal{H} \subseteq \{0,1\}^\mathcal{X}$, and let $\mathcal{D} \in \Delta(\mathcal{X} \times \{0,1\})$. For any $\beta \in (0,1)$ and for any $\mathcal{P} \in \Delta(\mathcal{H})$,*

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[\forall \mathcal{Q} \in \Delta(\mathcal{H}) : \quad \mathrm{L}_\mathcal{D}^{0\text{-}1}(\mathcal{Q}) \leq \mathrm{L}_S^{0\text{-}1}(\mathcal{Q}) + \sqrt{\frac{\mathsf{KL}(\mathcal{Q} \parallel \mathcal{P}) + \ln(m/\beta)}{2(m-1)}}\right] \geq 1 - \beta.$$

Definitions of Stability

Throughout the following section, let $\mathcal{X}$ be a set called the *domain*, let $\mathcal{H} \subseteq \{0,1\}^\mathcal{X}$ be a hypothesis class, and let $m \in \mathbb{N}$ be a sample size. A *randomized learning rule*, or a *learning* *rule* for short, is a function $A: (\mathcal{X} \times \{0,1\})^* \to \Delta\Big(\{0,1\}^{\mathcal{X}}\Big)$ that takes a training sample and outputs a distribution over hypotheses. A *population distribution* is a distribution $\mathcal{D} \in \Delta(\mathcal{X} \times \{0,1\})$ over labeled domain elements, and a *prior distribution* is a distribution $\mathcal{P} \in \Delta\Big(\{0,1\}^{\mathcal{X}}\Big)$ over hypotheses.

Differential Privacy

Differential privacy is a property of an algorithm that guarantees that the output will not reveal any meaningful amount of information about individual people that contributed data to the input (training data) used by the algorithm. See Dwork and Roth (2014) for an introduction.

Definition 6.3.3. *Let $\varepsilon, \delta \in \mathbb{R}_{\geq 0}$, and let $\mathcal{P}$ and $\mathcal{Q}$ be two probability measures over a measurable space $(\Omega, \mathcal{F})$. We say that $\mathcal{P}$ and $\mathcal{Q}$ are $\underline{(\varepsilon,\delta)\text{-indistinguishable}}$ and write $\mathcal{P} \approx_{\varepsilon,\delta} \mathcal{Q}$, if for every event $\mathcal{O} \in \mathcal{F}$, $\mathcal{P}(\mathcal{O}) \leq e^{\varepsilon} \cdot \mathcal{Q}(\mathcal{O}) + \delta$ and $\mathcal{Q}(\mathcal{O}) \leq e^{\varepsilon} \cdot \mathcal{P}(\mathcal{O}) + \delta$.*

Definition 6.3.4 (Differential Privacy; Dwork and Roth (2014))**.** *Let $\varepsilon, \delta \in \mathbb{R}_{\geq 0}$. A learning rule A is $\underline{(\varepsilon,\delta)\text{-differentially private}}$ if for every pair of training samples $S, S' \in (\mathcal{X} \times \{0,1\})^m$ that differ on a single example, $A(S)$ and $A(S')$ are (ε,δ)-indistinguishable.*

Typically, ε is chosen to be a small constant (e.g., $\varepsilon \leq 0.1$) and δ is negligible (i.e., $\delta(m) \leq m^{-\omega(1)}$). When $\delta = 0$ we say that A satisfies *pure* differentially privacy.

Definition 6.3.5 (Private PAC Learning)**.** *$\mathcal{H}$ is <u>privately learnable</u> or <u>DP learnable</u> if it is PAC learnable by a learning rule A which is $(\varepsilon(m), \delta(m))$-differentially-private, where $\varepsilon(m) \leq 1$ and $\delta(m) = m^{-\omega(1)}$. A is <u>pure DP learnable</u> if the same holds with $\delta(m) = 0$.*

D_α-Stability and KL-Stability

Definition 6.3.6 (D_α-Stability)**.** *Let $\alpha \in [1, \infty]$. Let A be a learning rule, and let $f: \mathbb{N} \to \mathbb{R}$ and $\beta: \mathbb{N} \to [0,1]$ satisfy $f(m) = o(m)$ and $\beta(m) = o(1)$.*

1. *A is $\underline{\text{distribution-independent } \mathsf{D}_\alpha\text{-stable}}$ if*

$$\exists \text{ prior } \mathcal{P}\, \forall \text{ population } \mathcal{D}\, \forall m \in \mathbb{N}: \ \mathbb{P}_{S \sim \mathcal{D}^m}[\mathsf{D}_\alpha(A(S) \,\|\, \mathcal{P}) \leq f(m)] \geq 1 - \beta(m).$$

2. *A is $\underline{\text{distribution-dependent } \mathsf{D}_\alpha\text{-stable}}$ if*

$$\forall \text{ population } \mathcal{D}\, \exists \text{ prior } \mathcal{P}_{\mathcal{D}}\, \forall m \in \mathbb{N}: \ \mathbb{P}_{S \sim \mathcal{D}^m}[\mathsf{D}_\alpha(A(S) \,\|\, \mathcal{P}_{\mathcal{D}}) \leq f(m)] \geq 1 - \beta(m).$$

The function f is called the divergence bound and β is called the confidence. The special case of $\alpha = 1$ is referred to as $\underline{\mathsf{KL}\text{-stability}}$ (McAllester, 1999).

Perfect Generalization

Definition 6.3.7 (One-Way Perfect Generalization)**.** *Let A be a learning rule, and let $\beta : \mathbb{N} \to [0,1]$ satisfy $\beta(m) = o(1)$.*

1. *Let $\varepsilon : \mathbb{N} \to \mathbb{R}$ satisfy $\varepsilon(m) = o(m)$. A is $\underline{\varepsilon\text{-pure perfectly generalizing}}$ with confidence β if*

$$\exists \text{ prior } \mathcal{P}\, \forall \text{ population } \mathcal{D}\, \forall m \in \mathbb{N}: \ \mathbb{P}_{S\sim\mathcal{D}^m}\Big[\forall \mathcal{O}: \ A(S)(\mathcal{O}) \le e^{\varepsilon(m)}\mathcal{P}(\mathcal{O})\Big] \ge 1-\beta(m).$$

2. *Let $\varepsilon, \delta \in \mathbb{R}_{\ge 0}$. A is $\underline{(\varepsilon,\delta)\text{-approximately perfectly generalizing}}$ (Cummings et al., 2016) with confidence β if*

$$\begin{aligned}&\forall \text{ population } \mathcal{D}\, \exists \text{ prior } \mathcal{P}_{\mathcal{D}}\, \forall m \in \mathbb{N}: \\ &\qquad \mathbb{P}_{S\sim\mathcal{D}^m}[\forall \mathcal{O}: \ A(S)(\mathcal{O}) \le e^{\varepsilon}\mathcal{P}_{\mathcal{D}}(\mathcal{O}) + \delta] \ge 1-\beta(m).\end{aligned}$$

Replicability

Definition 6.3.8 (Replicability; Bun et al., 2023; Impagliazzo et al., 2022)**.** *Let $\rho \in \mathbb{R}_{>0}$ and let $\mathcal{R}$ be a distribution over random strings. A learning rule A is $\underline{\rho\text{-replicable}}$ if*

$$\forall \text{ population } \mathcal{D}, \forall m: \ \mathbb{P}_{\substack{S_1,S_2\sim\mathcal{D}^m \\ r\sim\mathcal{R}}}[A(S_1;r) = A(S_2;r)] \ge \rho,$$

where r represents the random coins of A.

Remark 6.3.9. *Note that both in Bun et al. (2023) and in Impagliazzo et al. (2022) the definition of ρ-replicability is slightly different. In their definition, they treat the parameter ρ as the failure probability, i.e., A is a ρ-replicable learning rule by their definition if the probability that $A(S_1;r) = A(S_2;r)$ is at least $1-\rho$.*

There exists an alternative 2-parameter definition of replicability introduced in Impagliazzo et al. (2022).

Definition 6.3.10 ((η,ν)-Replicability; Bun et al., 2023; Impagliazzo et al., 2022)**.** *Let $\eta, \nu \in \mathbb{R}_{>0}$ and let $\mathcal{R}$ be a distribution over random strings. Coin tosses r are $\underline{\eta\text{-good}}$ for a learning rule A with respect to a population distribution $\mathcal{D}$ if there exists a canonical output h_r such that for every m, $\mathbb{P}_{S\sim\mathcal{D}^m}[A(S;r) = h_r] \ge \eta$. A learning rule A is $\underline{(\eta,\nu)\text{-replicable}}$ if*

$$\forall \text{ population } \mathcal{D}: \ \mathbb{P}_{r\sim\mathcal{R}}[r \text{ is } \eta\text{-good}] \ge \nu.$$

Global Stability

Definition 6.3.11 (Global Stability; Bun et al., 2020)**.** *Let $\eta > 0$ be a global stability parameter. A learning rule A is $\underline{(m,\eta)\text{-globally stable}}$ with respect to a population distribution $\mathcal{D}$ if there exists a canonical output h such that $\mathbb{P}[A(S) = h] \ge \eta$, where the probability is over $S \sim \mathcal{D}^m$ as well as the internal randomness of A.*

MI-Stability

Definition 6.3.12 (Mutual Information Stability; Xu and Raginsky, 2017; Bassily et al., 2018)**.** *A learning rule A is* MI*-stable if there exists $f : \mathbb{N} \to \mathbb{N}$ with $f = o(m)$ such that*

$$\forall \textit{ population } \mathcal{D} \, \forall m \in \mathbb{N} : I(A(S), S) \leq f(m),$$

where $S \sim \mathcal{D}^m$.

TV-Stability

Definition 6.3.13 (TV-Stability; Appendix A.3.1 in Kalavasis et al., 2023)**.** *Let A be a learning rule, and let $f : \mathbb{N} \to \mathbb{N}$ satisfy $f(m) = o(1)$.*

1. *A is distribution-independent* TV*-stable if*

$$\exists \textit{ prior } \mathcal{P} \, \forall \textit{ population } \mathcal{D} \, \forall m \in \mathbb{N} : \; \mathbb{E}_{S \sim \mathcal{D}^m}[\mathsf{TV}(A(S), \mathcal{P})] \leq f(m).$$

2. *A is distribution-dependent* TV*-stable if*

$$\forall \textit{ population } \mathcal{D} \, \exists \textit{ prior } \mathcal{P}_{\mathcal{D}} \, \forall m \in \mathbb{N} : \; \mathbb{E}_{S \sim \mathcal{D}^m}[\mathsf{TV}(A(S), \mathcal{P}_{\mathcal{D}})] \leq f(m).$$

Max Information

Definition 6.3.14. *Let A be a learning rule, and let $\varepsilon, \delta \in \mathbb{R}_{\geq 0}$. A has (ε, δ)-max-information with respect to product distributions if for every event $\mathcal{O}$ we have*

$$\mathbb{P}[(A(S), S) \in \mathcal{O}] \leq e^{\varepsilon} \mathbb{P}[(A(S), S') \in \mathcal{O}] + \delta$$

where are S, S' are independent samples drown i.i.d from a population distribution $\mathcal{D}$.

6.4 Proof of Theorem 6.2.2 (Stability Boosting)

Information Theoretic Preliminaries

Lemma 6.4.1 (Monotonicity of Rényi divergence; Theorem 3 in van Erven and Harremoës, 2014)**.** *Let $0 \leq \alpha < \beta \leq \infty$. Then $\mathsf{D}_\alpha(\mathcal{P} \,\|\, \mathcal{Q}) \leq \mathsf{D}_\beta(\mathcal{P} \,\|\, \mathcal{Q})$. Furthermore, the inequality is an equality if and only if $\mathcal{P}$ equals the conditional $\mathcal{Q}(\cdot \mid A)$ for some event A.*

Lemma 6.4.2 (Data Processing Inequality; Theorem 9 and Eq. 13 in van Erven and Harremoës, 2014)**.** *Let $\alpha \in [0, \infty]$. Let X and Y be random variables, and let $F_{Y|X}$ be the law of Y given X. Let $\mathcal{P}_Y, \mathcal{Q}_Y$ be the distributions of Y when X is sampled from $\mathcal{P}_X, \mathcal{Q}_X$, respectively. Then*

$$\mathsf{D}_\alpha(\mathcal{P}_Y \,\|\, \mathcal{Q}_Y) \leq \mathsf{D}_\alpha(\mathcal{P}_X \,\|\, \mathcal{Q}_X).$$

One interpretation of this is that processing an observation makes it more difficult to determine whether it came from $\mathcal{P}_X$ or $\mathcal{Q}_X$.

Definition 6.4.3 (Conditional KL-divergence; Definition 2.12 in Polyanskiy and Wu (2023+))**.** *Given joint distributions $\mathcal{P}(x,y), \mathcal{Q}(x,y)$, the KL-divergence of the marginals $\mathcal{P}(y|x), \mathcal{Q}(y|x)$ is*

$$\mathsf{KL}(\mathcal{P}(y|x) \parallel \mathcal{Q}(y|x)) = \sum_x \mathcal{P}(x) \sum_y \mathcal{P}(y|x) \log \frac{\mathcal{P}(y|x)}{\mathcal{Q}(y|x)}.$$

Lemma 6.4.4 (Chain Rule for KL-divergence; Theorem 2.13 in Polyanskiy and Wu, 2023+)**.** *Let $\mathcal{P}(x,y), \mathcal{Q}(x,y)$ be joint distributions. Then,*

$$\mathsf{KL}(\mathcal{P}(x,y) \parallel \mathcal{Q}(x,y)) = \mathsf{KL}(\mathcal{P}(x) \parallel \mathcal{Q}(x)) + \mathsf{KL}(\mathcal{P}(y|x) \parallel \mathcal{Q}(y|x)).$$

Lemma 6.4.5 (Conditioning increases KL-divergence; Theorem 2.14(e) in Polyanskiy and Wu, 2023+)**.** *For a distribution $\mathcal{P}_X$ and conditional distributions $\mathcal{P}_{Y|X}, \mathcal{Q}_{Y|X}$, let $\mathcal{P}_Y = \mathcal{P}_{Y|X} \circ \mathcal{P}_X$ and $\mathcal{Q}_Y = \mathcal{Q}_{Y|X} \circ \mathcal{P}_X$, where '$\circ$' denotes composition (see Section 2.4 in Polyanskiy and Wu, 2023+) Then*

$$\mathsf{KL}(\mathcal{P}_Y \parallel \mathcal{Q}_Y) \leq \mathsf{KL}\left(\mathcal{P}_{Y|X} \parallel \mathcal{Q}_{Y|X} \,\middle|\, \mathcal{P}_X\right),$$

with equality if and only if $\mathsf{KL}\left(\mathcal{P}_{X|Y} \parallel \mathcal{Q}_{X|Y} \,\middle|\, P_Y\right) = 0.$

Online Learning Preliminaries

Following is some basic background on the topic of online learning with expert advice. This will be useful in the proof of Theorem 6.2.2.

Let $Z = \{z_1, \ldots, z_m\}$ be a set of experts and I be a set of instances. For any instance $i \in I$ and expert $z \in Z$, following the advice of expert z on instance i provides utility $u(z,i) \in \{0,1\}$.

The online learning setting is a perfect-information, zero-sum game between two players, a *learner* and an *adversary*. In each round $t = 1, \ldots, T$:

1. The learner chooses a distribution $w_t \in \Delta(Z)$ over the set of experts.

2. The adversary chooses an instance $i_t \in I$.

3. The learner gains utility $u_t = \mathbb{E}_{z \sim w_t}[u(z, i_t)]$.

The *total utility* of a learner strategy $\mathcal{L}$ for the sequence of instances chosen by the adversary is

$$U(\mathcal{L}, T) = \sum_{t=1}^{T} u_t.$$

The *regret* of the learner is the difference between the utility of the best expert and the learner's utility. Namely, for each $z \in Z$, let

$$U(z, T) = \sum_{t=1}^{T} u(z, i_t)$$

be the utility the learner would have gained had they chosen $w_t(z) = \mathbb{1}(z = z_j)$ for all $t \in [T]$. Then the regret is

$$\mathsf{Regret}(\mathcal{L}, T) = \max_{z \in Z} U(z, T) - U(\mathcal{L}, T).$$

There are several well-studied algorithms for online learning using expert advice that guarantee regret sublinear in T for every possible sequence of T instances. A classic example is the *Multiplicative Weights* algorithm (e.g., Section 21.2 in Shalev-Shwartz and Ben-David (2014)), which enjoys the following guarantee.

Theorem 6.4.6 (Online Regret Bound)**.** *In the setting of online learning with expert advice, there exists a learner strategy $\mathcal{L}$ such that for any sequence of T instances selected by the adversary,*

$$\mathsf{Regret}(\mathcal{L}, T) \leq \sqrt{2T \log(m)},$$

where m is the number of experts.

Proof

Theorem (Theorem 6.2.2, Restatement)**.** Let $\mathcal{X}$ be a set, let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class, and let A be a learning rule. Assume there exists $k \in \mathbb{N}$ and $\gamma > 0$ such that

$$\forall \mathcal{D} \in \mathsf{Realizable}(\mathcal{H}): \ \mathbb{E}_{S \sim \mathcal{D}^k}\left[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}(A(S))\right] \leq \frac{1}{2} - \gamma, \tag{6.6}$$

and there exists $\mathcal{P} \in \Delta\Big(\{0,1\}^{\mathcal{X}}\Big)$ and $b \geq 0$ such that

$$\forall \mathcal{D} \in \mathsf{Realizable}(\mathcal{H}): \ \mathbb{E}_{S \sim \mathcal{D}^k}[\mathsf{KL}(A(S) \,\|\, \mathcal{P})] \leq b. \tag{6.7}$$

Then, there exists an interpolating learning rule $A^\star$ that PAC learns $\mathcal{H}$ with logarithmic KL-stability. More explicitly, there exists a prior distribution $\mathcal{P}^\star \in \Delta\Big(\{0,1\}^{\mathcal{X}}\Big)$ and function $b^\star$ and $\varepsilon^\star$ that depend on γ and b such that

$$\forall \mathcal{D} \in \mathsf{Realizable}(\mathcal{H}) \ \forall m \in \mathbb{N}:$$

$$\mathbb{P}_{S \sim \mathcal{D}^m}[\mathsf{KL}(A^\star(S) \,\|\, \mathcal{P}^\star) \leq b^\star(m) = O(\log(m))] = 1, \tag{6.8}$$

and

$$\mathbb{E}_{S \sim \mathcal{D}^m}\left[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}(A^\star(S))\right] \leq \varepsilon^\star(m) = O\left(\sqrt{\frac{\log(m)}{m}}\right). \tag{6.9}$$

Assumptions:

- $\gamma, b > 0$; $m, k \in \mathbb{N}$.
- $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$ is an $\mathcal{H}$-realizable sample.
- $\mathcal{O}_S$ is the online learning algorithm of Section 6.4, using expert set S.
- $T = \lceil 8 \log(m)/\gamma^2 \rceil + 1$.
- A satisfies Eqs. (6.2) and (6.3) (with respect to k, b, γ).

```
$A^\star(S)$:
  for  $t = 1, \dots, T$:
    $w_t \leftarrow$ expert distribution chosen by $\mathcal{O}_S$ for round $t$
    do:
      sample $S_t \leftarrow (w_t)^k$
    while $\mathsf{KL}(A(S_t) \,\|\, \mathcal{P}) \geq 2b/\gamma$                ▷ See Remark 6.4.7
    $f_t \leftarrow A(S_t)$
    $\mathcal{O}_S$ receives instance $f_t$ and gains utility $\mathbb{E}_{(x,y)\sim w_t}[\mathbb{1}(f_t(x) \neq y)]$
  return $\mathsf{Maj}(f_1, \dots, f_T)$
```

Algorithm 6.1: The stability-boosted learning rule $A^\star$, which uses A as a subroutine.

Proof of Theorem 6.2.2. Let $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$ and $m \in \mathbb{N}$. Learning rule $A^\star$ operates as follows. Given a sample $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$, $A^\star$ simulates an online learning game, in which S is the set of 'experts', $\mathcal{F} = \{0,1\}^{\mathcal{X}}$ is the set of 'instances', and the learner's utility for playing expert (x, y) on instance $f \in \mathcal{F}$ is $\mathbb{1}(f(x) \neq y)$. Namely, in this game the learner is attempting to select an (x, y) pair that disagrees with the instance f.

In this simulation, the learner executes an instance of the online learning algorithm of Section 6.4 with expert set S. Denote this instance $\mathcal{O}_S$.

The adversary's strategy is as follows. Recall that at each round t, $\mathcal{O}_S$ chooses a distribution w_t over S. Note that if S is realizable then so is w_t. At each round t, the adversary selects an instance $f \in \mathcal{F}$ by executing A on a training set sampled from w_t, as in Algorithm 6.1.

We prove the following:

1. $A^\star$ interpolates, namely $\mathbb{P}\Big[\mathrm{L}_S^{0\text{-}1}(A^\star(S)) = 0\Big] = 1$.
2. $A^\star$ has logarithmic KL-stability, as in Eq. (6.4).
3. $A^\star$ PAC learns $\mathcal{H}$ as in Eq. (6.5).

For Item 1, assume for contradiction that $A^\star$ does not interpolate. Seeing as $A^\star$ outputs $\mathsf{Maj}(f_1, \dots, f_T)$, there exists an index $i \in [m]$ such that

$$\frac{T}{2} \leq \sum_{t=1}^{T} \mathbb{1}(f_t(x_i) \neq y_i) = U(i, T), \tag{6.10}$$

where $U(i, T)$ is the utility of always playing expert i throughout the game.

Let $\mathcal{E}_t$ denote the event that S_t was resampled (i.e., there were multiple iterations of the do-while loop in round t). Eq. (6.3) and Markov's inequality imply

$$\mathbb{P}[\mathcal{E}_t] = \mathbb{P}[\mathsf{KL}(A(S_t) \,\|\, \mathcal{P}) \geq 2b/\gamma] \leq \gamma/2. \tag{6.11}$$

The utility of $\mathcal{O}_S$ at time t is

$$\begin{aligned} u_t^{\mathcal{O}_S} &= \mathop{\mathbb{E}}_{\substack{S_t \sim (w_t)^k \\ f_t \sim A(S_t) \\ (x,y) \sim w_t}} [\mathbb{1}(f_t(x) \neq y)] \\ &\leq \mathop{\mathbb{E}}_{S_t \sim (w_t)^k} \Big[\mathrm{L}_{w_t}^{0\text{-}1}(A(S_t)) \mid \neg \mathcal{E}_t\Big] + \mathbb{P}[\mathcal{E}_t] \leq \left(\frac{1}{2} - \gamma\right) + \frac{\gamma}{2}, \end{aligned}$$

where the last inequality follows from Eqs. (6.2) and (6.11). Hence, the utility of $\mathcal{O}_S$ throughout the game is

$$U(\mathcal{O}_S, T) = \sum_{t=1}^{T} u_t^{\mathcal{O}_S} \leq \left(\frac{1}{2} - \frac{\gamma}{2}\right) \cdot T. \tag{6.12}$$

Combining Eqs. (6.10) and (6.12) and Theorem 6.4.6 yields

$$\frac{\gamma}{2} \cdot T \leq U(i, T) - U(\mathcal{O}_S, T) \leq \mathsf{Regret}(\mathcal{O}_S, T) \leq \sqrt{2T \log(m)},$$

which is a contradiction for our choice of T. This establishes Item 1.

For Item 2, for every $\ell \in \mathbb{N}$ let $\mathcal{P}_\ell^\star \in \Delta\Big(\{0,1\}^{\mathcal{X}}\Big)$ be the distribution of $\mathsf{Maj}(g_1, \dots, g_\ell)$, where $(g_1, \dots, g_\ell) \sim \mathcal{P}^\ell$. Let $\mathcal{P}^\star = \frac{1}{z} \sum_{\ell=1}^{\infty} \mathcal{P}_\ell^\star / \ell^2$ where $z = \sum_{\ell=1}^{\infty} 1/\ell^2 = \pi^2/6$ is a normalization factor.

For any $S \in (\mathcal{X} \times \{0,1\})^m$,

$$\begin{aligned} \mathsf{KL}(A^\star(S) \,\|\, \mathcal{P}_T^\star) &= \mathsf{KL}(\mathsf{Maj}(f_1, \dots, f_T) \,\|\, \mathsf{Maj}(g_1, \dots, g_T)) \\ &\leq \mathsf{KL}((f_1, \dots, f_T) \,\|\, (g_1, \dots, g_T)) && \text{(By Lemma 6.4.2)} \\ &= \sum_{t=1}^{T} \mathsf{KL}((f_t | f_{<t}) \,\|\, (g_t | g_{<t})) && \text{(By Lemma 6.4.4)} \\ &= \sum_{t=1}^{T} \mathsf{KL}((f_t | f_{<t}) \,\|\, g_t). && \text{(g_i's are independent)} \\ &= \sum_{t=1}^{T} \mathsf{KL}(A(S_t) \,\|\, \mathcal{P}) \leq T \cdot 2b/\gamma = O(\log(m)), \end{aligned} \tag{6.13}$$

where the last inequality is due to the do-while loop in Algorithm 6.1. For any $S \in (\mathcal{X} \times \{0,1\})^m$,

$$\begin{aligned}
\mathsf{KL}(A^\star(S) \,\|\, \mathcal{P}^\star) &= \mathbb{E}_{h \sim P_{A^\star(S)}}\left[\log\left(\frac{P_{A^\star(S)}(h)}{\mathcal{P}^\star(h)}\right)\right] \\
&\leq \mathbb{E}_{h \sim P_{A^\star(S)}}\left[\log\left(\frac{P_{A^\star(S)}(h)}{\mathcal{P}^\star_T(h)/(zT^2)}\right)\right] \\
&= \mathsf{KL}(A^\star(S) \,\|\, \mathcal{P}^\star_T) + O(\log(T)) = O(\log(m)). \qquad \text{(By Eq. (6.13))}
\end{aligned}$$

This establishes Item 2.

Item 3 follows by plugging $\beta = \frac{1}{m}$ and Items 1 and 2 in the PAC-Bayes theorem (Theorem 6.3.2), yielding

$$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^m}\left[\mathrm{L}^{0\text{-}1}_{\mathcal{D}}(A^\star(S)) \leq O\left(\sqrt{\frac{\log(m)}{m}}\right)\right] \geq 1 - \frac{1}{m}.$$

This implies Item 3 because the 0-1 loss is at most 1. □

Remark 6.4.7. *Our definition of the learning rule $A^\star$ depends on A and $\mathcal{P}$. The mapping $S_t \mapsto \mathsf{KL}(A(S_t) \,\|\, \mathcal{P})$ is well-defined, so $A^\star$ is a well-defined learning rule.*[2]

6.5 Proof of Theorem 6.2.1 (DI Equivalences)

In this section, we prove Theorem 6.2.1.

Theorem (Theorem 6.2.1, Restatement)**.** Let $\mathcal{H}$ be a hypothesis class. The following is equivalent.

1. There exists a learning rule that PAC learns $\mathcal{H}$ and satisfied pure differential privacy (Definition 6.3.5).

2. $\mathcal{H}$ has finite fractional clique dimension.

[2] We remark that if A is a randomized Turing machine, then $\mathsf{KL}(A(S_t) \,\|\, \mathcal{P})$ can be estimated to arbitrary precision by a Turing machine with oracle access to the function $\mathcal{P}$. Namely, consider a Turing machine that can query an oracle for the value of $\mathcal{P}(h)$ up to precision 2^{-q} for any h and $q \in \mathbb{N}$ of its choosing. To see that such a machine can estimate $\mathsf{KL}(A(S_t) \,\|\, \mathcal{P})$, observe that if A uses some finite number of random coins, then $A(S_t)$ has a finite support, and so computing $\mathsf{KL}(A(S_t) \,\|\, \mathcal{P})$ involves querying $\mathcal{P}$ at a finite number of locations. Moreover, if A uses a number R of random coins, which is itself a random variable that may be unbounded but satisfies $\mathbb{E}[R] < \infty$, then by Markov's inequality there exists an explicit algorithm A' that uses at most $\mathbb{E}[R]/\alpha$ random coins, such that $\mathsf{TV}(A(S_t), A'(S_t)) < \alpha$. Hence, $\mathsf{KL}(A'(S_t) \,\|\, \mathcal{P})$ can be estimated to arbitrary precision as before. Taking small enough values of α yields a modified version of $A^\star$ that can be shown to satisfy the requirements of Theorem 6.2.2.

3. For every $\alpha \in [1, \infty]$, there exists a learning rule that PAC learns $\mathcal{H}$ and satisfied distribution-independent D_α-stability (Definition 6.3.6).

4. For every $\alpha \in [1, \infty]$, there exists a distribution-independent D_α-stable PAC learner A for $\mathcal{H}$, that satisfies the following:

   (i) A is interpolating almost surely. Namely, for every $\mathcal{H}$-realizable distribution $\mathcal{D}$, $\mathbb{P}_{S\sim\mathcal{D}^m}\left[\mathrm{L}_S^{0\text{-}1}(A(S)) = 0\right] = 1$.

   (ii) A admits a divergence bound of $f(m) = O(\log m)$, with confidence $\beta(m) \equiv 0$. I.e., for every $\mathcal{H}$-realizable distribution $\mathcal{D}$, $\mathsf{D}_\alpha(A(S) \parallel \mathcal{P}) \leq O(\log m)$ with probability 1, where $S \sim \mathcal{D}^m$ and $\mathcal{P}$ is a prior distribution independent of $\mathcal{D}$.

   (iii) For every $\mathcal{H}$-realizable distribution $\mathcal{D}$, the expected population loss of A with respect to $\mathcal{D}$ satisfies $\mathbb{E}_{S\sim\mathcal{D}^m}\left[\mathrm{L}_{\mathcal{D}}^{0\text{-}1}(A(S))\right] \leq O\left(\sqrt{m^{-1}\log m}\right)$.

In particular, plugging $\alpha = 1$ in Item (ii) implies KL-stability with divergence bound of $f(m) = O(\log m)$ and confidence $\beta(m) \equiv 0$. Plugging $\alpha = \infty$ implies distribution-independent one-way ε-pure perfect generalization, with $\varepsilon(m) \leq O(\log m)$ and confidence $\beta(m) \equiv 0$.

The next subsections contain Theorem 6.5.1, which is a useful result from Alon et al. (2023), followed by the statements and proofs of Lemmas 6.5.2 and 6.5.4, which rely on Theorem 6.5.1 and our boosting result (Theorem 6.2.2). The proof of Theorem 6.2.1 is a consequence of these results, as follows.

Proof of Theorem 6.2.1. The proof follows from:

$$\text{Item } 1 \xLeftrightarrow{\text{Theorem 6.5.1}} \text{Item } 2 \xRightarrow{\text{Lemma 6.5.2}} \text{Item } 4 \xRightarrow{(*)} \text{Item } 3 \xRightarrow{\text{Lemma 6.5.4}} \text{Item } 2,$$

where $(*)$ is immediate. □

Characterization of Pure DP Learnability via the Fractional Clique Dimension

For every hypothesis class $\mathcal{H}$, they define a quantity $\omega_m^\star = \omega_m^\star(\mathcal{H})$, called the *fractional clique number* of $\mathcal{H}$. The definition of $\omega_m^\star$ involves an LP relaxation of clique numbers on a certain graph corresponding to $\mathcal{H}$, but for our purposes it will be more convenient to use the following alternative characterization (Eq. 6 and Theorem 2.8 in Alon et al., 2023):

$$\forall m \in \mathbb{N} : \ \frac{1}{\omega_m^\star} = \sup_{\mathcal{P}} \inf_{\mathcal{S}} \underset{\substack{S\sim\mathcal{S} \\ h\sim\mathcal{P}}}{\mathbb{P}}\left[\mathrm{L}_S^{0\text{-}1}(h) = 0\right], \tag{6.14}$$

where the supremum is taken over distributions over $\mathcal{H}$, and the infimum is taken over distributions over samples of size m that are realizable by $\mathcal{H}$. In words, $1/\omega_m^\star$ is the value of a game in which player 1 selects a distribution of hypotheses over $\mathcal{H}$, player 2 selects a distribution over realizable samples of size m, and player 1 wins if and only if the hypothesis correctly labels all the points in the sample.

The fractional clique number characterizes pure DP learnability, as follows:

Theorem 6.5.1 (Restatement of Theorems 2.3 and 2.6 in Alon et al., 2023)**.** *For any hypothesis class $\mathcal{H}$, exactly one of the following statements holds:*

1. *$\mathcal{H}$ is pure DP learnable (as in Definition 6.3.5), and there exists a polynomial p such that $\omega^\star_m(\mathcal{H}) \leq p(m)$ for all $m \in \mathbb{N}$.*
2. *$\mathcal{H}$ is not pure DP learnable, and $\omega^\star_m(\mathcal{H}) = 2^m$ for all $m \in \mathbb{N}$.*

The *fractional clique dimension* of $\mathcal{H}$ is defined by $\mathsf{CD}^\star(\mathcal{H}) = \sup\{m \in \mathbb{N} : \ \omega^\star_m(\mathcal{H}) = 2^m\}$. So in other words, Theorem 6.5.1 states that $\mathcal{H}$ is pure DP learnable if and only if $\mathsf{CD}^\star(\mathcal{H})$ is finite.

Finite Fractional Clique Dimension $\implies$ DI Rényi-Stability

Lemma 6.5.2. *In the context of Theorem 6.2.1: Item 2 $\implies$ Item 4.*

Proof of Lemma 6.5.2. Given that $\mathcal{H}$ is DP learnable, we define a learning rule A and a prior $\mathcal{P}$, and show that A PAC learns $\mathcal{H}$ subject to distribution-independent KL-stability with respect to $\mathcal{P}$.

By Theorem 6.5.1 there exists a polynomial p such that $\omega^\star_m(\mathcal{H}) \leq p(m)$ for all $m \in \mathbb{N}$. By Eq. (6.14), for every $m \in \mathbb{N}$, there exists a prior $\mathcal{P}_m \in \Delta\Big(\{0,1\}^{\mathcal{X}}\Big)$ such that for any $\mathcal{H}$-realizable sample $S \in (\mathcal{X} \times \{0,1\})^m$,

$$\underset{h\sim\mathcal{P}_m}{\mathbb{P}}\Big[\mathrm{L}^{0\text{-}1}_S(h) = 0\Big] \geq \frac{1}{\omega^\star_m} \geq \frac{1}{p(m)}.$$

Let

$$\mathcal{P} = \frac{1}{z}\sum_{m=1}^{\infty}\frac{\mathcal{P}_m}{m^2}$$

be a mixture, where $z = \sum_{m=1}^{\infty} 1/m^2 = \pi^2/6$ is a normalization factor. $\mathcal{P}$ is a valid distribution over $\{0,1\}^{\mathcal{X}}$.

For every $m \in \mathbb{N}$ and for any $\mathcal{H}$-realizable sample $S \in (\mathcal{X} \times \{0,1\})^m$,

$$\underset{h\sim\mathcal{P}}{\mathbb{P}}\Big[\mathrm{L}^{0\text{-}1}_S(h) = 0\Big] \geq \frac{1}{zm^2} \cdot \underset{h\sim\mathcal{P}_m}{\mathbb{P}}\Big[\mathrm{L}^{0\text{-}1}_S(h) = 0\Big] \geq \frac{1}{zm^2p(m)} = \frac{1}{q(m)}, \tag{6.15}$$

where $q(m) = zm^2p(m)$.

For any sample S, let $C_S = \Big\{h \in \{0,1\}^{\mathcal{X}} : \ \mathrm{L}^{0\text{-}1}_S(h) = 0\Big\}$ be the set of hypotheses consistent with S. Let A be a randomized learning rule given by $S \mapsto \mathcal{Q}_S \in \Delta\Big(\{0,1\}^{\mathcal{X}}\Big)$ such that

$\mathcal{Q}_S(h) = \mathcal{P}(h \mid C_S)$ if $h \in C_S$, and $\mathcal{Q}_S(h) = 0$ otherwise. A can be written explicitly as a rejection sampling algorithm:

```
$A(S)$:
  do:
      sample $h \leftarrow \mathcal{P}$
  while $\mathrm{L}_S^{0\text{-}1}(h) > 0$
  return $h$
```

Algorithm A terminates with probability 1, because for any realizable sample S of size $m \in \mathbb{N}$ and any $t \in \mathbb{N}$,

$$\mathbb{P}[A \text{ did not terminate after } t \text{ iterations}] = \left(\mathbb{P}_{h\sim\mathcal{P}}\left[\mathrm{L}_S^{0\text{-}1}(h) > 0\right]\right)^t \leq \left(1 - \frac{1}{q(m)}\right)^t \xrightarrow{t\to\infty} 0,$$

where the inequality follows by Eq. (6.15).

To complete the proof, we show that A satisfies *(i)*, *(ii)* and *(iii)* in Item 4.

Item *(i)* is immediate from the construction of A. For Item *(ii)*, let $m \in \mathbb{N}$. For any sample S of size m and hypothesis $h \in C_S$,

$$\mathcal{Q}_S(h) = \mathcal{P}(h \mid C_S) = \frac{\mathcal{P}(\{h\} \cap C_S)}{\mathcal{P}(C_S)} \leq q(m) \cdot \mathcal{P}(h), \tag{6.16}$$

where the inequality follows from Eq. (6.15). Hence,

$$\begin{aligned}
\mathsf{D}_\infty(\mathcal{Q}_S \,\|\, \mathcal{P}) &= \log\left(\operatorname*{ess\,sup}_{\mathcal{Q}_S} \frac{\mathcal{Q}_S(h)}{\mathcal{P}(h)}\right) \\
&\leq \log\left(\operatorname*{ess\,sup}_{\mathcal{Q}_S} \frac{q(m)\cdot\mathcal{P}(h)}{\mathcal{P}(h)}\right) \qquad \text{(from Eq. (6.16) and } \mathcal{Q}_S(C_S) = 1\text{)} \\
&\leq \log(q(m)) = O(\log(m)).
\end{aligned}$$

Item *(ii)* follows from monotonicity of D_α with respect to α (Lemma 6.4.1). In particular, $\mathsf{KL}(\mathcal{Q}_S \,\|\, \mathcal{P}) = O(\log(m))$.

Item *(iii)* follows from the PAC-Bayes theorem (Theorem 6.3.2). Indeed, take $\beta = \frac{1}{m}$ and note that $\mathrm{L}_S^{0\text{-}1}(\mathcal{Q}_S) = 0$ for all realizable S. Then for any $\mathcal{H}$-realizable distribution $\mathcal{D}$,

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left[\mathrm{L}_\mathcal{D}^{0\text{-}1}(A(S)) \leq \sqrt{\frac{\mathsf{KL}(\mathcal{Q}_S \,\|\, \mathcal{P}) + 2\ln m}{2(m-1)}}\right] \geq 1 - \frac{1}{m}.$$

This implies that for any $\mathcal{H}$-realizable distribution $\mathcal{D}$,

$$\mathbb{E}_{S\sim\mathcal{D}^m}\left[\mathrm{L}_\mathcal{D}^{0\text{-}1}(A(S))\right] \leq \frac{1}{m} + \sqrt{\frac{\mathsf{KL}(\mathcal{Q}_S \,\|\, \mathcal{P}) + 2\ln m}{2(m-1)}} = O\left(\sqrt{\frac{\log m}{m}}\right),$$

as desired. □

Remark 6.5.3. *The 'furthermore' section of Lemma 6.4.1 implies that in the foregoing proof,* $\mathsf{D}_\alpha(\mathcal{Q}_S \,\|\, \mathcal{P}) = \mathsf{D}_\beta(\mathcal{Q}_S \,\|\, \mathcal{P})$ *for any* $\alpha, \beta \in [0, \infty]$.

DI Rényi-Stability $\implies$ Finite Fractional Clique Dimension

Lemma 6.5.4. *In the context of Theorem 6.2.1: Item 3* $\implies$ *Item 2.*

Proof of Lemma 6.5.4. By Theorem 6.5.1 and Eq. (6.14) it suffices to show that there exist $m \in \mathbb{N}$ and a prior $\mathcal{P}$ such that for every $\mathcal{H}$-realizable sample $S \in (\mathcal{X} \times \{0,1\})^m$,

$$\mathbb{P}_{h\sim\mathcal{P}}\left[\mathrm{L}_S^{0\text{-}1}(h) = 0\right] > \frac{1}{2^m}. \tag{6.17}$$

By the assumption (Item 3) and Theorem 6.2.2, there exists an interpolating learning rule $A^\star$, a prior $\mathcal{P}^\star$, and a constant $C > 0$ such that for every $\mathcal{D} \in \mathsf{Realizable}(\mathcal{H})$, the equality

$$\mathbb{P}_{S\sim\mathcal{D}^m}[\mathsf{KL}(A^\star(S) \,\|\, \mathcal{P}^\star) \le C\log(m)] = 1 \tag{6.18}$$

holds for all $m \in \mathbb{N}$ large enough. Fix such an m. We show that taking $\mathcal{P} = \mathcal{P}^\star$ satisfies Eq. (6.17) for this m.

Let $\mathcal{Q}$ denote the distribution of $A^\star(S')$ where $S' \sim (\mathrm{U}(S))^{m'} = P_{S'}$, $\mathrm{U}(S)$ is the uniform distribution over S, and $m' = m\ln(4m)$. The proof follows by noting that if $\mathsf{KL}(\mathcal{Q} \,\|\, \mathcal{P}^\star)$ is small then one can lower bounding the probability of an event according to $\mathcal{P}^\star$ by its probability according to $\mathcal{Q}$.

To see that the KL is indeed small, let $P_{A^\star(S'),S'}$ and $P_{H^\star,S'}$ be two joint distributions. The variable S' has marginal $P_{S'}$ in both distributions, $A^\star(S') \sim \mathcal{Q}$ depends on S', but $H^\star \sim \mathcal{P}^\star$ is independent of S'. Then,

$$\begin{aligned}
\mathsf{KL}(\mathcal{Q} \,\|\, \mathcal{P}^\star) &= \mathsf{KL}\Big(P_{A^\star(S')} \,\Big\|\, P_{H^\star}\Big) \\
&\le \mathsf{KL}\left(P_{A^\star(S')|S'} \,\|\, P_{H^\star|S'} \,\middle|\, P_{S'}\right) && \text{(Lemma 6.4.5)} \\
&= \mathsf{KL}\left(P_{A^\star(S')|S'} \,\|\, P_{H^\star} \,\middle|\, P_{S'}\right) && (H^\star \perp S') \\
&= \mathbb{E}_{S'}[\mathsf{KL}(A^\star(S') \,\|\, \mathcal{P}^\star)] && \text{(Definition of conditional } \mathsf{KL}) \\
&\le C\log(m). && \text{(By Eq. (6.18) and choice of } m)
\end{aligned} \tag{6.19}$$

Taking $k = 2C\log(m)$,

$$\mathbb{P}_{h\sim\mathcal{Q}}\left[\log\left(\frac{\mathcal{Q}(h)}{\mathcal{P}^\star(h)}\right) \ge k\right] \le \frac{\mathsf{KL}(\mathcal{Q} \,\|\, \mathcal{P}^\star)}{k} \le \frac{1}{2} \tag{6.20}$$

holds by Markov's inequality and the definition of the KL divergence. We are interested in the probability of the event $\mathcal{E} = \left\{h \in \{0,1\}^{\mathcal{X}} : \; \mathrm{L}_S^{0\text{-}1}(h) = 0\right\}$. Because $A^\star$ is interpolating,

$$\mathcal{Q}(\mathcal{E}) \ge \mathbb{P}_{\substack{S'\sim(\mathrm{U}(S))^{m'} \\ h\sim A^\star(S')}}[S \subseteq S'] \ge 1 - m\left(1 - \frac{1}{m}\right)^{m'} \ge \frac{3}{4}. \tag{6.21}$$

Finally, we lower bound $\mathcal{P}^\star(\mathcal{E})$ as follows.

$$\begin{aligned}
\mathcal{P}^\star(\mathcal{E}) &\geq \mathop{\mathbb{P}}_{h\sim\mathcal{P}^\star}\left[\mathcal{E} \ \wedge \ \log\left(\frac{\mathcal{Q}(h)}{\mathcal{P}^\star(h)}\right) \leq k\right] \\
&= \mathop{\mathbb{P}}_{h\sim\mathcal{P}^\star}\left[\mathcal{E} \ \wedge \ \mathcal{P}^\star(h) \geq 2^{-k}\cdot\mathcal{Q}(h)\right] \\
&\geq \mathop{\mathbb{P}}_{h\sim\mathcal{Q}}\left[\mathcal{E} \ \wedge \ \mathcal{P}^\star(h) \geq 2^{-k}\cdot\mathcal{Q}(h)\right]\cdot 2^{-k} \\
&= \mathop{\mathbb{P}}_{h\sim\mathcal{Q}}\left[\mathcal{E} \ \wedge \ \log\left(\frac{\mathcal{Q}(h)}{\mathcal{P}^\star(h)}\right) \leq k\right]\cdot 2^{-k}. \\
&\geq \left(\mathcal{Q}(\mathcal{E}) - \mathop{\mathbb{P}}_{h\sim\mathcal{Q}}\left[\log\left(\frac{\mathcal{Q}(h)}{\mathcal{P}^\star(h)}\right) \leq k\right]\right)\cdot 2^{-k}. && \text{(De Morgan's + union bound)} \\
&\geq \frac{1}{4}\cdot 2^{-k} = \frac{1}{4m^{2C}} = \frac{1}{\mathsf{poly}(m)}. && \text{(By Eqs. (6.20) and (6.21) and choice of } k\text{)}
\end{aligned}$$

This establishes Eq. (6.17), as desired. □

6.6 Proof of Theorem 6.1.4 (DD Equivalences)

Preliminaries

Littlestone Dimension

The Littlestone dimension is a combinatorial parameter which captures mistake and regret bounds in online learning (Littlestone, 1988; Ben-David et al., 2009).

Definition 6.6.1 (Mistake Tree)**.** *A mistake tree is a binary decision tree whose nodes are labeled with instances from $\mathcal{X}$ and edges are labeled by 0 or 1 such that each internal node has one outgoing edge labeled 0 and one outgoing edge labeled 1. A root-to-leaf path in a mistake tree can be described as a sequence of labeled examples $(x_1, y_1), \ldots, (x_d, y_d)$. The point x_i is the label of the i-th internal node in the path, and y_i is the label of its outgoing edge to the next node in the path.*

Definition 6.6.2 (Shattering)**.** *Let $\mathcal{H}$ be a hypothesis class and let T be a mistake tree. $\mathcal{H}$ shatters T if every root-to-leaf path in T is realizable by $\mathcal{H}$.*

Definition 6.6.3 (Littlestone Dimension)**.** *Let $\mathcal{H}$ be a hypothesis class. The Littlestone dimension of $\mathcal{H}$, denoted $\mathsf{LD}(\mathcal{H})$, is the largest number d such that there exists a complete mistake tree of depth d shattered by $\mathcal{H}$. If $\mathcal{H}$ shatters arbitrarily deep mistake trees then $\mathsf{LD}(\mathcal{H}) = \infty$.*

Clique Dimension

Definition 6.6.4 (Clique; Alon et al., 2023)**.** *Let $\mathcal{H}$ be a hypothesis class and let $m \in \mathbb{N}$. A clique in $\mathcal{H}$ of order m is a family $\mathcal{S}$ of realizable samples of size m such that (i) $|\mathcal{S}| = 2^m$; (ii) every two distinct samples $S', S'' \in \mathcal{S}$ contradicts, i.e., there exists a common example $x \in \mathcal{X}$ such that $(x, 0) \in S'$ and $(x, 1) \in S''$.*

Definition 6.6.5 (Clique Dimension; Alon et al., 2023)**.** *Let $\mathcal{H}$ be a hypothesis. The clique dimension of $\mathcal{H}$, denoted $\mathsf{CD}(\mathcal{H})$, is the largest number m such that $\mathcal{H}$ contains a clique of order m. If $\mathcal{H}$ contains cliques of arbitrary large order then we write $\mathsf{CD}(\mathcal{H}) = \infty$.*

Global Stability $\implies$ Replicability

Lemma 6.6.6. *Let $\mathcal{H}$ be a hypothesis class and let A be a (m, η)-globally stable learner for $\mathcal{H}$. Then, A is an η-replicable learner for $\mathcal{H}$.*

This follows immediately by noting that global stability is equivalent to 2-parameters replicability, which is qualitatively equivalent to 1-parameter replicability (Impagliazzo et al., 2022).

Lemma 6.6.7 (Impagliazzo et al., 2022)**.** *For every $\rho, \eta, \nu \in [0, 1]$,*

1. *Every ρ-replicable algorithm is also $\left(\frac{\rho-\nu}{1-\nu}, \nu\right)$-replicable.*
2. *Every (η, ν)-replicable algorithm is also $(\eta + 2\nu - 2)$-replicable.*

Proof of Lemma 6.6.6. By the assumption, there exists a hypothesis h such that for every population $\mathcal{D}$, we have $\mathbb{P}_{R\sim\mathcal{R}}[\mathbb{P}_{S\sim\mathcal{D}^m}[A(S; r) = h] \geq \eta] = 1$. Hence A is $(\eta, 1)$-replicable, and by Lemma 6.6.7 it is also η-replicable. □

DD KL-Stability $\implies$ Finite Littlestone Dimension

Lemma 6.6.8. *Let $\mathcal{H}$ be a hypothesis class that is distribution-dependent KL-stable. Then $\mathcal{H}$ has finite Littlestone dimension.*

This lemma is an immediate result of the relation between thresholds and the Littlestone dimension, and the fact that the class of thresholds on the natural numbers does not admit any learning rule that satisfies a non-vacuous PAC-Bayes bound (Livni and Moran, 2020). The next lemma is a corollary of Theorem 2 in Livni and Moran (2020).

Theorem 6.6.9 (Corollary of Theorem 2, Livni and Moran, 2020)**.** *Let $m \in \mathbb{N}$ and let $N \in \mathbb{N}$. Then, there exists $n \in \mathbb{N}$ large enough such that the following holds. For every learning rule A of the class of thresholds over $[n]$, $\mathcal{H}_n = \{\mathbb{1}_{[x>k]} : [n] \to \{0, 1\} \mid k \in [n]\}$, there exists a realizable population distribution $\mathcal{D} = \mathcal{D}_A$ such that for any prior distribution $\mathcal{P}$,*

$$\underset{S\sim\mathcal{D}^m}{\mathbb{P}}\left[\mathsf{KL}(A(S) \,\|\, \mathcal{P}) > N \quad \text{or,} \quad \mathrm{L}^{0\text{-}1}_{\mathcal{D}}(A(S)) > \frac{1}{4}\right] \geq \frac{1}{16}$$

Theorem 6.6.10 (Littlestone dimension and thresholds, Shelah, 1990)**.** *Let $\mathcal{H}$ be a hypothesis class. Then,*

1. *If $\mathsf{LD}(\mathcal{H}) \geq d$ then $\mathcal{H}$ contains $\lfloor \log d \rfloor$ thresholds.*
2. *If $\mathcal{H}$ contains d thresholds then $\mathsf{LD}(\mathcal{H}) \geq \lfloor \log d \rfloor$.*

Proof of Lemma 6.6.8. If by contradiction the Littlestone dimension of $\mathcal{H}$ is unbounded, then by Theorem 6.6.10, $\mathcal{H}$ contains a copy of $\mathcal{H}_n$, the class of thresholds over $[n]$, for arbitrary large n's. Hence, by Theorem 6.6.9 $\mathcal{H}$ does not admit a PAC learner that is KL-stable. □

MI-Stability $\implies$ DD KL-Stability

Lemma 6.6.11. *Let $\mathcal{H}$ be a hypothesis class and let A be a mutual information stable learner with information bound $f(m) = o(1)$. (I.e. for every population distribution $\mathcal{D}$, $I(A(S); S) \leq f(m)$ where $S \sim \mathcal{D}^m$.) Then, A is a distribution-dependent KL-stable learner with KL bound $g(m) = \sqrt{f(m) \cdot m}$ and confidence $\beta(m) = \sqrt{f(m)/m}$.*

The following statement is an immediate corollary.

Corollary 6.6.12. *Let $\mathcal{H}$ be a hypothesis class that is mutual information stable. Then $\mathcal{H}$ is distribution-dependent KL-stable.*

Proof of Lemma 6.6.11. Let $\mathcal{D}$ be a population distribution. Define a prior distribution $\mathcal{P}_{\mathcal{D}} = \mathbb{E}_S[A(S)]$, i.e. $\mathcal{P}_{\mathcal{D}}(h) = \mathbb{P}_{S\sim\mathcal{D}^m}[A(S) = h]$. We will show that A is KL stable with respect to the prior $\mathcal{P}_{\mathcal{D}}$. We use the identity $I(X; Y) = \mathsf{KL}(P_{X,Y}, P_X P_Y)$. Let $P_{A(S),S}$ be the joint distribution of the training sample S and the hypothesis selected by A when given S as an input, and let $P_{A(S)}P_S$ be the product of the marginals. Note that $P_{A(S)}P_S$ is equal in distribution to $P_{A(S')}P_S$, where S' is an independent copy of S. Hence,

$$\begin{aligned} I(A(S); S) &= \mathsf{KL}(P_{A(S),S}, P_{A(S)}P_S) \\ &= \mathsf{KL}(P_{A(S)|S}P_S, P_{A(S')}P_S), \\ &= \mathsf{KL}(P_S, P_S) + \mathbb{E}_{s\sim P_S}\Big[\mathsf{KL}(P_{A(S)|S=s}, P_{A(S')|S=s})\Big] \qquad \text{(Chain rule)} \\ &= \mathbb{E}_{s\sim P_S}\Big[\mathsf{KL}(P_{A(S)|S=s}, P_{A(S')|S=s})\Big] \\ &= \mathbb{E}_{s\sim P_S}\Big[\mathsf{KL}(P_{A(S)|S=s}, P_{A(S')})\Big]. \end{aligned}$$

Note that $P_{A(S')}$ and the prior $\mathcal{P}_{\mathcal{D}}$ are identically distributed, and $P_{A(S)|S=s}$ is exactly the posterior produced by A given the input sample s. By Markov's inequality,

$$\begin{aligned} \mathop{\mathbb{P}}_{S\sim D^m}\Big[\mathsf{KL}(A(S) \parallel P_{\mathcal{D}}) \geq \sqrt{m \cdot I(A(S); S)}\Big] \leq &\frac{I(A(S); S)}{\sqrt{mI(A(S); S)}} \\ = &\sqrt{\frac{I(A(S); S)}{m}}. \end{aligned} \tag{6.22}$$

Since $I(A(S); S) \leq f(m)$, by Eq. (6.22)

$$\underset{S \sim D^m}{\mathbb{P}}\left[\mathsf{KL}(A(S) \parallel P_{\mathcal{D}}) \geq \sqrt{f(m) \cdot m}\right] \leq \sqrt{\frac{f(m)}{m}}.$$

Note that since $f(m) = o(m)$, indeed $\sqrt{f(m)/m} \xrightarrow{m \to \infty} 0$ and $\sqrt{f(m) \cdot m} = o(m)$. □

Finite Littlestone Dimension $\implies$ MI-Stability

Lemma 6.6.13. *Let $\mathcal{H}$ be a hypothesis class with finite Littlestone dimension. Then $\mathcal{H}$ admits an information stable learner.*

This lemma is a direct consequence of Theorem 2 in Pradeep et al. (2022).

Definition 6.6.14. *The information complexity of a hypothesis class $\mathcal{H}$ is*

$$\mathsf{IC}(\mathcal{H}) = \sup_{|S|} \inf_{A} \sup_{\mathcal{D}} I(A(S); S)$$

where the supremum is over all sample sizes $|S| \in \mathbb{N}$ and the infimum is over all learning rules that PAC learn $\mathcal{H}$.

Theorem 6.6.15 (Theorem 2, Pradeep et al., 2022)**.** *Let $\mathcal{H}$ be a hypothesis class of with Littlestone dimension d. Then the information complexity of $\mathcal{H}$ is bounded by*

$$\mathsf{IC}(\mathcal{H}) \leq 2^d + \log(d+1) + 3 + \frac{3}{e \ln 2}.$$

Proof of Lemma 6.6.13. Since finite information complexity implies that $\mathcal{H}$ admits an information stable learner, the proof follows from Theorem 6.6.15 □

Bibliography

[1] Noga Alon and Joel H. Spencer. *The Probabilistic Method.* John Wiley, 2000. doi:10.1002/0471722154.

[2] Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private PAC learning implies finite littlestone dimension. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 852–860. ACM, 2019. doi:10.1145/3313276.3316312.

[3] Noga Alon, Omri Ben-Eliezer, Yuval Dagan, Shay Moran, Moni Naor, and Eylon Yogev. Adversarial laws of large numbers and optimal regret in online classification. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC 2021: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 447–455. ACM, 2021. doi:10.1145/3406325.3451041.

[4] Noga Alon, Mark Bun, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private and online learnability are equivalent. *Journal of the ACM*, 69(4):28:1–28:34, 2022. doi:10.1145/3526074.

[5] Noga Alon, Shay Moran, Hilla Schefler, and Amir Yehudayoff. A unified characterization of private learnability via graph theory. *ArXiv preprint*, abs/2304.03996, 2023. doi:10.48550/arXiv.2304.03996.

[6] Dana Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106, 1987. doi:10.1016/0890-5401(87)90052-6.

[7] Cem Anil, Guodong Zhang, Yuhuai Wu, and Roger B. Grosse. Learning to give checkable answers with prover-verifier games. *ArXiv preprint*, abs/2108.12099, 2021. doi:10.48550/arXiv.2108.12099.

[8] András Antos and Gábor Lugosi. Strong minimax lower bounds for learning. In Avrim Blum and Michael J. Kearns, editors, *Proceedings of the Ninth Annual Conference on Computational Learning Theory, COLT 1996, Desenzano del Garda, Italy, June 28-July 1, 1996*, pages 303–309. ACM, 1996. doi:10.1145/238061.238160.

[9] András Antos and Gábor Lugosi. Strong minimax lower bounds for learning. *Machine Learning*, 30(1):31–56, 1998. doi:10.1023/A:1007454427662.

[10] Hilal Asi, Jonathan R. Ullman, and Lydia Zakynthinou. From robustness to privacy and back. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 1121–1146. PMLR, 2023. URL `https://proceedings.mlr.press/v202/asi23b.html`.

[11] Association for Computing Machinery. Goldwasser and Micali receive ACM turing award for advances that revolutionized the science of cryptography. Press Release, 2013. URL `https://web.archive.org/web/20170917082059/https://www.acm.org/binaries/content/assets/press_releases/turing_award_2012.pdf`.

[12] Brian Axelrod, Shivam Garg, Vatsal Sharan, and Gregory Valiant. Sample amplification: Increasing dataset size even when learning is impossible. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 442–451. PMLR, 2020. URL `http://proceedings.mlr.press/v119/axelrod20a.html`.

[13] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 21–31, 1991. doi:10.1145/103418.103428.

[14] Maria-Florina Balcan, Eric Blais, Avrim Blum, and Liu Yang. Active property testing. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 21–30. IEEE Computer Society, 2012. doi:10.1109/FOCS.2012.64.

[15] Raef Bassily, Shay Moran, Ido Nachum, Jonathan Shafer, and Amir Yehudayoff. Learners that use little information. In Firdaus Janoos, Mehryar Mohri, and Karthik Sridharan, editors, *Algorithmic Learning Theory, ALT 2018, 7-9 April 2018, Lanzarote, Canary Islands, Spain*, volume 83 of *Proceedings of Machine Learning Research*, pages 25–55. PMLR, 2018. URL `http://proceedings.mlr.press/v83/bassily18a.html`.

[16] Tugkan Batu, Eldar Fischer, Lance Fortnow, Ravi Kumar, Ronitt Rubinfeld, and Patrick White. Testing random variables for independence and identity. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 442–451. IEEE, 2001. doi:10.1109/SFCS.2001.959920.

[17] C. Glenn Begley and Lee M. Ellis. Raise standards for preclinical cancer research. *Nature*, 483(7391):531–533, 2012. doi:10.1038/483531a.

[18] Shai Ben-David and Nadav Eiron. Self-directed learning and its relation to the vc-dimension and to teacher-directed learning. *Machine Learning*, 33(1):87–104, 1998. doi:10.1023/A:1007510732151.

[19] Shai Ben-David, Nadav Eiron, and Eyal Kushilevitz. On self-directed learning. In Wolfgang Maass, editor, *Proceedings of the Eigth Annual Conference on Computational Learning Theory, COLT 1995, Santa Cruz, California, USA, July 5-8, 1995*, pages 136–143. ACM, 1995. doi:10.1145/225298.225314.

[20] Shai Ben-David, Eyal Kushilevitz, and Yishay Mansour. Online learning versus offline learning. *Machine Learning*, 29(1):45–63, 1997. doi:10.1023/A:1007465907571.

[21] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*, 2009. URL `http://www.cs.mcgill.ca/&#126;colt2009/papers/032.pdf#page=1`.

[22] Shalev Ben-David and Shai Ben-David. Learning a classifier when the labeling is known. In Jyrki Kivinen, Csaba Szepesvári, Esko Ukkonen, and Thomas Zeugmann, editors, *Algorithmic Learning Theory – 22nd International Conference, ALT 2011, Espoo, Finland, October 5-7, 2011. Proceedings*, volume 6925 of *Lecture Notes in Computer Science*, pages 440–451. Springer, 2011. doi:10.1007/978-3-642-24412-4_34.

[23] Shalev Ben-David and Eric Blais. A new minimax theorem for randomized algorithms (extended abstract). In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 403–411. IEEE, 2020. doi:10.1109/FOCS46700.2020.00045.

[24] Gyora M. Benedek and Alon Itai. Learnability with respect to fixed distributions. *Theoretical Computer Science*, 86(2):377–390, 1991. doi:10.1016/0304-3975(91)90026-X.

[25] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*, 57(1):289–300, 1995. doi:10.1111/j.2517-6161.1995.tb02031.x.

[26] Avrim Blum and Lunjia Hu. Active tolerant testing. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018*, volume 75 of *Proceedings of Machine Learning Research*, pages 474–497. PMLR, 2018. URL `http://proceedings.mlr.press/v75/blum18a.html`.

[27] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM*, 50(4):506–519, 2003. doi:10.1145/335305.335355.

[28] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the SuLQ framework. In Chen Li, editor, *Proceedings of the Twenty-fourth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 13-15, 2005, Baltimore, Maryland, USA*, pages 128–138. ACM, 2005. doi:10.1145/1065167.1065184.

[29] Manuel Blum and Sampath Kannan. Designing programs that check their work. *Journal of the ACM*, 42(1):269–291, 1995. doi:10.1145/200836.200880.

[30] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery*, 36(4):929–965, 1989. doi:10.1145/76359.76371.

[31] Nick Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, 2014. ISBN 0199678111.

[32] Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002. URL `http://jmlr.org/papers/v2/bousquet02a.html`.

[33] Olivier Bousquet, Steve Hanneke, Shay Moran, Ramon van Handel, and Amir Yehudayoff. A theory of universal learning. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 532–541. ACM, 2021. doi:10.1145/3406325.3451087.

[34] Olivier Bousquet, Steve Hanneke, Shay Moran, Jonathan Shafer, and Ilya O. Tolstikhin. Fine-grained distribution-dependent learning curves. In Gergely Neu and Lorenzo Rosasco, editors, *The Thirty Sixth Annual Conference on Learning Theory, COLT 2023, 12-15 July 2023, Bangalore, India*, volume 195 of *Proceedings of Machine Learning Research*, pages 5890–5924. PMLR, 2023. URL `https://proceedings.mlr.press/v195/bousquet23a.html`.

[35] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996. doi:10.1007/BF00058655.

[36] Andrew Browder. *Mathematical analysis: An introduction*. Springer Science & Business Media, 1996. doi:10.1007/978-1-4612-0715-3.

[37] Nataly Brukhim, Daniel Carmon, Irit Dinur, Shay Moran, and Amir Yehudayoff. A characterization of multiclass learnability. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 943–955. IEEE, 2022. doi:10.1109/FOCS54457.2022.00093.

[38] Mark Bun, Roi Livni, and Shay Moran. An equivalence between private classification and online prediction. In Sandy Irani, editor, *61st IEEE Annual Symposium on*

Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020, pages 389–402. IEEE, 2020. doi:10.1109/FOCS46700.2020.00044.

[39] Mark Bun, Marco Gaboardi, Max Hopkins, Russell Impagliazzo, Rex Lei, Toniann Pitassi, Satchit Sivakumar, and Jessica Sorrell. Stability is stable: Connections between replicability, privacy, and adaptive generalization. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 520–527. ACM, 2023. doi:10.1145/3564246.3585246.

[40] Annalisa Buniello, Jacqueline A. L. MacArthur, Maria Cerezo, Laura W. Harris, James Hayhurst, Cinzia Malangone, Aoife McMahon, Joannella Morales, Edward Mountjoy, Elliot Sollis, et al. The NHGRI-EBI GWAS catalog of published genome-wide association studies, targeted arrays and summary statistics 2019. *Nucleic Acids Research*, 47(D1): D1005–D1012, 2019. doi:10.1093/nar/gky1120.

[41] Ran Canetti and Ari Karchmer. Covert learning: How to learn with an untrusted intermediary. In Kobbi Nissim and Brent Waters, editors, *Theory of Cryptography - 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8-11, 2021, Proceedings, Part III*, volume 13044 of *Lecture Notes in Computer Science*, pages 1–31. Springer, 2021. doi:10.1007/978-3-030-90456-2_1.

[42] Clément L. Canonne. A short note on learning discrete distributions. *ArXiv preprint*, abs/2002.11457, 2020. doi:10.48550/arXiv.2002.11457.

[43] Clément L. Canonne. A survey on distribution testing: Your data is big. But is it blue? *Theory of Computing*, pages 1–100, 2020. doi:10.4086/toc.gs.2020.009.

[44] Clément L. Canonne, Ayush Jain, Gautam Kamath, and Jerry Li. The price of tolerance in distribution testing. In Po-Ling Loh and Maxim Raginsky, editors, *Conference on Learning Theory, 2-5 July 2022, London, UK*, volume 178 of *Proceedings of Machine Learning Research*, pages 573–624. PMLR, 2022. URL `https://proceedings.mlr.press/v178/canonne22a.html`.

[45] Matthias C. Caro, Marcel Hinsche, Marios Ioannou, Alexander Nietner, and Ryan Sweke. Classical verification of quantum learning. *ArXiv preprint*, abs/2306.04843, 2023. doi:10.48550/arXiv.2306.04843.

[46] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, 2006. doi:10.1017/CBO9780511546921.

[47] Nicolò Cesa-Bianchi and Ohad Shamir. Efficient transductive online learning via randomized rounding. In Bernhard Schölkopf, Zhiyuan Luo, and Vladimir Vovk, editors, *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, pages 177–194. Springer, 2013. doi:10.1007/978-3-642-41136-6_16.

[48] Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 47–60. ACM, 2017. doi:10.1145/3055399.3055491.

[49] Alessandro Chiesa and Tom Gur. Proofs of proximity for distribution testing. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPIcs*, pages 53:1–53:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ITCS.2018.53.

[50] Thomas M. Cover. Behavior of sequential predictors of binary sequences. In *Transactions of the Fourth Prague Conference on Information Theory, Statistical Decision Functions, Random Processes*. Academic Press, 1965. URL `https://isl.stanford.edu/~cover/papers/paper3.pdf`.

[51] Rachel Cummings, Katrina Ligett, Kobbi Nissim, Aaron Roth, and Zhiwei Steven Wu. Adaptive learning with robust generalization guarantees. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, volume 49 of *JMLR Workshop and Conference Proceedings*, pages 772–814. JMLR.org, 2016. URL `http://proceedings.mlr.press/v49/cummings16.html`.

[52] Amit Daniely and Shai Shalev-Shwartz. Optimal learners for multiclass problems. In Maria-Florina Balcan, Vitaly Feldman, and Csaba Szepesvári, editors, *Proceedings of The 27th Conference on Learning Theory, COLT 2014, Barcelona, Spain, June 13-15, 2014*, volume 35 of *JMLR Workshop and Conference Proceedings*, pages 287–316. JMLR.org, 2014. URL `http://proceedings.mlr.press/v35/daniely14b.html`.

[53] Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the ERM principle. *Journal of Machine Learning Research*, 16:2377–2404, 2015. doi:10.5555/2789272.2912074.

[54] Constantinos Daskalakis, Themis Gouleakis, Chistos Tzamos, and Manolis Zampetakis. Efficient statistics, in high dimensions, from truncated samples. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 639–649. IEEE, 2018. doi:10.1109/FOCS.2018.00067.

[55] Luc Devroye, László Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag New York, 1996. doi:10.1007/978-1-4612-0711-5.

[56] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional Gaussians and Gaussian mixtures. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 73–84. IEEE, 2017. doi:10.1109/FOCS.2017.16.

[57] Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robustly learning a gaussian: Getting optimal error, efficiently. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2683–2702. SIAM, 2018. doi:10.1137/1.9781611975031.171.

[58] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864, 2019. doi:10.1109/FOCS.2016.85.

[59] John F. Dooley. *History of Cryptography and Cryptanalysis: Codes, Ciphers, and their Algorithms*. Springer, 2018. doi:10.1007/978-3-319-90443-6.

[60] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014. doi:10.1561/0400000042.

[61] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. Springer, 2006. doi:10.1007/11761679_29.

[62] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006. doi:10.1007/11681878_14.

[63] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015. doi:10.1126/science.aaa9375.

[64] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 117–126. ACM, 2015. doi:10.1145/2746539.2746580.

[65] Funda Ergün, Ravi Kumar, and Ronitt Rubinfeld. Fast approximate probabilistically checkable proofs. *Information and Computation*, 189(2):135–159, 2004. doi:10.1016/j.ic.2003.09.005.

[66] Amedeo R. Esposito, Michael Gastpar, and Ibrahim Issa. Robust generalization via f-mutual information. In *IEEE International Symposium on Information Theory, ISIT 2020, Los Angeles, CA, USA, June 21-26, 2020*, pages 2723–2728. IEEE, 2020. doi:10.1109/ISIT44484.2020.9174117.

[67] Amedeo R. Esposito, Michael Gastpar, and Ibrahim Issa. Robust generalization via α-mutual information. *ArXiv preprint*, abs/2001.06399, 2020. doi:10.48550/arXiv.2001.06399.

[68] Fiona Fidler and John Wilcox. Reproducibility of scientific results. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2018 edition, 2018. URL `https://plato.stanford.edu/archives/win2018/entries/scientific-reproducibility`.

[69] Yoav Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995. doi:10.1006/INCO.1995.1136.

[70] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55 (1):119–139, 1997. doi:10.1006/JCSS.1997.1504.

[71] Future of Life Institute. Pause giant AI experiments: An open letter. Press Release, 2023. URL `https://futureoflife.org/open-letter/pause-giant-ai-experiments`.

[72] David Gale and Frank M. Stewart. Infinite games with perfect information. In *Contributions to the theory of games, vol. 2*, Annals of Mathematics Studies, no. 28, pages 245–266. Princeton University Press, Princeton, N. J., 1953. doi:10.1515/9781400881970-014.

[73] Sally A. Goldman and Robert H. Sloan. The power of self-directed learning. *Machine Learning*, 14(1):271–294, 1994. doi:10.1023/A:1022605628675.

[74] Oded Goldreich. *Foundations of Cryptography: Volume I, Basic Tools*. Cambridge university press, 2007. doi:10.1017/CBO9780511546891.

[75] Oded Goldreich. The uniform distribution is complete with respect to testing identity to a fixed distribution. In Oded Goldreich, editor, *Computational Complexity and Property Testing - On the Interplay Between Randomness and Computation*, volume 12050 of *Lecture Notes in Computer Science*, pages 152–172. Springer, 2020. doi:10.1007/978-3-030-43662-9_10.

[76] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32. ACM, 1989. doi:10.1145/73007.73010.

[77] Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. In Oded Goldreich, editor, *Studies in Complexity and Cryptography: Miscellanea on the Interplay between Randomness and Computation*, volume 6650 of *Lecture Notes in Computer Science*, pages 68–75. Springer, 2011. doi:10.1007/978-3-642-22670-0_9.

[78] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998. doi:10.1145/285055.285060.

[79] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989. doi:10.1145/3335741.3335750.

[80] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: Interactive proofs for muggles. *Journal of the ACM*, 62(4):27:1–27:64, 2015. doi:10.1145/2699436.

[81] Shafi Goldwasser, Guy N. Rothblum, Jonathan Shafer, and Amir Yehudayoff. Interactive proofs for verifying machine learning. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 41:1–41:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ITCS.2021.41.

[82] James Hannan. Approximation to Bayes risk in repeated play. In *Contributions to the Theory of Games (AM-39), Volume III*, pages 97–140, Princeton, 1958. Princeton University Press. ISBN 9781400882151. doi:10.1515/9781400882151-006.

[83] Steve Hanneke, Shay Moran, and Jonathan Shafer. A trichotomy for transductive online learning. *CoRR*, abs/2311.06428, 2023. doi:10.48550/ARXIV.2311.06428.

[84] David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100:78–150, 1992. doi:10.1201/9780429492525-4.

[85] David Haussler and Philip M. Long. A generalization of sauer's lemma. *Journal of Combinatorial Theory, Series A*, 71(2):219–240, 1995. doi:10.1016/0097-3165(95)90001-2.

[86] David Haussler, Nick Littlestone, and Manfred K. Warmuth. Predicting $\{0, 1\}$-functions on randomly drawn points. *Information and Computation*, 115(2):248–292, 1994. doi:10.1006/inco.1994.1097.

[87] Úrsula Hébert-Johnson, Michael P. Kim, Omer Reingold, and Guy N. Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on*

Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, volume 80 of *Proceedings of Machine Learning Research*, pages 1944–1953. PMLR, 2018. URL http://proceedings.mlr.press/v80/hebert-johnson18a.html.

[88] Tal Herman and Guy N. Rothblum. Verifying the unseen: interactive proofs for label-invariant distribution properties. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1208–1219. ACM, 2022. doi:10.1145/3519935.3519987.

[89] Joel N. Hirschhorn, Kirk Lohmueller, Edward Byrne, and Kurt Hirschhorn. A comprehensive review of genetic association studies. *Genetics in Medicine*, 4(2):45–61, 2002. doi:10.1097/00125817-200203000-00002.

[90] Wilfrid Hodges. *A Shorter Model Theory*. Cambridge University Press, 1997. ISBN 978-0-521-58713-6.

[91] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, pages 13–30, 1963. doi:10.2307/2282952.

[92] Samuel B. Hopkins, Gautam Kamath, Mahbod Majid, and Shyam Narayanan. Robustness implies privacy in statistical estimation. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 497–506. ACM, 2023. doi:10.1145/3564246.3585115.

[93] W. John Hutchins. The Georgetown-IBM experiment demonstrated in january 1954. In Robert E. Frederking and Kathryn B. Taylor, editors, *Machine Translation: From Real Users to Research*, pages 102–114, Berlin, Heidelberg, 2004. Springer. doi:10.1007/978-3-540-30194-3_12. URL https://web.archive.org/web/20230829180421/https://open.unive.it/hitrade/books/HutchinsFirst.pdf. See expanded version in URL.

[94] Andrew Ilyas, Ajil Jalal, Eirini Asteri, Constantinos Daskalakis, and Alexandros G. Dimakis. The robust manifold defense: Adversarial training using generative models. *ArXiv preprint*, abs/1712.09196, 2017. doi:10.48550/arXiv.1712.09196.

[95] Russell Impagliazzo, Rex Lei, Toniann Pitassi, and Jessica Sorrell. Reproducibility in learning. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 818–831. ACM, 2022. doi:10.1145/3519935.3519973.

[96] John P. A. Ioannidis. Why most published research findings are false. *PLoS Medicine*, 2(8), 2005. doi:10.1371/journal.pmed.1004085.

[97] David Kahn. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet*. Simon and Schuster, 1996. ISBN 9781439103555.

[98] Sham M. Kakade and Adam Kalai. From batch to transductive online learning. In *Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]*, pages 611–618, 2005. URL https://proceedings.neurips.cc/paper/2005/hash/17693c91d9204b7a7646284bb3adb603-Abstract.html.

[99] Alkis Kalavasis, Amin Karbasi, Shay Moran, and Grigoris Velegkas. Statistical indistinguishability of learning algorithms. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 15586–15622. PMLR, 2023. URL https://proceedings.mlr.press/v202/kalavasis23a.html.

[100] Friedrich W. Kasiski. *Die Geheimschriften und die Dechiffrirkunst: Mit besonderer Berücksichtigung der deutschen und der französischen Sprache.* E.S. Mittler und Sohn, 1863. URL https://worldcat.org/title/5792850.

[101] Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998. doi:10.1145/293347.293351.

[102] Michael J. Kearns and Dana Ron. Testing problems with sublearning sample complexity. *Journal of Computer and System Sciences*, 61(3):428–456, 2000. doi:10.1006/jcss.1999.1656.

[103] Christian Kuhlmann. On teaching and learning intersection-closed concept classes. In Paul Fischer and Hans Ulrich Simon, editors, *Computational Learning Theory, 4th European Conference, EuroCOLT 1999, Nordkirchen, Germany, March 29-31, 1999, Proceedings*, volume 1572 of *Lecture Notes in Computer Science*, pages 168–182. Springer, 1999. doi:10.1007/3-540-49097-3_14.

[104] Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the Fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993. doi:10.1137/0222080.

[105] John Langford, Matthias W. Seeger, and Nimrod Megiddo. An improved predictive accuracy bound for averaging classifiers. In Carla E. Brodley and Andrea Pohoreckyj Danyluk, editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 290–297. Morgan Kaufmann, 2001. URL https://dl.acm.org/doi/10.5555/645530.655657.

[106] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms.* Cambridge University Press, 2020. doi:10.1017/9781108571401.

[107] James Lighthill. Artificial intelligence: a general survey. In *Artificial Intelligence: A Paper Symposium.* UK Science Research Council, 1973.

URL https://web.archive.org/web/20150905162940/https://www.aiai.ed.ac.uk/events/lighthill1973/lighthill.pdf.

[108] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, Fourier transform, and learnability. *Journal of the ACM*, 40(3):607–620, 1993. doi:10.1145/174130.174138.

[109] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1988. doi:10.1007/BF00116827.

[110] Roi Livni and Shay Moran. A limitation of the pac-bayes framework. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/ec79d4bed810ed64267d169b0d37373e-Abstract.html.

[111] Yiwen Lu. A bumpy ride for San Francisco's driverless taxis. In *The New York Times*, August 2023. URL https://www.nytimes.com/2023/08/22/us/california-autonomous-vehicles.html.

[112] Maryanthe Malliaris and Shay Moran. The unstable formula theorem revisited. *ArXiv preprint*, abs/2212.05050, 2022. doi:10.48550/arXiv.2212.05050.

[113] Yishay Mansour. Learning boolean functions via the Fourier transform. In *Theoretical advances in neural computation and learning*, pages 391–424. Springer, 1994. doi:10.1007/978-1-4615-2696-4_11.

[114] Urko M. Marigorta, Juan Antonio Rodríguez, Greg Gibson, and Arcadi Navarro. Replicability and prediction: lessons and challenges from GWAS. *Trends in Genetics*, 34(7):504–517, 2018. doi:10.1016/j.tig.2018.03.005.

[115] David A. McAllester. Some PAC-Bayesian theorems. *Machine Learning*, 37(3):355–363, 1999. doi:10.1023/A:1007618624809.

[116] David A. McAllester. Simplified pac-bayesian margin bounds. In Bernhard Schölkopf and Manfred K. Warmuth, editors, *Computational Learning Theory and Kernel Machines, 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003, Proceedings*, volume 2777 of *Lecture Notes in Computer Science*, pages 203–215. Springer, 2003. doi:10.1007/978-3-540-45167-9_16.

[117] Silvio Micali. CS proofs. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 436–453, 1994. doi:10.1109/SFCS.1994.365746.

[118] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. Adaptive computation and machine learning. MIT Press, 2018. URL https://mitpress.mit.edu/9780262039406.

[119] Shay Moran, Hilla Schefler, and Jonathan Shafer. The bayesian stability zoo. *CoRR*, abs/2310.18428, 2023. doi:10.48550/ARXIV.2310.18428.

[120] Saachi Mutreja and Jonathan Shafer. PAC verification of statistical algorithms. *ArXiv preprint*, abs/2211.17096, 2022. URL https://arxiv.org/abs/2211.17096v1.

[121] Saachi Mutreja and Jonathan Shafer. PAC verification of statistical algorithms. In Gergely Neu and Lorenzo Rosasco, editors, *The Thirty Sixth Annual Conference on Learning Theory, COLT 2023, 12-15 July 2023, Bangalore, India*, volume 195 of *Proceedings of Machine Learning Research*, pages 5021–5043. PMLR, 2023. URL https://proceedings.mlr.press/v195/mutreja23a.html.

[122] Balas K. Natarajan. On learning sets and functions. *Machine Learning*, 4:67–97, 1989. doi:10.1007/BF00114804.

[123] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. doi:10.1017/CBO9781139814782.

[124] Cameron Palmer and Itsik Pe'er. Statistical correction of the winner's curse explains replication variability in quantitative trait genome-wide association studies. *PLoS Genetics*, 13(7), 2017. doi:10.1371/journal.pgen.1006916.

[125] Liam Paninski. A coincidence-based test for uniformity given very sparsely sampled discrete data. *IEEE Transactions on Information Theory*, 54(10):4750–4755, 2008. doi:10.1109/TIT.2008.928987.

[126] Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *Journal of Computer and System Sciences*, 72(6):1012–1042, 2006. doi:10.1016/j.jcss.2006.03.002.

[127] Harold Pashler and Eric-Jan Wagenmakers. Editors' introduction to the special section on replicability in psychological science: A crisis of confidence? *Perspectives on Psychological Science*, 7(6):528–530, 2012. doi:10.1177/1745691612465253.

[128] Dmitry Pechyony. *Theory and Practice of Transductive Learning*. PhD thesis, Technion – Israel Institute of Technology, Israel, 2008. URL https://technion.primo.exlibrisgroup.com/permalink/972TEC_INST/q1jq5o/alma990023032150203971.

[129] Itsik Pe'er, Roman Yelensky, David Altshuler, and Mark J Daly. Estimation of the multiple testing burden for genomewide association studies of nearly all common variants. *Genetic Epidemiology: The Official Publication of the International Genetic Epidemiology Society*, 32(4):381–385, 2008. doi:10.1002/gepi.20303.

[130] David L. Phillips. A technique for the numerical solution of certain integral equations of the first kind. *Journal of the ACM*, 9(1):84–97, 1962. doi:10.1145/321105.321114.

[131] Yury Polyanskiy and Yihong Wu. *Information Theory: From Coding to Learning*. Forthcoming, Cambridge University Press, 2023+. URL `https://people.lids.mit.edu/yp/homepage/data/itbook-export.pdf`.

[132] Aditya Pradeep, Ido Nachum, and Michael Gastpar. Finite Littlestone dimension implies finite information complexity. In *IEEE International Symposium on Information Theory, ISIT 2022, Espoo, Finland, June 26 - July 1, 2022*, pages 3055–3060. IEEE, 2022. doi:10.1109/ISIT50566.2022.9834457.

[133] Florian Prinz, Thomas Schlange, and Khusru Asadullah. Believe it or not: how much can we rely on published data on potential drug targets? *Nature Reviews Drug Discovery*, 10(9):712–712, 2011. doi:10.1038/nrd3439-c1.

[134] Sofya Raskhodnikova, Dana Ron, Amir Shpilka, and Adam D. Smith. Strong lower bounds for approximating distribution support size and the distinct elements problem. *SIAM Journal on Computing*, 39(3):813–842, 2009. doi:10.1137/070701649.

[135] Alfréd Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 547–562. University of California Press, 1961. URL `https://worldcat.org/title/25234660`.

[136] Reverse Fatou's Lemma. In *ProofWiki*. August 2022. URL `https://proofwiki.org/wiki/Reverse_Fatou%27s_Lemma`.

[137] Sebastien Roch. *Modern Discrete Probability: An Essential Toolkit*. Cambridge University Press, 2023. doi:10.1017/9781009305129.

[138] Dana Ron and Gilad Tsur. On approximating the number of relevant variables in a function. *ACM Transactions on Computation Theory*, 5(2):7:1–7:19, 2013. doi:10.1145/2493246.2493250.

[139] Guy N. Rothblum, Salil P. Vadhan, and Avi Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 793–802. ACM, 2013. doi:10.1145/2488608.2488709.

[140] Ronitt Rubinfeld and Arsen Vasilyan. Testing distributional assumptions of learning algorithms. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1643–1656. ACM, 2023. doi:10.1145/3564246.3585117.

[141] Stuart Russell. *Human Compatible: Artificial Intelligence and the Problem of Control*. Viking, 2019. ISBN 0525558616.

[142] Norbert Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145–147, 1972. doi:10.1016/0097-3165(72)90019-2.

[143] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990. doi:10.1007/BF00116037.

[144] Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. MIT Press, 2012. doi:10.7551/mitpress/8291.001.0001.

[145] Dale Schuurmans. Characterizing rational versus exponential learning curves. *Journal of Computer and System Sciences*, 55(1):140–160, 1997. doi:10.1006/jcss.1997.1505.

[146] Sanjit A. Seshia, Dorsa Sadigh, and S. Shankar Sastry. Toward verified artificial intelligence. *Communications of the ACM*, 65(7):46–55, 2022. doi:10.1145/3503914.

[147] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014. doi:10.1017/CBO9781107298019.

[148] Saharon Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261, 1972. doi:10.2140/pjm.1972.41.247.

[149] Saharon Shelah. *Classification Theory: And the Number of Non-Isomorphic Models*, volume 92 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1990. ISBN 978-0-444-70260-9.

[150] Herbert A. Simon. *The New Science of Management Decision*. Harper & Brothers, 1960. doi:10.1037/13978-000.

[151] Herbert A. Simon. *The Shape of Automation for Men and Management*. Harper & Row, 1965. URL `https://worldcat.org/title/175200`.

[152] Simon Singh. *The Code Book: The Evolution of Secrecy from Mary, Queen of Scots, to Quantum Cryptography*. Doubleday, USA, 1999. ISBN 0307787842.

[153] Charles J. Stone. Consistent nonparametric regression. *The Annals of Statistics*, pages 595–620, 1977. doi:10.1214/aos/1176343886.

[154] John D. Storey and Robert Tibshirani. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences*, 100(16):9440–9445, 2003. doi:10.1073/pnas.1530509100.

[155] Vasilis Syrgkanis, Akshay Krishnamurthy, and Robert E. Schapire. Efficient algorithms for adversarial contextual learning. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2159–2168. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/syrgkanis16.html.

[156] Andrey N. Tikhonov. On the stability of inverse problems. *Proceedings of the USSR Academy of Sciences*, 39:195–198, 1943. URL https://api.semanticscholar.org/CorpusID:202866372.

[157] Gregory Valiant. *Algorithmic Approaches to Statistical Questions*. PhD thesis, UC Berkeley, 2012. URL https://dl.acm.org/doi/10.5555/2520412.

[158] Gregory Valiant and Paul Valiant. A CLT and tight lower bounds for estimating entropy. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:179, 2010. URL http://eccc.hpi-web.de/report/2010/179.

[159] Gregory Valiant and Paul Valiant. Estimating the unseen: A sublinear-sample canonical estimator of distributions. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:180, 2010. URL http://eccc.hpi-web.de/report/2010/180.

[160] Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11): 1134–1142, 1984. doi:10.1145/1968.1972.

[161] Paul Valiant. Testing symmetric properties of distributions. *SIAM Journal on Computing*, 40(6):1927–1968, 2011. doi:10.1137/080734066.

[162] Tim van Erven and Peter Harremoës. Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014. doi:10.1109/TIT.2014.2320500.

[163] Ramon van Handel. The universal Glivenko-Cantelli property. *Probability and Related Fields*, 155:911–934, 2013. doi:10.1007/s00440-012-0416-5.

[164] Vladimir N. Vapnik. *Estimation of Dependencies Based on Empirical Data*. Springer-Verlag, New York, 1982. doi:10.1007/0-387-34239-7.

[165] Vladimir N. Vapnik and Alexey Y. Chervonenkis. The uniform convergence of frequencies of the appearance of events to their probabilities. *Doklady Akademii Nauk*, 181(4): 781–783, 1968. URL http://www.ams.org/mathscinet-getitem?mr=0231431.

[166] Vladimir N. Vapnik and Alexey Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971. doi:10.1007/978-3-319-21852-6_3.

[167] Vladimir N. Vapnik and Alexey Y. Chervonenkis. *Theory of Pattern Recognition*. Nauka, Moscow, 1974. URL `https://worldcat.org/title/246651066`.

[168] Michael Walfish and Andrew J. Blumberg. Verifying computations without reexecuting them. *Communications of the ACM*, 58(2):74–84, 2015. doi:10.1145/2641562.

[169] Benjamin Weiser. Here's what happens when your lawyer uses ChatGPT. In *The New York Times*, May 2023. URL `https://www.nytimes.com/2023/05/27/nyregion/avianca-airline-lawsuit-chatgpt.html`.

[170] Benjamin Weiser. ChatGPT lawyers are ordered to consider seeking forgiveness. In *The New York Times*, June 2023. URL `https://www.nytimes.com/2023/06/22/nyregion/lawyers-chatgpt-schwartz-loduca.html`.

[171] Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 2524–2533, 2017. URL `https://proceedings.neurips.cc/paper/2017/hash/ad71c82b22f4f65b9398f76d8be4c615-Abstract.html`.

[172] Bin Yu. Assouad, Fano, and Le Cam. In *Festschrift for Lucien Le Cam*, pages 423–435. Springer, 1997. doi:10.1007/978-1-4612-1880-7_29.

[173] Yu Yu and John Steinberger. Pseudorandom functions in almost constant depth from low-noise LPN. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 154–183. Springer, 2016. doi:10.1007/978-3-662-49896-5_6.

Appendix A

Appendices for Chapter 3

A.1 Proof of Lower Bound for Littlestone Classes

Proof of Theorem 3.3.1. Let T be a Littlestone tree of depth d that is shattered by $\mathcal{H}$, and let $\mathcal{H}_1 \subseteq \mathcal{H}$ be a collection of 2^{d+1} functions that witness the shattering. T contains $n_T = 2^{d+1} - 1$ nodes. The adversary selects the sequence

$$x_1, x_2, \dots, x_n$$

consisting of the first n nodes of T in breadth-first order (if $n > n_T$, then the adversary chooses the suffix $x_{n_T+1}, \dots, x_n$ arbitrarily). For each time step $t \in [n]$, let $\mathcal{H}_t$ denote the version space, i.e., the subset of $\mathcal{H}_1$ that is consistent with all previously-assigned labels. Namely, for any $t > 1$,

$$\mathcal{H}_t = \{h \in \mathcal{H}_1 : \ (\forall s \in [t-1] : \ h(x_s) = y_s)\} .$$

Similarly, for each $b \in \{0, 1\}$, let $\mathcal{H}_{t,b} = \{h \in \mathcal{H}_t : \ h(x_t) = b\}$.

The adversary operates according to Algorithm A.1. Conceptually, at each time step $t \in [n]$, if $\mathcal{H}_t$ is very unbalanced, meaning that a large majority of the functions in $\mathcal{H}_t$ assign the same value to x_t, then the adversary chooses y_t to be that value. Otherwise, if $\mathcal{H}_t$ is fairly balanced, the adversary forces a mistake. Note that if $\mathcal{H}_t$ is fairly balanced then the adversary can force a mistake without violating $\mathcal{H}$-realizability.

We now argue that using this strategy, the adversary forces $\Omega(\log(d))$ mistakes. Let $F = \{t_1, t_2, \dots\} = \{t \in [n] : \ r_t \in [\varepsilon_t, 1 - \varepsilon_t]\}$ be the set of time steps where the adversary forces a mistake. Note that in the for-loop in Algorithm A.1, the value of k at the beginning of iteration t_k is k (e.g., at the beginning of iteration t_3, $k = 3$).

We argue by induction that for any $k \in \mathbb{N}$, if $m_k := 2^{2^{2k}} \leq n$ then:

1. $|F| \geq k$ and $t_k \leq m_k$; and
2. $|\mathcal{H}_{t_k}| \geq (1/m_k)^2 \cdot |\mathcal{H}_1|$.

```
send $x_1, \ldots, x_n$ to learner
$k \leftarrow 1$
for  $t \leftarrow 1, 2, \ldots, n$:
    $m_k \leftarrow 2^{2^{2k}}$
    $\varepsilon_t \leftarrow 1/m_k$
    $r_t \leftarrow |\mathcal{H}_{t,1}|/|\mathcal{H}_t|$
    receive $\hat{y}_t$ from learner
    $y_t \leftarrow \begin{cases} 1 - \hat{y}_t & r_t \in [\varepsilon_t, 1 - \varepsilon_t] \\ 0 & r_t \in [0, \varepsilon_t) \\ 1 & r_t \in (1 - \varepsilon_t, 1] \end{cases}$
    send $y_t$ to learner
    if  $r_t \in [\varepsilon_t, 1 - \varepsilon_t]$:
        $k \leftarrow k + 1$
```

Algorithm A.1: An adversary that forces $\Omega(\log(\mathsf{LD}(\mathcal{H})))$ mistakes.

The base case is immediate for $t_1 = 1 \in F$. For the induction step, assuming that Items 1 and 2 hold for some $k \in \mathbb{N}$ such that $m_{k+1} \leq n$, we show that they also hold for $k + 1$. For Item 1, assume for contradiction that $t \notin F$ for all t such that $t_k < t \leq m_{k+1}$.

For each t, $t_k < t \leq m_{k+1}$, the definition of r_t and the adversary's labeling strategy imply that the label y_t agrees with at least a $(1 - \varepsilon_t)$-majority of the functions in the version space $\mathcal{H}_t$. Hence,

$$\begin{aligned} \left|\mathcal{H}_{m_{k+1}}\right| &\geq |\mathcal{H}_{t_k}| \cdot \prod_{t=t_k+1}^{m_{k+1}} (1 - \varepsilon_t) \\ &= |\mathcal{H}_{t_k}| \cdot (1 - 1/m_{k+1})^{m_{k+1} - t_k} \\ &\geq |\mathcal{H}_{t_k}| \cdot (1 - 1/m_{k+1})^{m_{k+1}} \\ &\geq |\mathcal{H}_1| \cdot (1/m_k)^2 \cdot (1 - 1/m_{k+1})^{m_{k+1}} && \text{(Induction hypothesis for Item 2)} \\ &\geq |\mathcal{H}_1| \cdot (1/m_k)^2 \cdot (1/4) \\ &= |\mathcal{H}_1| \cdot 2^{-2^{2k+1} - 2}. && \text{(A.1)} \end{aligned}$$

Observe that for every $t \in [n]$, if x_t is a node with depth ℓ in T (i.e., the shortest path from the root to x_t contains ℓ edges), then there exists an 'active' node x^*_ℓ with the same depth ℓ in T such that the version space $\mathcal{H}_t$ contains only functions from $\mathcal{H}_1$ that are consistent with the labels along the path from the root of T to x^*_ℓ. Namely, $\mathcal{H}_t$ is a subset of the $2^{d-\ell+1}$ functions in $\mathcal{H}_1$ that witness the shattering of the subtree T_ℓ of T that is rooted at x^*_ℓ. In particular, the depth (distance from the root) of node $x_{m_{k+1}}$ is $\log\left(2^{2^{2(k+1)}}\right) = 2^{2k+2}$, so

$$\left|\mathcal{H}_{m_{k+1}}\right| \leq 2^{d-2^{2k+2}+1} = 2^{d+1}\cdot 2^{-2^{2k+2}} = |\mathcal{H}_1|\cdot 2^{-2^{2k+2}}. \tag{A.2}$$

Combining Eqs. (A.1) and (A.2) yields $2^{-2^{2k+1}-2} \leq 2^{-2^{2k+2}}$, which is a contradiction. This establishes Item 1. Item 2 follows by a similar calculation, which accounts for the fact that at time t_{k+1} the adversary forces a mistake, and this reduces the version space by a factor of at most $\varepsilon_{t_{k+1}}$:

$$\begin{aligned}
\left|\mathcal{H}_{t_{k+1}}\right| &\geq |\mathcal{H}_{t_k}|\cdot\left(\prod_{t=t_k+1}^{t_{k+1}-1}(1-\varepsilon_t)\right)\cdot\varepsilon_{t_{k+1}} \\
&\geq |\mathcal{H}_{t_k}|\cdot(1-1/m_{k+1})^{m_{k+1}}\cdot(1/m_{k+1}) \\
&\geq |\mathcal{H}_{t_k}|\cdot(1/4)\cdot(1/m_{k+1}) \\
&\geq |\mathcal{H}_1|\cdot(1/m_k)^2\cdot(1/4)\cdot(1/m_{k+1}) && \text{(Induction hypothesis for Item 2)} \\
&= |\mathcal{H}_1|\cdot 2^{-2\cdot 2^{2k}}\cdot(1/4)\cdot 2^{-2^{2k+2}} = |\mathcal{H}_1|\cdot 2^{-6\cdot 2^{2k}-2} \\
&\geq |\mathcal{H}_1|\cdot 2^{-8\cdot 2^{2k}} = |\mathcal{H}_1|\cdot 2^{-2\cdot 2^{2(k+1)}} = |\mathcal{H}_1|\cdot(1/m_{k+1})^2.
\end{aligned}$$

This completes the induction.

To complete the proof, let $k^* = \min\{\lfloor\log(d)/2\rfloor, \lfloor\log\log(n)/2\rfloor\}$. Then $m_{k^*} \leq 2^d < 2^{d+1}-1 = n_T$, so T contains at least m_{k^*} nodes. Additionally, $m_{k^*} \leq n$, so Item 1 implies that $|F| \geq k^*$, namely, the adversary can force at least k^* mistakes, as desired. □

A.2 Multiclass Trichotomy

The following generalization of the Littlestone dimension to the multiclass setting is due to Daniely, Sabato, Ben-David, and Shalev-Shwartz (2015).

Definition A.2.1 (Multiclass Littlestone Dimension)**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be sets and let $d \in \mathbb{N}$. A Littlestone tree of depth d with domain $\mathcal{X}$ and label set $\mathcal{Y}$ is a set*

$$T = \left\{(x_u, y_{u\circ 0}, y_{u\circ 1}) \in \mathcal{X}\times\mathcal{Y}\times\mathcal{Y} :\ u \in \bigcup_{s=0}^{d}\{0,1\}^s \ \wedge\ y_{u\circ 0} \neq y_{u\circ 1}\right\}, \tag{A.3}$$

where '$\circ$' denotes string concatenation. Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$. We say that $\mathcal{H}$ shatters a tree T as in Eq. (A.3) if for every $u \in \{0,1\}^{d+1}$ there exists $h_u \in \mathcal{H}$ such that

$$\forall i \in [d+1]:\ h(x_{u_{\leq i-1}}) = y_{u_{\leq i}}.$$

The Littlestone dimension of $\mathcal{H}$, denoted $\mathsf{LD}(\mathcal{H})$, is the supremum over all $d \in \mathbb{N}$ such that there exists a Littlestone tree of depth d with domain $\mathcal{X}$ and label set $\mathcal{Y}$ that is shattered by $\mathcal{H}$.

The Natarajan dimension is a popular generalization of the VC dimension to the multiclass setting.

Definition A.2.2 (Natarajan, 1989)**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be sets, let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, let $d \in \mathbb{N}$, and let $X = \{x_1, \ldots, x_d\} \subseteq \mathcal{X}$. We say that $\underline{\mathcal{H}\text{ Natarajan-shatters }X}$ if there exist $f_0, f_1 : X \to \mathcal{Y}$ such that:*

1. *$\forall x \in X: \ f_0(x) \neq f_1(x)$; and*
2. *$\forall A \subseteq X \ \exists h \in \mathcal{H} \ \forall x \in X: \ h(x) = f_{\mathbb{1}(x \in A)}(x)$.*

The $\underline{\text{Natarajan dimension}}$ of $\mathcal{H}$ is

$$\mathsf{ND}(\mathcal{H}) = \sup\left\{|X| : \ X \subseteq \mathcal{X} \textit{ finite } \wedge \ \mathcal{H} \textit{ Natarajan-shatters } X\right\}.$$

We show the following generalization of Theorem 3.4.1 for the multiclass setting.

Theorem A.2.3 (**Formal Version of Theorem 3.5.1**)**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be sets with $k = |\mathcal{Y}| < \infty$, let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, and let $n \in \mathbb{N}$ such that $n \leq |\mathcal{X}|$.*

1. *If $\mathsf{ND}(\mathcal{H}) = \infty$ then $M(\mathcal{H}, n) = n$.*
2. *Otherwise, if $\mathsf{ND}(\mathcal{H}) = d < \infty$ and $\mathsf{LD}(\mathcal{H}) = \infty$ then*

$$\max\{\min\{d, n\}, \lfloor \log(n) \rfloor\} \leq M(\mathcal{H}, n) \leq O(d \log(nk/d)). \tag{A.4}$$

   The $\Omega(\cdot)$ and $O(\cdot)$ notations hide universal constants that do not depend on $\mathcal{X}$, $\mathcal{Y}$ or $\mathcal{H}$.

3. *Otherwise, there exists a number $C(\mathcal{H}) \in \mathbb{N}$ (that depends on $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{H}$ but does not depend on n) such that $M(\mathcal{H}, n) \leq C(\mathcal{H})$.*

The proof of Theorem A.2.3 uses the following generalization of the Sauer–Shelah–Perles lemma.

Theorem A.2.4 (Natarajan, 1989; Corollary 5 in Haussler and Long, 1995)**.** *Let $d, n, k \in \mathbb{N}$, let $\mathcal{X}$ and $\mathcal{Y}$ be sets of cardinality n and k respectively, and let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ such that $\mathsf{ND}(\mathcal{H}) \leq d$. Then*

$$|\mathcal{H}| \leq \sum_{i=0}^{d} \binom{n}{i} \binom{k+1}{2}^{i} \leq \left(\frac{enk^2}{d}\right)^{d}.$$

Proof of Theorem A.2.3. Items 1 and 3 and the $\min\{d, n\}$ lower bound in Item 2 follow similarly to the corresponding items in Theorem 3.4.1. The upper bound in Item 2 also follows similarly to the corresponding item in Theorem 3.4.1, except that it uses Theorem A.2.4 instead of the Sauer–Shelah–Perles lemma. The $\lfloor \log(n) \rfloor$ lower bound in Item 2 follows from Theorem A.4.5 and Claim A.4.4. □

A.3 Combinatorics of Trees

In this section we present a simple lemma from Ramsey theory about trees that is used for proving Theorem A.4.5. We start with a generalized definition of subtrees.

Definition A.3.1. *Let X be a finite set and let $(X, \preceq)$ be a partial order relation. For $p, c \in X$, we say that c is a child of p if $p \preceq c$ and there does not exist $m \in X$ such that $p \preceq m \preceq c$. We say that $z \in X$ is a leaf if there exists no $x \in X$ such that $z \preceq x$. $(X, \preceq)$ is a binary tree if every non-leaf $x \in X$ has precisely 2 children. The depth of $z \in X$ is the largest $d \in \mathbb{N}$ for which there exist distinct $x_1, \ldots, x_d \in X$ such that $x_1 \preceq x_2 \preceq \cdots \preceq x_d \preceq z$. For $d \in \mathbb{N}$, we say that $(X, \preceq)$ is a complete binary tree of depth d if $(X, \preceq)$ is a binary tree and all the leaves in X have depth d. We say that a partial order $(X', \preceq')$ is a subtree of $(X, \preceq)$ if $X' \subseteq X$, and $\forall a, b \in X' : \ a \preceq' b \iff a \preceq b$.*

Lemma A.3.2 (Lemma 16 in Alon et al., 2019)**.** *Let $p, q \in \mathbb{R}$ be non-negative such that $p + q \in \mathbb{N}$. Let $T = (X, \preceq)$ be a complete binary tree of depth $d = p + q - 1$, and let $f : \ X \to \{0, 1\}$. Then at least one of the following statements holds:*

- *T has a 0-monochromatic complete binary subtree of depth at least p. Namely, there exists $T' = (X', \preceq')$ such that T' is a subtree of T, T' is a complete binary tree of depth at least p, and $f(x) = 0$ for all $x \in X'$.*

- *T has a 1-monochromatic complete binary tree subtree of depth at least q.*

For completeness, we include a proof of this lemma.

Proof of Lemma A.3.2. We prove the claim by induction on the depth d. The base case of $d = 0$ (a tree with a single node) is immediate. For the induction step, let a denote the root of T, and let T_ℓ and T_r denote the subtrees of T of depth $d - 1$ consisting of all descendants of the left and right child of a respectively. Assume that $f(a) = 0$. If T_ℓ or T_r contain a 1-monochromatic subtree of depth at least q, then we are done. Otherwise, by the induction hypothesis, both trees contain a 0-monochromatic subtree of depth at least $p - 1$. Joining these two subtrees as children of the root a yields a 0-monochromatic subtree of depth at least p, as desired. The proof for the case $f(a) = 1$ is similar. □

We use the following corollary of Lemma A.3.2.

Lemma A.3.3. *Let $k, d \in \mathbb{N}$. Let $T = (X, \preceq)$ be a complete binary tree of depth $d \in \mathbb{N}$, and let $f : \ X \to [k]$. Then T has an f-monochromatic complete binary subtree $T' = (X', \preceq')$ of depth at least*

$$d' = \frac{d+1}{2^{\lceil \log(k) \rceil}}.$$

Namely, there exists T' such that T' is a subtree of T, T' is a complete binary tree of depth at least d', and $|\{f(a) : \ a \in X'\}| = 1$.

Proof of Lemma A.3.3. We will show that for any $b \in \mathbb{N}$, if $k \leq 2^b$ then there exists an f-monochromatic subtree of T of depth at least

$$\frac{d+1}{2^b}.$$

This implies the lemma, which corresponds to the special case $b = \lceil \log(k) \rceil$.

We proceed by induction on b. The base case $b = 1$ follows from Lemma A.3.2. For the induction step, we assume that the claim holds for b and prove that it holds for $b+1$. Namely, we show that if $f : X \to [k]$ and $k \leq 2^{b+1}$ then there exists an f-monochromatic subtree of depth at least $(d+1)/2^{b+1}$.

Define $g : X \to \{1, 2\}$ by $g(x) = 1 + (f(x) \mod 2)$. By Lemma A.3.2, there exists a g-monochromatic complete binary subtree $T_0 = (X_0, \preceq)$ of T of depth at least $(d+1)/2$. In particular $|\{f(x) : x \in X_0\}| \leq 2^b$. By invoking the induction hypotheses on T_0, there exists a complete binary subtree of T_0 that is f-monochromatic and has depth at least

$$\frac{\frac{d+1}{2}+1}{2^b} > \frac{d+1}{2^{b+1}},$$

as desired. □

A.4 Multiclass Threshold Bounds

Definition A.4.1. *Let $\mathcal{X}$ and $\mathcal{Y}$ be sets, let $X = \{x_1, \dots, x_t\} \subseteq \mathcal{X}$, and let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$. We say that X is threshold-shattered by $\mathcal{H}$ if there exist distinct $y_0, y_1 \in \mathcal{Y}$ and functions $h_1, \dots, h_t \in \mathcal{H}$ such that $h_i(x_j) = y_{\mathbb{1}(j \leq i)}$. The threshold dimension of $\mathcal{H}$, denoted $\mathsf{TD}(\mathcal{H})$, is the supremum of the set of integers t for which there exists a threshold-shattered set of cardinality t.*

We introduce the following generalization of the threshold dimension.

Definition A.4.2. *Let $\mathcal{X}$ and $\mathcal{Y}$ be sets, let $X = \{x_1, \dots, x_t\} \subseteq \mathcal{X}$, and let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$. We say that X is multi-class threshold-shattered by $\mathcal{H}$ if there exist $y_1, y'_1 \dots, y_t, y'_t \in \mathcal{Y}$ such that $y_i \neq y'_j$ for all $i, j \in [t]$, and there exist functions $h_1, \dots, h_t \in \mathcal{H}$ such that*

$$h_i(x_j) = \begin{cases} y_i & (j \leq i) \\ y'_j & (j > i). \end{cases}$$

The multi-class threshold dimension of $\mathcal{H}$, denoted $\mathsf{MTD}(\mathcal{H})$, is the supremum of the set of integers t for which there exists a threshold-shattered set of cardinality t.

See Table A.1 for an illustration of this definition.

Claim A.4.3. *Let $\mathcal{X}$ and $\mathcal{Y}$ be sets, $k = |\mathcal{Y}| < \infty$, and let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$. Then $\mathsf{TD}(\mathcal{H}) \geq \lfloor \mathsf{MTD}(\mathcal{H})/k^2 \rfloor$.*

	x_1	x_1	x_3	x_4	x_5
h_1	y_1	$\mathbf{y}'_2$	$\mathbf{y}'_3$	$\mathbf{y}'_4$	$\mathbf{y}'_5$
h_2	y_2	y_2	$\mathbf{y}'_3$	$\mathbf{y}'_4$	$\mathbf{y}'_5$
h_3	y_3	y_3	y_3	$\mathbf{y}'_4$	$\mathbf{y}'_5$
h_4	y_4	y_4	y_4	y_4	$\mathbf{y}'_5$
h_5	y_5	y_5	y_5	y_5	y_5

Table A.1: An illustration of Definition A.4.2. The table shows a collection of points $\{x_1, \ldots, x_5\}$ that are multi-class threshold shattered by functions $\{h_1, \ldots, h_5\}$.

Proof of Claim A.4.3. The proof follows from two applications of the pigeonhole principle. □

Claim A.4.4. *Let $\mathcal{X}$ and $\mathcal{Y}$ be sets, let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ such that $d = \mathsf{TD}(\mathcal{H}) < \infty$, and let $n \in \mathbb{N}$. Then*

$$M(\mathcal{H}, n) \geq \min\{\lfloor \log(d) \rfloor, \lfloor \log(n) \rfloor\}.$$

The proof of Claim A.4.4 is similar to that of Claim 3.3.4.

Theorem A.4.5. *Let $\mathcal{X}$ and $\mathcal{Y}$ be sets with $k = |\mathcal{Y}| < \infty$, let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$. If $\mathsf{LD}(\mathcal{H}) = \infty$ then $\mathsf{MTD}(\mathcal{H}) = \infty$.*

Proof of Theorem A.4.5. Let $f_k(d)$ be the largest number such that every class with Littlestone dimension d has multi-class threshold dimension at least $f_k(d)$. We show by induction on d that f_k satisfies the following recurrence relation:

$$f_k(d) \geq \begin{cases} 1 & d = 1 \\ 1 + f_k(\lceil d/2k \rceil - 1) & d > 1 \end{cases}.$$

In particular, this implies that $f_k(d) \xrightarrow{d \to \infty} \infty$, which implies the theorem.

The base case $d = \mathsf{LD}(\mathcal{H}) = 1$ is immediate. For the induction step, we assume the relation holds whenever $\mathsf{LD}(\mathcal{H}) \in [d-1]$, and prove that it holds for $\mathsf{LD}(\mathcal{H}) = d$. Let T be a Littlestone tree of depth d that is shattered by $\mathcal{H}$. Fix $h \in \mathcal{H}$. Then h is a k-coloring of the nodes of T. By Lemma A.3.3, there exists an h-monochromatic subtree $T' \subseteq T$ of depth at least $\lceil d/2k \rceil$. Let y be the color assigned by h to all nodes of T'. T' is shattered by $\mathcal{H}$, so there exists a child c of the root x of T' such that edge from x to c is labeled by some value $y' \neq y$. Let $\mathcal{H}' = \{g \in \mathcal{H} : \ g(x) = y'\}$. $\mathcal{H}'$ shatters the subtree rooted at c, so $\mathsf{LD}(\mathcal{H}') \geq \lceil d/2k \rceil - 1$. By the induction hypothesis, there exist $x_1, \ldots, x_s$ for $s = f_k(\lceil d/2k \rceil - 1)$ that are multi-class threshold shattered by functions $h_1, \ldots, h_s \in \mathcal{H}'$. By construction, the set $X = \{x_1, \ldots, x_s, x_{s+1} = x\}$ is multi-class threshold shattered by $\{h_1, \ldots, h_s, h_{s+1} = h\}$, because $h_{s+1}(x_j) = y$ for all $j \in [s+1]$, and $h_i(x_{s+1}) = y'$ for all $i \in [s]$. Hence, $f_k(d) \geq s + 1 = 1 + f_k(\lceil d/2k \rceil - 1)$, as desired. □

A.5 Proof of Agnostic Lower Bound

The lower bound in Theorem 3.6.1 is derived using an anti-concentration technique from Lemma 14 of Ben-David et al. (2009). Specifically, this technique uses the following inequality.

Theorem A.5.1 (Khintchine's inequality; Lemma 8.2 in Cesa-Bianchi and Lugosi, 2006)**.** *Let $k \in \mathbb{N}$, and let $\sigma_1, \sigma_2, \ldots, \sigma_k$ be random variables sampled independently and uniformly at random from $\{\pm 1\}$. Then*

$$\mathbb{E}\left[\left|\sum_{i\in k} \sigma_k\right|\right] \geq \sqrt{k/2}.$$

Proof of lower bound in Theorem 3.6.1. Let $d = \mathsf{VC}(\mathcal{H})$. Let $\{x_1^*, \ldots, x_d^*\} \subseteq \mathcal{X}$ be a set of cardinality d that is VC-shattered by $\mathcal{H}$. Let $k \in \mathbb{N}$ be the largest integer such that $kd \leq n$.

Let $x \in \mathcal{X}^n$ be a sequence consisting of k copies of the shattered set, namely,

$$(x_1, \ldots, x_{kd}) = \left(x_1^1, x_1^2, \ldots, x_1^k, x_2^1, x_2^2, \ldots, x_2^k, \ldots, x_d^1, x_d^2, \ldots, x_d^k\right),$$

such that $x_i^j = x_i^*$ for all $i \in [d]$ and $j \in [k]$. If $kd < n$ then the remaining $n - kd$ elements of x may be arbitrary.

Consider a randomized adversary that selects the sequence x, and chooses all labels to be i.i.d. uniform random bits. For each $i \in [d]$ and $j \in [k]$, let $y_i^j = y_{(i-1)k+j}$ and $\hat{y}_i^j = \hat{y}_{(i-1)k+j}$ denote, respectively, the labels and the predictions corresponding to x_i^j. Then for any learner strategy A,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{y\sim \mathrm{U}(\{0,1\}^n)}[R(A,\mathcal{H},x,y)] \\
&\quad= \mathop{\mathbb{E}}_{y\sim \mathrm{U}(\{0,1\}^n)}\left[\mathop{\mathbb{E}}_{\hat{y}}\left[\sum_{i\in[n]} \mathbb{1}\left(\hat{y}_t \neq y_t\right)\right] - \min_{h\in\mathcal{H}} \sum_{i\in[n]} \mathbb{1}\left(h(x_t) \neq y_t\right)\right] \\
&\quad= \frac{n}{2} - \mathop{\mathbb{E}}_{y\sim \mathrm{U}(\{0,1\}^n)}\left[\min_{h\in\mathcal{H}} \sum_{i\in[n]} \mathbb{1}\left(h(x_t) \neq y_t\right)\right] && (y_i \perp \hat{y}_i) \\
&\quad\geq \frac{kd}{2} - \mathop{\mathbb{E}}_{y\sim \mathrm{U}\left(\{0,1\}^{kd}\right)}\left[\min_{h\in\mathcal{H}} \sum_{i\in[d]}\sum_{j\in[k]} \mathbb{1}\left(h(x_i^j) \neq y_i^j\right)\right] && \text{(A.5)} \\
&\quad= \frac{kd}{2} - \mathop{\mathbb{E}}_{y\sim \mathrm{U}\left(\{0,1\}^{kd}\right)}\left[\sum_{i\in[d]}\min_{h\in\mathcal{H}} \sum_{j\in[k]} \mathbb{1}\left(h(x_i^j) \neq y_i^j\right)\right] && (\mathcal{H} \text{ shatters } \{x_1^*, \ldots, x_d^*\}) \\
&\quad= \sum_{i=1}^{d} \frac{k}{2} - \mathop{\mathbb{E}}_{y_i\sim \mathrm{U}\left(\{0,1\}^{k}\right)}\left[\min_{h\in\mathcal{H}} \sum_{j\in[k]} \mathbb{1}\left(h(x_i^j) \neq y_i^j\right)\right] \\
&\quad= \sum_{i=1}^{d} \frac{k}{2} - \mathop{\mathbb{E}}_{y_i\sim \mathrm{U}\left(\{0,1\}^{k}\right)}\left[\min\{r_i, k - r_i\}\right] && (\text{Let } r_i = \sum_{j\in[k]} y_i^j)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{d} \mathop{\mathbb{E}}_{y_i \sim \mathrm{U}\left(\{0,1\}^k\right)} \left[\left| \frac{k}{2} - r_i \right| \right] \\
&= \sum_{i=1}^{d} \mathop{\mathbb{E}}_{y_i \sim \mathrm{U}\left(\{0,1\}^k\right)} \left[\left| \frac{k}{2} - \sum_{j \in [k]} \left(\frac{1}{2} + \frac{\sigma_i^j}{2} \right) \right| \right] && \left(\text{Let } \sigma_i^j = \begin{cases} 1 & y_i^j = 1 \\ -1 & y_i^j = 0 \end{cases} \right) \\
&= \frac{1}{2} \sum_{i=1}^{d} \mathop{\mathbb{E}}_{y_i \sim \mathrm{U}\left(\{0,1\}^k\right)} \left[\left| \sum_{j \in [k]} \sigma_i^j \right| \right] \\
&\geq \frac{1}{2} \sum_{i=1}^{d} \sqrt{\frac{k}{2}} = \frac{d\sqrt{k}}{2\sqrt{2}} = \Omega\left(\sqrt{nd}\right), && \text{(A.6)}
\end{aligned}
$$

where the final inequality is Khintchine's inequality (Theorem A.5.1). To see that Inequality (A.5) holds, let $h^* \in \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{kd} \mathbb{1}\left(h(x_t) \neq y_t\right)$, and then

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{y \sim \mathrm{U}(\{0,1\}^n)} \left[\min_{h \in \mathcal{H}} \sum_{i \in [n]} \mathbb{1}\left(h(x_t) \neq y_t\right) \right] \\
&\quad \leq \mathop{\mathbb{E}}_{y \sim \mathrm{U}(\{0,1\}^n)} \left[\sum_{i=1}^{kd} \mathbb{1}\left(h^*(x_t) \neq y_t\right) \right] + \mathop{\mathbb{E}}_{y \sim \mathrm{U}(\{0,1\}^n)} \left[\sum_{i=kd+1}^{n} \mathbb{1}\left(h^*(x_t) \neq y_t\right) \right] \\
&\quad \leq \mathop{\mathbb{E}}_{y \sim \mathrm{U}(\{0,1\}^n)} \left[\sum_{i=1}^{kd} \mathbb{1}\left(h^*(x_t) \neq y_t\right) \right] + \frac{n - kd}{2}. && \left(\{y_t\}_{t > kd} \perp h^*\right)
\end{aligned}
$$

In particular, Eq. (A.6) implies that for every learner strategy A there exists $y \in \{0,1\}^n$ such that $R(A, \mathcal{H}, x, y) = \Omega\left(\sqrt{nd}\right)$. □

Appendix B

Appendices for Chapter 4

B.1 Types of Scientific Studies Amenable to PAC Verification

In Section 4.1, we suggested that PAC verification can be used to verify some types of experiments in a manner that is cheaper than a traditional replication. In this appendix we discuss three such types of experiments.

Before doing so, we would like to mention a possible objection that may be troubling the attentive reader: experimental findings typically assert that some specific hypothesis is true, or has some specified loss; for instance, that smoking cigarettes predicts lung cancer with some specified accuracy. Replication consists of verifying that the hypothesis indeed has a loss close to that stated in the original publication. But as we saw in Section 4.1, verifying that a specific hypothesis has a specified loss can be done with $O(\frac{1}{\varepsilon^2})$ independent samples, without using any special PAC verification techniques. In contrast, the strength of PAC verification lies in its ability to prove that the distance between some *class* of hypotheses and the unknown distribution is *large*, or alternatively, that "no better hypothesis exists" – but this appears unrelated to scientific replication. Nonetheless, we now explain how this CoNP-like flavor of PAC verification can indeed be very useful for replicating or verifying scientific publications.

Consider the following four types of scientific settings.

1. **Confounding variables.** Consider a publication that claims to have found a strong positive correlation between playing the accordion and developing a specific type of cancer. While this effect might be real, best practices would require that the study attempt to control for confounding variables. For instance, if people who play the accordion tend to be older, and older people tend to have more cancer, that could explain the correlation between accordions and cancer.

   Controlling for confounding variables is often performed by *binning* (also called *cross-tabulation*), which in the above example would mean dividing the participants into age groups, and checking whether the effect exists within each age group. Another common practice is to perform multiple regression, in which the "treatment" variable (playing the accordion) is used together with the potentially confounding variables (such as age) as the input variables that the regression model uses for predicting the response variable (having cancer). After performing the regression, the strength of the association between the response variable and each of the individual input variables is captured by the parameters corresponding to that variable in the regression model (e.g., in a linear regression, this would be the linear coefficient associated with the specific input variable).

This is a place where the CoNP-like flavor of PAC verification becomes useful. Regardless of the specific technique used, the notion of controlling for confounding variables is essentially this: the published result is purportedly "the best explanation" even after considering various other variables and the ways in which they might affect the response variable. Hence, verification of a study that controls for confounding variables can be viewed as verifying that the proposed hypothesis is the best within some class that includes alternative hypotheses that explicitly account for the effect of potential confounders. In the example above, one would need to PAC verify that predicting cancer is indeed best achieved by a hypothesis that places a lot of weight on playing the accordion, rather than some alternative hypothesis that attributes less weight to playing the accordion and more weight to age. In cases where there are many potentially confounding variables which might interact in various ways, the class of alternative hypotheses that must be ruled out can be large, and so PAC verification may be useful.

2. **Regression analysis.** Consider an empirical study that attempts to find a formula for predicting the value of a dependent variable $Y \in \mathbb{R}$ given the values of independent variables $X_1, \ldots, X_n \in \mathbb{R}$. The study can repeatedly measure the values of the dependent and independent variables in various cases, and then perform a regression analysis to identify the function $f(X_1, \ldots, X_n)$ that best predicts Y within some class of functions H (e.g., linear functions, low degree polynomials, etc.). In this setting, it is natural to perform PAC verification to ensure that the proposed hypothesis is indeed the best within the class of functions that the regression analysis considered.

3. **Multiple hypothesis testing.** In this setting, there is a finite class of hypothesis $H = \{h_1, ..., h_k\}$, and the researchers perform an experiment in order to decide for each hypothesis h_i whether it should be accepted or rejected. This scenario is common in many branches of science. As a concrete example, consider genome-wide association studies (GWAS), in which researchers compare genotypic information throughout the genome in large cohorts in order to identify genetic variants[1] that are associated with a certain phenotype of interest, such as a disease.[2] In a GWAS, we can think of each h_i as

[1] Such as single-nucleotide polymorphisms (SNPs).

[2] Buniello, MacArthur, Cerezo, Harris, Hayhurst, Malangone, McMahon, Morales, Mountjoy, Sollis, et al. (2019) is a catalog of over 70,000 different GWAS publications. Pe'er, Yelensky, Altshuler, and Daly (2008) and Palmer and Pe'er (2017) discuss statistical aspects of GWAS.

the hypothesis stating that genetic variant i is associated with the disease, and for each i, the study will either accept or reject hypothesis h_i. Huge efforts have been invested in the past two decades to ensure that GWAS publications can (and do) get replicated (Marigorta, Rodríguez, Gibson, and Navarro, 2018; Hirschhorn, Lohmueller, Byrne, and Hirschhorn, 2002).

We argue that scientific publications like GWAS that perform multiple hypothesis testing could potentially benefit from PAC verification protocols. As a loose illustration, consider a study that claims to have compiled a list containing the "100 genetic variants that are most associated with the disease." Formally, we can think of this as follows. For each genetic variant i there is some (unknown) real number $\alpha_i \in [0, 1]$ that represents the true amount of association between genetic variant i and the disease in the general population, where 1 indicates the strongest possible association and 0 indicates complete lack of association. For each i, we write $\tilde{h}_i = 1$ if the study included genetic variant i in the list, and $\tilde{h}_i = 0$ otherwise. We can now think of the list of the top 100 genetic variants proposed by the study as being represented by a vector $\tilde{h} = (\tilde{h}_1, \tilde{h}_2, \tilde{h}_3, \ldots, \tilde{h}_k) \in \{0, 1\}^k$, where k is the total number of genetic variants considered in the study, and $\tilde{h}$ has precisely 100 non-zero entries. We define the total loss of the study to be $L(\tilde{h}) = \sum_i (\alpha_i - \tilde{h}_i)^2$. The study is ε-good if $L(\tilde{h}) \leq \min_{h \in T} L(h) + \varepsilon$, where T is the set of all possible lists of length 100, namely $T = \{h \in \{0, 1\}^k : \ \|h\|_1 \leq 100\}$. The problem of verifying that the hypothesis $\tilde{h}$ is ε-good with respect to the class T is technically not an instance of PAC verification, but it is very similar to PAC verification.

4. **Negative results.** In the GWAS setting, consider a publication that claims "none of the genetic variants on chromosome j are associated with the disease." This claim falls squarely within the framework of PAC verification. To see this, let $\mathcal{X}$ be the set of possible genomes, and let the unknown distribution $\mathcal{D}$ provide samples of the form (x, y), where $x \in \mathcal{X}$ is the genome of a random person from the general population, and $y \in \{0, 1\}$ indicates whether that person has the disease. For each genetic variant i on the chromosome of interest, the class $\mathcal{H}$ contains a hypothesis h_i such $h_i(x) = 1$ if genome x contains genetic variant i, and $h_i(x) = 0$ otherwise. In addition, $\mathcal{H}$ contains the constant functions $c_0(x) \equiv 0$ and $c_1(x) \equiv 1$. The notion of verifying the negative claim in the publication is captured by PAC verifying that one of the constant functions is ε-good with respect to $\mathcal{H}$.

Whenever the number of samples necessary for PAC verification is lower than the number of samples used in the original publication, PAC verification becomes cheaper than a full replication of the study, but still provides the same benefits. PAC verification is most likely to be useful in settings where the researchers do not attempt to avoid errors completely, but rather are interested in balancing the risks of false positive and false negative errors via mechanism like controlling the false discovery rate (FDR; see Benjamini and Hochberg, 1995; Storey and Tibshirani, 2003). Note that in some cases, PAC verification might actually provide stronger evidence in favor of the publication than a traditional replication would, because it verifies the result in a manner that is qualitatively different from how the result was originally obtained.[3]

We hope that the first steps taken in this work may eventually lead to the development of practical PAC verification protocols that will be useful for the scientific community.

B.2 Proofs for Query Delegation Protocols

Simple Query Delegation

Assumptions:

- A is a 1-PAC learning ERM algorithm for $\mathcal{H}$ with sample complexity $m_{\mathcal{H}}(\varepsilon, \delta)$.
- $m' = m_{\mathcal{H}}^{\mathrm{UC}}(\varepsilon/4, \delta/2)$, where $m_{\mathcal{H}}^{\mathrm{UC}}$ denotes uniform convergence sample complexity of $\mathcal{H}$.
- $k = \left\lceil \frac{4\log(\frac{2}{\delta})}{\varepsilon} \right\rceil$

---

1. V takes i.i.d. labeled samples $(x_1, y_1), \ldots, (x_k, y_k)$ from $\mathcal{D}$.
2. V takes i.i.d. unlabeled samples $x_{k+1}, \ldots, x_{m'}$ from $\mathcal{D}_{\mathcal{X}}$.
3. V chooses a random permutation $\pi : [m'] \to [m']$, and sends $(x_{\pi(1)}, \ldots, x_{\pi(m')})$ to P.
4. P uses query access to $f_{\mathcal{D}}$ to obtain $\tilde{y}_{\pi(i)} = f_{\mathcal{D}}(x_{\pi(i)})$ for each $i \in [m']$, and sends $(\tilde{y}_{\pi(1)}, \ldots, \tilde{y}_{\pi(m')})$ to V.
5. V checks that $\tilde{y}_i = y_i$ for all $i \in [k]$. If this does not hold, V outputs $h = \text{reject}$. Otherwise, V executes A with precision parameter $\varepsilon/4$, confidence parameter $\delta/2$ and input sample $\tilde{S} = ((x_1, \tilde{y}_1), \ldots, (x_{m'}, \tilde{y}_{m'}))$, and then outputs the hypothesis h returned by A.

Protocol B.1: Simple Query Delegation

Proof for Claim 4.5.2. For the completeness, note that if P is honest, then V outputs $h = A((x_1, f_{\mathcal{D}}(x_1)), \ldots, (x_{m'}, f_{\mathcal{D}}(x_{m'})))$ such that all x_i are sampled i.i.d. from $\mathcal{D}_{\mathcal{X}}$. Because A is a 1-PAC learner, it holds that with probability at least $1 - \frac{\delta}{2}$, h is $\frac{\varepsilon}{4}$-good for $\mathcal{H}$ w.r.t. $\mathcal{D}$.

[3] This is reminiscent of the *little oh property* of Blum and Kannan (1995).

For soundness, we show that the following hold for any (possibly unbounded and malicious) prover:

(i) If V did not reject, then with probability at least $1 - \frac{\delta}{2}$, it holds that

$$\frac{|\{i \in [m] :\ \tilde{y}_i \neq f_{\mathcal{D}}(x_i)\}|}{m} \leq \frac{\varepsilon}{4}. \tag{B.1}$$

(ii) If (B.1) holds, then with probability at least $1 - \frac{\delta}{2}$, the hypothesis h returned by A is ε-good for $\mathcal{H}$ w.r.t. $\mathcal{D}$.

Together, these two conditions imply soundness, i.e.

$$\mathbb{P}\Big[h \neq \text{reject} \ \wedge \ \Big(L_{\mathcal{D}}(h) > L_{\mathcal{D}}(\mathcal{H}) + \varepsilon\Big)\Big] \leq \delta.$$

For (i), let $t = |\{i \in [m] :\ \tilde{y}_i \neq f_{\mathcal{D}}(x_i)\}|$. If $t > \varepsilon m/4$ then

$$\mathbb{P}_{\pi}[h \neq \text{reject}] \leq \left(1 - \frac{k}{m}\right)^{t} < e^{-kt/m} < e^{-\varepsilon k/4} \leq \frac{\delta}{2},$$

where the probability is over the choice of the permutation π, and the final inequality follows from our choice of $k \geq \frac{4\log(\frac{2}{\delta})}{\varepsilon}$.

For (ii), note that hypothesis h returned by A is an ERM hypothesis with respect to $\tilde{S}$. Let $S = ((x_1, f_{\mathcal{D}}(y_1), \ldots, (x_m, f_{\mathcal{D}}(y_m))$, and let h' be an ERM hypothesis with respect to S, and let h^* be any hypothesis in $\mathcal{H}$. Then

$$\begin{aligned}
L_{\mathcal{D}}(h) &\leq L_S(h) + \varepsilon/4 && \text{(uniform convergence of } \mathcal{H}\text{)}\\
&\leq L_{\tilde{S}}(h) + 2\varepsilon/4 && \text{(from B.1)}\\
&\leq L_{\tilde{S}}(h') + 2\varepsilon/4 && (h \text{ is an ERM with respect to } \tilde{S})\\
&\leq L_S(h') + 3\varepsilon/4 && \text{(from B.1)}\\
&\leq L_S(h^*) + 3\varepsilon/4 && (h' \text{ is an ERM with respect to } S)\\
&\leq L_{\mathcal{D}}(h^*) + \varepsilon. && \text{(uniform convergence of } \mathcal{H}\text{)}
\end{aligned}$$

Hence, $L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(h^*) + \varepsilon$ for all $h^* \in \mathcal{H}$, as desired.

For the query complexity, note that P makes m' queries, and from Theorem 4.1.15, $m' = O(m(\varepsilon, \delta))$. □

Compressed Query Delegation

We only prove Claim 4.5.3 for the special case in which $\mathcal{X} = \{0,1\}^n$, and $\mathcal{D}_{\mathcal{X}}$ is the uniform distribution. This implies the claim for other domains and distributions, because uniformly random bits can be used to simulate samples from other (efficiently samplable) distributions.

The PAC verification protocol uses the following compression protocol as a subroutine. Assume V takes a labeled sample $(x, y) \sim \mathcal{D}$. The compression protocol enables V to send a (randomized) message to P such that:

(i) The length of the message is roughly $|x|$ plus some constant.

(ii) The message specifies a sequence of $t+1$ unlabeled samples $x_0, \dots, x_t \in \mathcal{X}$.

(iii) The message contains x, so that $x = x_{i^*}$ for some $i^* \in \{0, 1, 2, \dots, t\}$.

(iv) P does not know i^*, that is, i^* is uniformly random and independent of the message that P received.

This compression protocol uses a pseudorandom generator f_{PRG} of the following form. $f_{\mathrm{PRG}}(s)$ is a deterministic function that takes a seed $s \in \{0,1\}^{\ell}$ and returns a sequence $x_1, \dots, x_t \in \mathcal{X}$ for fixed $\ell, t \in \mathbb{N}$. We assume that f_{PRG} is pseudorandom with respect to the learning algorithm A in the sense that A successfully 1-PAC learns $\mathcal{H}$ with respect to the uniform distribution over $\mathcal{X}$ if it receives a labeled sample $(x_1, y_1), \dots, (x_m, y_m)$ in which the x_i's were chosen according to a certain procedure that uses f_{PRG}, rather than being sampled i.i.d. from the uniform distribution.[4]

[4] Technically, the x_i's are sampled by repeatedly invoking Protocol B.2, as is done is Protocol B.3.

Assumptions:

- f_{PRG} is a pseudorandom generator with seed size ℓ and stretch t as above.
- $X \in \mathcal{X}$.

```
GenerateCompressedMessage($X$):
  Take the following samples independently:
    – $I^* \sim \text{Uniform}(\{0, 1, 2, \ldots, t\})$
    – $S \sim \text{Uniform}(\{0, 1\}^\ell)$
  $W_1, \ldots, W_t \leftarrow f_{\text{PRG}}(s)$
  $X_0 \leftarrow X \oplus \left(\bigoplus_{1 \leq j \leq I^*} W_j\right)$      ▷ "$\oplus$" denotes bitwise XOR; $\oplus$ of an empty set is 0.
  $M \leftarrow (X_0, S)$
  Output $(M, I^*)$

ExpandCompressedMessage($M$):
  $(X_0, S) \leftarrow M$
  $W_1, \ldots, W_t \leftarrow f_{\text{PRG}}(s)$
  for $i \in [t]$
      $X_i \leftarrow X_{i-1} \oplus W_i$
  Output $X_0, \ldots, X_t$
```

Protocol B.2: A compression protocol

The compressed query delegation protocol operates as follows, using the above protocol as a subroutine.

Assumptions:

- A is a 1-PAC learning ERM algorithm for $\mathcal{H}$ with sample complexity $m_{\mathcal{H}}(\varepsilon, \delta)$.
- $m' = m_{\mathcal{H}}^{\mathrm{UC}}(\varepsilon/4, \delta/2)$, where $m_{\mathcal{H}}^{\mathrm{UC}}$ denotes uniform convergence sample complexity of $\mathcal{H}$.
- $k = \left\lceil \frac{4\log(\frac{2}{\delta})}{\varepsilon} \right\rceil$
- The stretch of the pseudorandom generator f_{PRG} is $t = \lfloor m'/k \rfloor$

```
1. $V$ performs the following:
       for $j \in [k]$
           Sample $(X_j, Y_j) \sim \mathcal{D}$
           $M_j, I_j^* \leftarrow \textsc{GenerateCompressedMessage}(X_j)$
       Send $(M_1, \ldots, M_k)$ to $P$

2. $P$ performs the following:
       $\tilde{Y} \leftarrow$ new matrix
       for $j \in [k]$
           $X_0, \ldots, X_t \leftarrow \textsc{ExpandCompressedMessage}(M_j)$
           for $i \in \{0, 1, \ldots, t\}$
               $\tilde{Y}_{j,i} \leftarrow f_{\mathcal{D}}(X_i)$
       Send $\tilde{Y}$ to $V$

3. $V$ performs the following:
       $X \leftarrow$ new matrix
       for $j \in [k]$
           $i^* \leftarrow I_j^*$
           if $\tilde{Y}_{j,i^*} \neq Y_j$
               Output 'reject' and halt
           $(X_{j,0}, \ldots, X_{j,t}) \leftarrow \textsc{ExpandCompressedMessage}(M_j)$
       $\mathit{Sample} \leftarrow \{(X_{j,i}, \tilde{Y}_{j,i}) : \; j \in [k], i \in \{0, 1, \ldots, t\}\}$
       $h \leftarrow A(\mathit{Sample}, \varepsilon/4, \delta/2)$
       Output $h$
```

Protocol B.3: Compressed Query Delegation.

Claim B.2.1. *Assume X is sampled uniformly from $\mathcal{X}$, and then the subroutine* Generate CompressedMessage(X) *is executed and outputs the tuple (M, I^*). Then the random variables M and I^* satisfy that $M \perp I^*$.*

Proof. Let $(X_0, S) = M$. Fix $x_0 \in \mathcal{X}$, $s \in \{0,1\}^\ell$ and $i \in \{0, 1, 2, \ldots, t\}$. Observe that

$$\begin{aligned}\mathbb{P}[X_0 = x_0 |\ S = s \ \wedge\ I^* = i] &= \mathbb{P}\left[X \oplus \left(\bigoplus_{1 \le j \le I^*} W_j\right) = x_0 \ \middle|\ S = s \ \wedge\ I^* = i\right] \\ &= \mathbb{P}\left[X = x_0 \oplus \left(\bigoplus_{1 \le j \le I^*} W_j\right) \ \middle|\ S = s \ \wedge\ I^* = i\right] \\ &= \frac{1}{|\mathcal{X}|}.\end{aligned}$$

Hence,

$$\begin{aligned}\mathbb{P}[X_0 = x_0 \ \wedge\ S = s \ \wedge\ I^* = i] &= \mathbb{P}[I^* = i] \cdot \mathbb{P}[S = s \mid I^* = i] \cdot \mathbb{P}[X_0 = x_0 \mid S = s \ \wedge\ I^* = i] \\ &= \mathbb{P}[I^* = i] \cdot \mathbb{P}[S = s] \cdot \mathbb{P}[X_0 = x_0 \mid S = s \ \wedge\ I^* = i] \\ & \qquad (S \perp I^*) \\ &= \mathbb{P}[I^* = i] \cdot \mathbb{P}[S = s] \cdot \frac{1}{|\mathcal{X}|} \\ &= \mathbb{P}[I^* = i] \cdot \mathbb{P}[X_0 = x_0 \ \wedge\ S = s]. \qquad (X_0 \perp S)\end{aligned}$$

□

Proof of Claim 4.5.3. The proof is similar to the proof of Claim 4.5.2. For completeness, notice that if P is honest, then V never rejects, and outputs a hypothesis h returned by A on a sample where the x's were generated using f_{PRG}, and the labels are all correct. Because A is a 1-PAC learner, and f_{PRG} is pseudorandom with respect to A, it holds that h is ε-good with probability at least $1 - \delta$.

Soundness follows from (i) and (ii) in the same manner as in the proof of Claim 4.5.2. Notice that (ii) holds in the current case by the same argument as in that proof. To complete the proof we need to establish (i). Let b_j be the number of dishonest labels that P provided for X's generated by message M_j, and let $b = \sum_{j \in [k]} b_j$ be the total number of dishonest labels provided by P. We need to show that if $\frac{b}{m''} > \frac{\varepsilon}{4}$, then V rejects with probability at least $1 - \frac{\delta}{2}$, where $m'' = (t+1)k \ge m'$ is the total number of samples.

For each $j \in [k]$, V knows the correct label for X_{j,i^*} such that $i^* = I_j^*$. From Claim B.2.1, I_j^* is independent of M_j. Because I_j^* is uniformly random, the chance that V does not detect a dishonest label for message M_j is $p_j = \left(1 - \frac{b_j}{t+1}\right)$. In total, if $\frac{b}{m''} > \frac{\varepsilon}{4}$ then

$$\begin{aligned}\mathbb{P}[h \neq \text{reject}] = \prod_{j \in [k]} p_j = \left(\sqrt[k]{\prod_{j \in [k]} p_j}\right)^k \leq \left(\frac{1}{k} \sum_{j \in [k]} p_j\right)^k = \left(1 - \frac{b}{m''}\right)^k \\ < \left(1 - \frac{\varepsilon}{4}\right)^k < e^{-\frac{\varepsilon}{4}k} \leq \frac{\delta}{2},\end{aligned}$$

where the probability is over the choice of the indices I_j^*, and the first inequality is the AM-GM inequality, and the final inequality follows from our choice of k. □

Noninteractive Query Delegation

Assumptions:

- A is a 1-PAC learning ERM algorithm for $\mathcal{H}$ with sample complexity $m_{\mathcal{H}}(\varepsilon, \delta)$.
- $m' = m_{\mathcal{H}}^{\mathrm{UC}}(\varepsilon/4, \delta/2)$, where $m_{\mathcal{H}}^{\mathrm{UC}}$ denotes uniform convergence sample complexity of $\mathcal{H}$.
- $k = \left\lceil \frac{4\log(\frac{2}{\delta})}{\varepsilon} \right\rceil$
- $f_{\mathrm{CRS}}(t)$ is a source of common randomness that provides the same i.i.d. samples $x_1, \ldots, x_t$ from $\mathcal{D}_{\mathcal{X}}$ to all parties.

```
1. $P$ performs the following:
      $X_1, \ldots, X_{m'} \leftarrow f_{\mathrm{CRS}}(m')$
      for $i \in [m']$
          $\tilde{Y}_i \leftarrow f_{\mathcal{D}}(X_i)$
      Publish $(\tilde{Y}_1, \ldots, \tilde{Y}_{m'})$

2. $V$ performs the following:
      $X_1, \ldots, X_{m'} \leftarrow f_{\mathrm{CRS}}(m')$
      Sample $\{I_1, \ldots, I_k\} \leftarrow \mathrm{Uniform}\left(\binom{[m']}{k}\right)$
      for $i \in \{I_1, \ldots, I_k\}$
          if $\tilde{Y}_i \neq f_{\mathcal{D}}(X_i)$
              Output 'reject' and halt
      $h \leftarrow A(\{(X_i, \tilde{Y}_i) : \; i \in [m']\}, \varepsilon/4, \delta/2)$
      Output $h$
```

Protocol B.4: Noninteractive Query Delegation.

The proof of Claim 4.5.4 is similar to that of Claim 4.5.2. Note that the amount of common randomness required can be reduced substantially by using an appropriate PRG, as in Claim 4.5.3, while the security remains information-theoretic.

B.3 Thresholds Over Discrete Sets

In Section 4.3 we presented the class $\mathcal{T}$ of thresholds over the interval $[0, 1] \subseteq \mathbb{R}$, and neglected issues pertaining to the representation of real numbers. Here, we outline how similar results can be obtained for the class of threshold over a finite set $\mathcal{X} \subseteq [0,1]$. We write $\mathcal{T}^{\mathcal{X}} = \{f_t\}_{t \in \mathcal{X}} \subseteq \mathcal{T}$, and are interested in 2-PAC verification of $\mathcal{T}^{\mathcal{X}}$ with respect to any distribution $\mathcal{D} \in \Delta(\mathcal{X} \times \{0,1\})$.[5]

This boils down to the following. Recall that when constructing certificates of loss for $\mathcal{T}$, we used the following primitive in the proof of Claim 4.3.5:

Fact B.3.1. *Let $[\alpha, \beta] \subseteq \mathbb{R}$ be an interval, and let p be a distribution over $\mathbb{R}$ that is absolutely continuous with respect to the Lebesgue measure. If $p\Big([\alpha,\beta]\Big) > r \geq 0$, then there exists $\gamma \in [\alpha, \beta]$ such that $p\Big([\alpha,\gamma]\Big) = r$.*

The following alternative primitive, which has the additional property that $\gamma \in \mathcal{X}$, will be used instead when producing certificates for $\mathcal{T}^{\mathcal{X}}$ that have succinct representations.

Claim B.3.2. *Let $N \in \mathbb{N}$, let $[\alpha, \beta] \subseteq \mathbb{R}$ be an interval with $\alpha, \beta \in \mathcal{X}$, and let p be a probability mass function over $\mathcal{X}$. If $p\Big([\alpha,\beta]\Big) > r \geq 0$, then there exists a pair (γ, q) where $\gamma \in \mathcal{X} \cap [\alpha, \beta]$ and $q \in [N]$, such that:*

$$\left| p\Big([\alpha,\gamma)\Big) + \frac{q}{N} \cdot p(\gamma) - r \right| \leq \frac{1}{2N}.$$

Likewise, there exists (γ', q') such that

$$\left| p\Big((\gamma',\beta]\Big) + \frac{q'}{N} \cdot p(\gamma') - r \right| \leq \frac{1}{2N}.$$

Proof. Take

$$\gamma = \min\left\{ x \in \mathcal{X} :\ p\Big([\alpha, x]\Big) \geq r \right\}$$

and

$$q = \operatorname*{arg\,min}_{i \in [N]} \left| \frac{i}{N} - \frac{r - p\Big([\alpha,\gamma)\Big)}{p(\gamma)} \right|.$$

[5] That is, any probability space $(\Omega, \mathcal{D}, \Sigma)$ with sample space $\Omega = \mathcal{X} \times \{0,1\}$, probability mass function $\mathcal{D}$, and σ-algebra $\Sigma = 2^{\Omega}$.

Observe that

$$\begin{aligned}\left|p\Big([\alpha,\gamma)\Big)+\frac{q}{N}\cdot p(\gamma)-r\right| &\leq \left|p\Big([\alpha,\gamma)\Big)+\frac{r-p\Big([\alpha,\gamma)\Big)}{p(\gamma)}\cdot p(\gamma)-r\right| \\ &\quad + p(\gamma)\left|\frac{r-p\Big([\alpha,\gamma)\Big)}{p(\gamma)}-\frac{q}{N}\right| \\ &= \left|\frac{r-p\Big([\alpha,\gamma)\Big)}{p(\gamma)}-\frac{q}{N}\right| \leq \frac{1}{2N}.\end{aligned}$$

The proof for (γ', q') is similar. □

Recall that a 0-valid certificate of loss ℓ for $\mathcal{T}$ with respect to distribution $\mathcal{D}$ was a pair (a,b) such that $\mathcal{D}^1\Big([0,a)\Big)=\mathcal{D}^0\Big([b,1]\Big)=\ell$, where $\mathcal{D}^i(X):=\mathcal{D}(X\times\{i\})$. For the discrete case, we use the following definition of a *certificate with finite resolution.*

Definition B.3.3. *Fix $N\in\mathbb{N}$, and let $\mathcal{X}\subseteq[0,1]$ be a finite set. Let $\mathcal{D}\in\Delta(\mathcal{X}\times\{0,1\})$ be a distribution and $\ell,\eta\geq 0$. A certificate of loss at least ℓ for class $\mathcal{T}^{\mathcal{X}}$ with resolution $\frac{1}{N}$ is a tuple*

$$(a,q_a,b,q_b)$$

where $0<a\leq b<1$ and $q_a,q_b\in[N]$, and if $a=b$ then $q_a+q_b\leq N$.

We say that the certificate is η-valid with respect to distribution $\mathcal{D}$ if

$$\left|\mathcal{D}^1\Big([0,a)\Big)+\frac{q_a}{N}\cdot p(a)-\ell\right|+\left|\mathcal{D}^0\Big((b,1]\Big)+\frac{q_b}{N}\cdot p(b)-\ell\right|\leq\eta.$$

Using Claim B.3.2, one can repeat the proof of Claim 4.3.5 to show the following.

Claim B.3.4. *Fix $N\in\mathbb{N}$, and let $\mathcal{X}\subseteq[0,1]$ be a finite set. Let $\mathcal{D}\in\Delta(\mathcal{X}\times\{0,1\})$ be a distribution and $\ell\geq 0$. If $L_{\mathcal{D}}(\mathcal{T}^{\mathcal{X}})=\ell$, then there exist (a,q_a,b,q_b) such that $a,b\in\mathcal{X}$ and $q_a,q_b\in[N]$, which constitute a certificate of loss $\frac{\ell}{2}$ for the class $\mathcal{T}^{\mathcal{X}}$ that is $\frac{1}{N}$-valid with respect to $\mathcal{D}$.*

In particular, one can obtain an η-valid certificate of finite precision by choosing the precision parameter N to satisfy $N\geq\frac{1}{\eta}$. Likewise, it is possible to repeat the rest of the analysis, and show that an η-valid certificate of loss ℓ entails that $L_{\mathcal{D}}(\mathcal{T}^{\mathcal{X}})\geq\ell-\eta$, and that certificates can be generated and verified efficiently. Finally, we can generalize these results to a multi-threshold class $\mathcal{T}_d^{\mathcal{X}}$, and obtain that $\mathcal{T}_d^{\mathcal{X}}$ is 2-PAC verifiable, and exhibits a quadratic gap in sample complexity between learning and verification, as in Theorem 4.3.8.

B.4 Uniform Convergence for Set Systems

The following theorem is due to Vapnik and Chervonenkis (1968, 1971). See also the exposition by Alon and Spencer (2000, Theorem 13.4.4).

Definition B.4.1. *A set system is a tuple $(X, \mathcal{S})$, where X is any set, and $\mathcal{S} \subseteq 2^X$ is any collection of subsets of X. The members of X are called points.*

The VC dimension of a set system $(X, \mathcal{S})$ is the VC dimension of the set of indicator functions $\{\mathbb{1}_S : \ S \in \mathcal{S}\}$ as defined in Definition 4.1.14.

Definition B.4.2. *Let $(X, \mathcal{S})$ be a set system, let $\mathcal{D}$ be a distribution over X, and let $\varepsilon \in (0, 1)$. We say that a multiset $A \subseteq X$ is an ε-sample with respect to $\mathcal{D}$ if*

$$\forall S \in \mathcal{S} : \quad \left| \frac{|A \cap S|}{|A|} - \mathcal{D}(S) \right| \leq \varepsilon.$$

Theorem B.4.3. *There exists a constant $c > 0$ such that for any set system $(X, \mathcal{S})$ of VC-dimension at most d and any $0 < \epsilon, \delta < \frac{1}{2}$, a sequence of at least*

$$\frac{c}{\epsilon^2} \left(d \log \frac{d}{\epsilon} + \log \frac{1}{\delta} \right)$$

i.i.d. samples from $\mathcal{D}$ will be an ϵ-sample with respect to $\mathcal{D}$ with probability at least $1 - \delta$.

B.5 Identity Testing for Distributions

The following theorem is due to Batu, Fischer, Fortnow, Kumar, Rubinfeld, and White (2001, Theorem 24). See also Theorem 3.2.7 in Canonne (2020).

Theorem B.5.1. *Let $\mathcal{D}^* = (d_1, \ldots, d_n)$ be a distribution over a finite set of size n, and let $\varepsilon \in (0, 1)$. There exists an algorithm which, given the full specification of D^* and sample access to an unknown distribution D, takes*

$$O\left(\frac{\sqrt{n}}{\varepsilon^6} \log(n) \right)$$

samples from D, and satisfies:

- *Completeness. If*

$$d_{\mathsf{TV}}\left(D, D^*\right) \leq \frac{\varepsilon^3}{300\sqrt{n}\log n},$$

then the algorithm accepts with probability at least $\frac{2}{3}$.

- *Soundness. If*
$$d_{\mathsf{TV}}\left(D, D^*\right) > \varepsilon,$$
then the algorithm rejects with probability at least $\frac{2}{3}$.

A standard amplification argument yields the following:

Corollary B.5.2. *Taking*
$$O\left(\log\left(\frac{1}{\delta}\right)\frac{\sqrt{n}}{\varepsilon^6}\log(n)\right)$$
samples is sufficient to ensure completeness and soundness at least $1-\delta$ *(instead of* $\frac{2}{3}$*).*

B.6 Total Variation Distance

Claim B.6.1. *Let* $\delta \in (0,1)$, $\mathcal{X} := [n]$. *Consider a sequence* $x_1, x_2 \ldots, x_t$ *of i.i.d. samples taken from* $\mathcal{U}_{\mathcal{X}}$, *and let* G *denote the event in which all the samples are distinct, that is* $|\{x_1, \ldots, x_t\}| = t$. *Then taking*
$$n \geq \frac{\log(2e)}{\log\left(\frac{1}{1-\delta}\right)} \cdot t^2$$
entails that
$$\mathbb{P}[G] \geq 1 - \delta.$$

Claim B.6.2. *Let* $\mathbb{P}, \mathbb{Q}$ *be probability functions over a probability space* $(\Omega, \mathcal{F})$. *Then for all* $\alpha \in [0,1]$,
$$\mathsf{TV}((1-\alpha)\mathbb{P} + \alpha\mathbb{Q}, \mathbb{P}) \leq \alpha.$$
In particular, if X *is a random variable and* E *is an event, then*
$$\mathsf{TV}(X, X|E) \leq 1 - \mathbb{P}[E] = \mathbb{P}\left[\overline{E}\right].$$

Proof.
$$\begin{aligned}\mathsf{TV}((1-\alpha)\mathbb{P} + \alpha\mathbb{Q}, \mathbb{P}) &= \max_{A\in\mathcal{F}}\ (1-\alpha)\mathbb{P}(A) + \alpha\mathbb{Q}(A) - \mathbb{P}(A) \\ &= \max_{A\in\mathcal{F}}\ \alpha\cdot(\mathbb{Q}(A) - \mathbb{P}(A)) \leq \alpha.\end{aligned}$$

In particular, if $\mathbb{P}_X, \mathbb{P}_{X|E}$ denote the distributions of X and $X|E$ then
$$\mathsf{TV}\Big(\mathbb{P}_X, \mathbb{P}_{X|E}\Big) = \mathsf{TV}\Big((1 - \mathbb{P}\big[\overline{E}\big]) \cdot \mathbb{P}_{X|E} + \mathbb{P}\big[\overline{E}\big] \cdot \mathbb{P}_{X|\overline{E}}, \mathbb{P}_{X|E}\Big) \leq \mathbb{P}\big[\overline{E}\big].$$

□

B.7 Learning Fourier-Sparse Functions By Estimating Heavy Coefficients

Let $\mathcal{H}$ be the set of t-sparse functions $\{0,1\}^n \to \mathbb{R}$. In this appendix we recall the proof that one can PAC learn $\mathcal{H}$ with respect to $\mathbb{D}_{\mathcal{U}}^{\mathrm{func}}(\{0,1\}^n)$ by estimating heavy Fourier coefficients. We stress that this is a well-known result and is included for completeness only (see Mansour, 1994).

Claim B.7.1. *Let $\varepsilon > 0$. Let $\mathcal{D} \in \mathbb{D}_{\mathcal{U}}^{\mathrm{func}}(\{0,1\}^n)$ have target function $f:\ \{0,1\}^n \to \{1,-1\}$. Consider the function*

$$h(x) = \sum_{T \in L} \alpha_T \chi_T(x),$$

where L is a set such that $\widehat{f}^{\geq \tau} \subseteq L$ for $\tau = \frac{\varepsilon}{4t}$. If

$$\forall T \in L:\ |\alpha_T - \widehat{f}(L)| \leq \sqrt{\frac{\varepsilon}{8|L|}},$$

then $L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(\mathcal{H}) + \varepsilon$.

Before proving this claim, we show that if a function f is close to being sparse, then it can be approximated by a sparse function g that includes only coefficients where f has high Fourier weight.

Claim B.7.2. *Let $t \in \mathbb{N}$, let $\beta, \ell \in (0,1)$, and let $\mathcal{D} \in \mathbb{D}_{\mathcal{U}}^{\mathrm{func}}(\{0,1\}^n)$ have target function $f:\ \{0,1\}^n \to \{1,-1\}$. Assume $L_{\mathcal{D}}(\mathcal{H}) \leq \ell$. Then exists $g \in \mathcal{H}$ such that*

$$L_{\mathcal{D}}(g) \leq (1+\beta)\cdot \ell,$$

and $\widehat{g}^{>0} = \{T :\ |\widehat{g}(T)| > 0\} \subseteq \widehat{f}^{\geq \tau}$ with $\tau := \sqrt{\frac{\beta \cdot \ell}{t}}$.

Proof. Because $L_{\mathcal{D}}(\mathcal{H}) \leq \ell$, there exists a function $w \in \mathcal{H}$ such that $L_{\mathcal{D}}(w) \leq \ell$. Let $\widehat{w}^{>0} = \{T :\ |\widehat{w}(T)| > 0\}$. Consider the function

$$g(x) = \sum_{T \in \left(\widehat{w}^{>0} \cap \widehat{f}^{\geq \tau}\right)} \widehat{f}(T)\chi_T(x).$$

Clearly, g is t-sparse (because w is t-sparse), and $\widehat{g}^{>0} \subseteq \widehat{f}^{\geq\tau}$. Furthermore, we have

$$\begin{aligned} L_{\mathcal{D}}(g) &= \mathbb{E}_{x\in\{0,1\}^n}\Big[(f(x)-g(x))^2\Big] \\ &= \sum_{T\subseteq[n]} \Big(\widehat{f}(T)-\widehat{g}(T)\Big)^2 && \text{(Parseval's identity)} \\ &= \sum_{T\notin\left(\widehat{w}^{>0}\cap\widehat{f}^{\geq\tau}\right)} \Big(\widehat{f}(T)-\widehat{g}(T)\Big)^2 \\ &= \sum_{T\notin\widehat{w}^{>0}} \widehat{f}^2(T) + \sum_{T\in\widehat{w}^{>0}\setminus\widehat{f}^{\geq\tau}} \widehat{f}^2(T). \end{aligned}$$

We bound each sum separately.

$$\sum_{T\notin\widehat{w}^{>0}} \widehat{f}^2(T) = \sum_{T\notin\widehat{w}^{>0}} \Big(\widehat{f}(T)-\widehat{w}(T)\Big)^2 \leq \sum_{T\subseteq[n]} \Big(\widehat{f}(T)-\widehat{w}(T)\Big)^2 = L_{\mathcal{D}}(w) \leq \ell,$$

and

$$\sum_{T\in\widehat{w}^{>0}\setminus\widehat{f}^{\geq\tau}} \widehat{f}^2(T) \leq |\widehat{w}^{>0}|\cdot\tau^2 \leq t\cdot\frac{\beta\ell}{t} = \beta\ell.$$

□

Proof of Claim B.7.1. Observe that

$$L_{\mathcal{D}}(h) = \mathbb{E}\Big[(f(x)-h(x))^2\Big] = \sum_{T\in L}\Big(\widehat{f}(T)-\widehat{h}(T)\Big)^2 + \sum_{T\notin L}\widehat{f}^2(T),$$

and the first sum is bounded by

$$\sum_{T\in L}\Big(\widehat{f}(T)-\widehat{h}(T)\Big)^2 \leq |L|\cdot\frac{\varepsilon}{2|L|} = \frac{\varepsilon}{2}.$$

Therefore, to complete the proof it suffices to show that $\sum_{T\notin L}\widehat{f}^2(T) \leq L_{\mathcal{D}}(\mathcal{H}) + \frac{\varepsilon}{2}$. Invoking Claim B.7.2 with $\beta := \frac{\varepsilon}{2}$ and $\ell := \max\{L_{\mathcal{D}}(\mathcal{H}), \frac{\varepsilon}{4}\}$, there exists a t-sparse function $g:\ \{0,1\}^n \to \mathbb{R}$ such that

$$L_{\mathcal{D}}(g) \leq (1+\beta)\ell \leq L_{\mathcal{D}}(\mathcal{H}) + \frac{\varepsilon}{2},$$

and $\widehat{g}^{>0} = \{T:\ |\widehat{g}(T)|>0\} \subseteq \widehat{f}^{\geq\tau}$ with $\tau := \sqrt{\frac{\varepsilon\ell}{2t}} \geq \frac{\varepsilon}{4t}$. This entails that

$$\begin{aligned} \sum_{T\notin L}\widehat{f}^2(T) &\leq \sum_{T\in\widehat{f}^{<\tau}}\widehat{f}^2(T) \\ &\leq \sum_{T\subseteq[n]}\Big(\widehat{f}(T)-\widehat{g}(T)\Big)^2 \\ &= \mathbb{E}\Big[(f(x)-g(x))^2\Big] \leq L_{\mathcal{D}}(\mathcal{H}) + \frac{\varepsilon}{2}. \end{aligned}$$

□

B.8 Random Matrices Have Full Rank

Claim B.8.1. *Let $\tau > 0$, $n \in \mathbb{N}$. If $\tau \geq 2^{-\frac{n}{10}}$ then*

$$\frac{n}{\tau^4 2^n} \leq \frac{1}{128 \log\left(\frac{n}{\tau^4}\right)}$$

for n large enough.

Proof.

$$\tau \geq 2^{-0.1n} \implies \tau^8 \geq 2^{-0.8n} \geq \frac{128n\log(n)}{2^n} \implies \frac{2^n\tau^8}{n} \geq 128\log(n)$$
$$\implies \frac{2^n\tau^4}{n} \geq \frac{1}{\tau^4}128\log(n) \geq 128\log\left(\frac{n}{\tau^4}\right).$$

□

Claim B.8.2. *Let $n, m \in \mathbb{N}$, $\tau \geq 2^{-\frac{n}{10}}$, $m \leq \log\left(\frac{32n}{\tau^4}\right)$. Let $X = \{x_1, \ldots, x_m\}$ be a set of m vectors chosen independently and uniformly from $(\mathbb{F}_2)^n$. Then with probability at least $\frac{3}{4}$, the set X is linearly independent for n large enough.*

Proof. Think of the vectors as being chosen one by one. The probability that the first vector is non-zero is

$$\frac{2^n - 1}{2^n},$$

because we can choose any vector except 0. The probability that vector x_{k+1} is linearly independent of the first k vectors is

$$\frac{2^n - 2^k}{2^n},$$

because we can choose any vector not in $\text{span}(\{x_1, \ldots, x_k\})$. Because the choices are made independently, the probability that all m vectors are linearly independent is

$$\frac{2^n - 2^0}{2^n} \cdot \frac{2^n - 2^1}{2^n} \cdots \frac{2^n - 2^{m-1}}{2^n} \geq \left(\frac{2^n - 2^m}{2^n}\right)^m$$
$$\geq \left(\frac{2^n - \frac{32n}{\tau^4}}{2^n}\right)^m = \left(1 - \frac{\frac{32n}{\tau^4}}{2^n}\right)^m = \left(1 - \frac{1}{4\log\left(\frac{n}{\tau^4}\right)}\right)^{\log\left(\frac{n}{\tau^4}\right)} \geq 1 - \frac{1}{4},$$

where the last inequality is Bernoulli's inequality. □

Appendix C

Appendices for Chapter 5

C.1 Protocol for Unions of Intervals

The verification protocol for unions of d-intervals is detailed in Protocol C.1.

C.2 Verification of Statistical Query Algorithms

Definitions

Statistical Query Algorithms

Definition C.2.1 (Kearns, 1998)**.** *Let Ω be a set, let $\mathcal{D} \in \Delta(\Omega)$ be a distribution, and let $\tau \geq 0$. A statistical query is an indicator function $q : \Omega \to \{0,1\}$. An oracle $\mathcal{O}$ is a statistical query oracle for $\mathcal{D}$ with precision τ, denoted $\mathcal{O} \in \mathsf{SQ}(\mathcal{D},\tau)$, if at each invocation, $\mathcal{O}$ takes a statistical query q as input and produces an arbitrary evaluation $\mathcal{O}(q) \in [0,1]$ as output such that*

$$\Big|\mathcal{O}(q) - \mathbb{E}_{X\sim\mathcal{D}}[q(X)]\Big| \leq \tau. \tag{C.1}$$

In particular, the oracle's evaluations may be adversarial and adaptive, as long as each of them satisfies Eq. (C.1)*.*

Remark C.2.2. *The notion of PAC verification of an algorithm (Definition 5.2.3) requires that the verifier's output be competitive with $L_{\mathcal{D}}(A) = \mathbb{E}\Big[L_{\mathcal{D}}\Big(A^{\mathcal{O}}\Big)\Big]$, the expected loss of algorithm A when executed with access to oracle $\mathcal{O}$. For this expectation to be defined, throughout this chapter we only consider oracles whose behavior can be described by a probability measure. In particular, oracles may be adaptive and adversarial in a deterministic or randomized manner, but they cannot be arbitrary.*

Definition C.2.3. *A statistical query algorithm is a (possibly randomized) algorithm A that takes no inputs and has access to a statistical query oracle $\mathcal{O}$. At each time step $t = 1,2,3,\dots$:*

- *A chooses a finite batch $\mathbf{q}_t = (q_t^1, \dots, q_t^{n_t})$ of statistical queries and sends it to the oracle $\mathcal{O}$.*
- *$\mathcal{O}$ sends a batch of evaluations $\mathbf{v}_t = (v_t^1, \dots, v_t^{n_t}) \in [0,1]^{n_t}$ to A, such that $v_t^i = \mathcal{O}(q_t^i)$ for all $i \in [n_t]$.*
- *A either produces an output and terminates, or continues to time step $t+1$.*

The resulting sequence $\mathbf{r} = (\mathbf{q}_1, \mathbf{v}_1, \mathbf{q}_2, \mathbf{v}_2, \dots)$ is called a transcript of the execution.

Note that for each t, the choice of $\mathbf{q}_t$ is a deterministic function of $(\mathbf{r}_{<t}, \rho)$, where

$$\mathbf{r}_{<t} = (\mathbf{q}_1, \mathbf{v}_1, \mathbf{q}_2, \mathbf{v}_2, \dots, \mathbf{q}_{t-1}, \mathbf{v}_{t-1}),$$

and ρ denotes the randomness of A. If A terminates, its final output is a deterministic function of $(\mathbf{r}, \rho)$.

The Partition Size

Definition C.2.4. *Let Ω be a set, and let $\mathcal{S} \subseteq 2^\Omega$ be a collection of subsets. We say that $\mathcal{S}$ is a σ-algebra for Ω if it satisfies the following properties:*

- *$\Omega \in \mathcal{S}$.*
- *$\forall S \subseteq \mathcal{S}: \ \Omega \setminus S \in \mathcal{S}$.*
- *For any countable sequence $S_1, S_2, \dots \in \mathcal{S}: \ \cup_{i=1}^\infty S_i \in \mathcal{S}$.*

Definition C.2.5. *Let Ω be a set.*

- *Let $\mathcal{A} \subseteq 2^\Omega$ be a collection of subsets. The σ-algebra generated by $\mathcal{A}$ for Ω, denoted $\sigma(\mathcal{A})$, is the intersection of all σ-algebras for Ω that are supersets of $\mathcal{A}$.*
- *Let $\mathcal{F} \subseteq \{0,1\}^\Omega$ be a set of indicator functions. The σ-algebra generated by $\mathcal{F}$ for Ω is $\sigma(\mathcal{F}) = \sigma\left(\{A \subseteq \Omega : \ \mathbb{1}_A \in \mathcal{F}\}\right)$.*

Definition C.2.6. *Let $\mathcal{S}$ be a σ-algebra. The set of atoms of $\mathcal{S}$ is*

$$\mathsf{Atoms}(\mathcal{S}) = \{S \in \mathcal{S} : \ (\forall S' \in \mathcal{S} \setminus \varnothing : \ S' \not\subset S)\}.$$ [1]

Definition C.2.7. *Let Ω be a set and let $\mathcal{F} = \{f_1, f_2, \dots, f_k\} \subseteq \{0,1\}^\Omega$ be a finite set of indicator functions. The partition size of $\mathcal{F}$ is $\mathsf{PS}(\mathcal{F}) = |\mathsf{Atoms}(\sigma(\mathcal{F}))| \in \mathbb{N}$, i.e., the number of atoms in the σ-algebra generated by $\mathcal{F}$ for Ω.*

[1] $S' \not\subset S$ denotes that S' is not a strict subset of S.

Formal Statements

Theorem C.2.8 (PAC Verification of an SQ Algorithm)**.** *Let $b, s \in \mathbb{N}$, let Ω be a set and $\mathcal{H}$ be a discrete set. Let A be a statistical query algorithm that adaptively and randomly generates some random number T of batches $\mathbf{q}_1, \ldots, \mathbf{q}_T$ of statistical queries $\Omega \to \{0,1\}$ such that with probability 1, $T \leq b$ and $\mathsf{PS}(\mathbf{q}_t) \leq s$ for each $t \in [T]$, and the algorithm outputs a random value $h \in \mathcal{H}$. Let $\mathbb{D} \subseteq \Delta(\Omega)$ be a set of distributions, let $\tau > 0$, and let $L : \ \Omega \times \mathcal{H} \to [0,1]$ be a loss function.*

Then there exists a collection of oracles $\mathbb{O} = \{\mathcal{O}_\mathcal{D}\}_{\mathcal{D}\in\mathbb{D}}$ where $\mathcal{O}_\mathcal{D} \in \mathsf{SQ}(\mathcal{D}, \tau)$ for all $\mathcal{D} \in \mathbb{D}$, such that algorithm A with access to oracles $\mathbb{O}$ is PAC verifiable with respect to $\mathbb{D}$ by a verification protocol that uses random samples, where the verifier and honest prover respectively use

$$m_V = \Theta\left(\frac{\sqrt{s}\log(bk/\delta)}{\tau^2} + \frac{\log(k/\delta)}{\varepsilon^2}\right),$$

and

$$m_P = \Theta\left(\frac{s^3\log(sbk/\delta\tau)}{\tau^2}\right)$$

i.i.d. samples, with $k = \lceil 8\log(4/\delta)/\varepsilon \rceil$.

As a corollary, we obtain that for statistical query algorithms of a particular type, the sample complexity of PAC verification has a quadratically lower dependence on the VC dimension of the batches of statistical queries compared to simulating the algorithm using random samples.

Corollary C.2.9. *Let A be a statistical query algorithm as in Theorem C.2.8, and let $d \in \mathbb{N}$. Assume that in each time step $t \in [T]$, $\mathsf{VC}(\mathbf{q}_t) = d$ and $|\mathbf{q}_t| = 2^d$. Namely, $\mathbf{q}_t$ is the set of indicator functions of a σ-algebra with d atoms. Consider an implementation of A that uses random samples to simulate the SQ oracle accessed by A, such that the implementation uses random samples only and does not use any oracles. Simulating an oracle $\mathcal{O} \in \mathsf{SQ}(\mathcal{D}, \tau)$ requires*

$$m = \Omega\left(\frac{d + \log(1/\delta)}{\tau^2}\right)$$

i.i.d. samples from $\mathcal{D}$. In contrast, there exists a protocol that PAC verifies A such that the verifier uses only

$$m_V = \Theta\left(\frac{\sqrt{d}\log(bk/\delta)}{\tau^2} + \frac{\log(k/\delta)}{\varepsilon^2}\right)$$

i.i.d. samples from $\mathcal{D}$, with $k = \lceil 8\log(4/\delta)/\varepsilon \rceil$.

The lower bound in the corollary is the standard VC lower bound.

Proofs

Definition C.2.10. *Let A be a statistical query algorithm, let $\mathbb{D}$ be a collection of distributions, and let $\varepsilon, \tau > 0$. We say that a collection of oracles $\mathbb{O} = \{\mathcal{O}_{\mathcal{D}}\}_{\mathcal{D} \in \mathbb{D}}$ is $\underline{\varepsilon\text{-maximizing with respect to } A \text{ and } \mathbb{D}}$ if for each $\mathcal{D} \in \mathbb{D}$, $\mathcal{O}_{\mathcal{D}} \in \mathsf{SQ}(\mathcal{D}, \tau)$ and $\mathbb{E}\Big[L_{\mathcal{D}}\Big(A^{\mathcal{O}_{\mathcal{D}}}\Big)\Big] \geq \sup_{\mathcal{O} \in \mathsf{SQ}(\mathcal{D},\tau)} \mathbb{E}\Big[L_{\mathcal{D}}\Big(A^{\mathcal{O}}\Big)\Big] - \varepsilon$.*

Proof of Theorem C.2.8. Fix a collection of oracles $\mathbb{O} = \{\mathcal{O}_{\mathcal{D}}\}_{\mathcal{D} \in \mathbb{D}}$ that is $\varepsilon/4$-maximizing with respect to A and $\mathbb{D}$. We show that algorithm A with access to the oracles $\mathbb{O}$ is PAC verified by Protocol C.2.

To establish completeness, notice that each batch $\mathbf{a}_t$ of queries sent to the prover by VerifierIteration satisfies $\mathsf{VC}(\mathbf{a}_t) = 1$, and there are at most $b \cdot k$ such batches. Hence, by Theorem 5.4.3 and a union bound, taking m_P as in the statement is sufficient to guarantee that with probability at least $1 - 1/4$,

$$\forall \text{ iteration } i \in [k] \ \forall t \in [T]: \ \ \|\tilde{\mathbf{p}}_t - \mathbf{p}_t\|_\infty \leq \frac{\tau}{s\sqrt{s}},$$

where $\mathbf{p}_t$ is the vector of correct evaluations, with components $p_t^j = \mathbb{E}_{Z \sim \mathcal{D}}\Big[a_t^j(Z)\Big]$. Hence, with probability at least $1 - 1/4$,

$$\forall \text{ iteration } i \in [k] \ \forall t \in [T]: \ \ \|\tilde{\mathbf{p}}_t - \mathbf{p}_t\|_1 \leq \frac{\tau}{\sqrt{s}}. \tag{C.2}$$

By Eq. (C.2), Theorem 5.4.1, and the choice of m_V, with probability at least $1 - 1/4$, none of the executions of IdentityTest cause the verifier to reject.

By a union bound, with probability at least $1 - 1/2$, Eq. (C.2) holds and the verifier does not reject. Then, by Lemma C.2.11,

$$\forall i \in [k]: \ \mathbb{P}\Big[L_{\mathcal{D}}(h_i) \leq L_{\mathcal{D}}(A) + \frac{\varepsilon}{2}\Big] \geq \frac{\varepsilon}{8}. \tag{C.3}$$

By the choice of k,

$$\mathbb{P}\Big[\forall i \in [k]: \ L_{\mathcal{D}}(h_i) > L_{\mathcal{D}}(A) + \frac{\varepsilon}{2}\Big] \leq \Big(1 - \frac{\varepsilon}{8}\Big)^k \leq e^{-\varepsilon k/8} \leq \frac{1}{4}\delta. \tag{C.4}$$

By Hoeffding's inequality, a union bound, and the choice of m_V,

$$\mathbb{P}\Big[\forall i \in [k]: \ \ \Big|L_{S'_V}(h_i) - L_{\mathcal{D}}(h_i)\Big| \leq \frac{\varepsilon}{2}\Big] \geq 1 - \frac{1}{4}\delta. \tag{C.5}$$

Combining Eqs. (C.2), (C.4) and (C.5) via a union bound, we conclude that with probability $1 - 1$, the verifier does not reject and it outputs $h \in \mathcal{H}$ such that $L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(A) + \varepsilon$. This establishes completeness.

To establish soundness, consider an interaction between the verifier of Protocol C.2 and any deterministic or randomized (possibly malicious and computationally unbounded) prover P', and examine the following two events.

- Event I: the evaluations provided by P' satisfy

  $$\forall \text{ iteration } i \in [k]\ \forall t \in [T]: \quad \|\tilde{\mathbf{p}}_t - \mathbf{p}_t\|_1 \leq \tau. \tag{C.6}$$

  If the verifier does not reject then Lemma C.2.11 implies that Eq. (C.3) holds. As we saw in the proof for the completeness requirement, this implies that with probability at least $1 - 1$, the verifier outputs $h \in \mathcal{H}$ such that $L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(A) + \varepsilon$.

- Event II: there exists an iteration $i \in [k]$ containing a time step $t^* \in [T]$ such that $\|\tilde{\mathbf{p}}_{t^*} - \mathbf{p}_{t^*}\|_1 > \tau$. By Theorem 5.4.1 and the choice of m_V, with probability at least $1 - {}^1\!/\!_4$ the verifier rejects in time step t^*.

We conclude that in both cases,

$$\mathbb{P}_{S_V \sim \mathcal{D}^{m_V}}[h = \mathsf{reject} \ \vee \ L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(A) + \varepsilon] \geq 1 - \delta,$$

and this establishes soundness. □

Lemma C.2.11. *In the context of Theorem C.2.8, fix a distribution $\mathcal{D} \in \mathbb{D}$ and let $\mathcal{O}_{\mathcal{D}} \in \mathsf{SQ}(\mathcal{D}, \tau)$ be an oracle such that*

$$\mathbb{E}\Big[L_{\mathcal{D}}\Big(A^{\mathcal{O}_{\mathcal{D}}}\Big)\Big] \geq \sup_{\mathcal{O} \in \mathsf{SQ}(\mathcal{D},\tau)} \mathbb{E}\Big[L_{\mathcal{D}}\Big(A^{\mathcal{O}}\Big)\Big] - \varepsilon/4.$$

Consider an execution of VerifierIteration *(Protocol C.3). Let G denote the event in which the verifier does not reject, and the query evaluations $\tilde{\mathbf{p}}_t$ provided by the prover satisfy*

$$\forall t \in [T]: \quad \|\tilde{\mathbf{p}}_t - \mathbf{p}_t\|_1 \leq \tau, \tag{C.7}$$

where $\mathbf{p}_t$ is the vector of correct evaluations $p_t^i = \mathbb{E}_{Z \sim \mathcal{D}}[a_t^i(Z)]$. Then the output $h_i \in \mathcal{H}$ returned by VerifierIteration *satisfies*

$$\mathbb{P}\Big[L_{\mathcal{D}}(h_i) \leq \mathbb{E}\Big[L_{\mathcal{D}}\Big(A^{\mathcal{O}_{\mathcal{D}}}\Big)\Big] + \frac{\varepsilon}{2} \ \Big| \ G\Big] \geq \frac{\varepsilon}{8}. \tag{C.8}$$

Proof. Let $\mathcal{O}_G$ denote the oracle with evaluations that are equal in distribution to the evaluations provided by the prover conditioned on event G occurring. By the choice of $\mathcal{O}_{\mathcal{D}}$,

$$\mathbb{E}[L_{\mathcal{D}}(h_i) \mid G] = \mathbb{E}\Big[L_{\mathcal{D}}\Big(A^{\mathcal{O}_G}\Big)\Big] \leq \mathbb{E}\Big[L_{\mathcal{D}}\Big(A^{\mathcal{O}_{\mathcal{D}}}\Big)\Big] + \varepsilon/4.$$

By Markov's inequality,

$$\begin{aligned}
\mathbb{P}\Big[L_{\mathcal{D}}(h_i) > \mathbb{E}\Big[L_{\mathcal{D}}\Big(A^{\mathcal{O}_{\mathcal{D}}}\Big)\Big] + \varepsilon/2 \ \Big| \ G\Big] &\leq \mathbb{P}\Big[L_{\mathcal{D}}(h_i) > \mathbb{E}[L_{\mathcal{D}}(h_i) \mid G] + \varepsilon/4 \ \Big| \ G\Big] \\
&\leq \frac{\mathbb{E}[L_{\mathcal{D}}(h_i) \mid G]}{\mathbb{E}[L_{\mathcal{D}}(h_i) \mid G] + \varepsilon/4} \\
&\leq \frac{1}{1 + \varepsilon/4},
\end{aligned}$$

since $L_{\mathcal{D}}$ is at most 1. Hence, the complement satisfies

$$\mathbb{P}\Big[L_{\mathcal{D}}(h_i) \le \mathbb{E}\Big[L_{\mathcal{D}}\Big(A^{\mathcal{O}_{\mathcal{D}}}\Big)\Big] + \frac{\varepsilon}{2} \;\Big|\; G\Big] \le \frac{\varepsilon/4}{1+\varepsilon/4} \le \frac{\varepsilon}{8},$$

as desired. □

C.3 Concentration of Measure

Theorem C.3.1 (Hoeffding, 1963)**.** *Let $a, b, \mu \in \mathbb{R}$ and $m \in \mathbb{N}$. Let $Z_1, \ldots, Z_m$ be a sequence of i.i.d. real-valued random variables and let $Z = \frac{1}{m}\sum_{i=1}^{m} Z_i$. Assume that $\mathbb{E}[Z] = \mu$, and for every $i \in [m]$, $\mathbb{P}[a \le Z_i \le b] = 1$. Then, for any $\varepsilon > 0$,*

$$\mathbb{P}[|Z - \mu| > \varepsilon] \le 2\exp\left(\frac{-2m\varepsilon^2}{(b-a)^2}\right).$$

Assumptions:

- $d, 1/\varepsilon \in \mathbb{N}$ (this can always be achieved by making ε smaller if necessary), $k = 12d/\varepsilon$.
- $m_P = O((d^2 \log(d/\varepsilon) + \log(1/\delta))\varepsilon^{-4})$ is a multiple of k.
- $m_V = O\left(\sqrt{d}\log(1/\delta)\varepsilon^{-2.5}\right)$.
- $S_V \sim \mathcal{D}^{m_V}$, $S_P \sim \mathcal{D}^{m_P}$.
- $\mathcal{D} \in \Delta([0,1] \times \{0,1\})$ is an unknown target distribution.

```
PROVER($S_P, \delta, \varepsilon$):
  $I_1, I_2, \dots, I_k \leftarrow$ a partition of $[0,1]$ into disjoint intervals such that $\cup_{i\in[k]} I_i = [0,1]$
            and $\forall j \in [k]: \ |\{x_1^P, \dots, x_{m_P}^P\} \cap I_j| = m_P/k$.

  for $j \in [k]$:
    for $b \in \{0,1\}$:
      $\tilde{P}_{j,b} \leftarrow |\{(x,y) \in S_P : \ x \in I_j \ \wedge \ y = b\}|/m_P$        ▷ Counted as a multiset
  send $(I_1, \dots, I_k)$ and $\left(\tilde{P}_{j,y}\right)_{j\in[k], y\in\{0,1\}}$ to the verifier

VERIFIER($S_V, \delta, \varepsilon$):
  receive $(I_1, \dots, I_k)$ and $\left(\tilde{P}_{j,y}\right)_{j\in[k], y\in\{0,1\}}$ from the prover

  if $\exists j \in [k]$ s.t. $\tilde{P}_{j,0} + \tilde{P}_{j,1} \neq 1/k$:
    output reject and terminate

  $x_1^*, \dots, x_k^* \leftarrow$ arbitrary points such that $\forall j \in [k]: \ x_j^* \in I_j$

  execute the tester of Theorem 5.4.1 with parameters $\varepsilon/6$, $\delta/2$ where $\mathcal{P}, \tilde{\mathcal{P}} \in$
  $\Delta([0,1] \times \{0,1\})$ are as follows:
    - $\mathcal{P}$ is the distribution generated by sampling $(x,y) \sim \mathcal{D}$ and then outputting
      $(x^*, y)$ where $x^* = x_j^*$ such that $x \in I_j$
    - $\tilde{\mathcal{P}}$ is the distribution such that $\mathbb{P}\left[(x_j^*, y)\right] = \tilde{P}_{j,y}$ for all $j \in [k], y \in \{0,1\}$

  if distribution identity tester rejects:
    output reject and terminate

  $h \leftarrow \arg\min_{h' \in \mathcal{H}_d} L_{\tilde{\mathcal{P}}}^{0\text{-}1}(h')$

  output $h$
```

Protocol C.1: Verification protocol for unions of d-intervals.

Assumptions:

- Ω is a set, $\mathcal{D} \in \Delta(\Omega)$ is the population distribution.
- A is a statistical query algorithm to be verified.
- $\tau \in (0,1)$ is the accuracy parameter for statistical queries used by A.
- $b \in \mathbb{N}$ is an upper bound on the number of statistical query batches generated by A.
- $\varepsilon, \delta \in (0,1)$ are the desired accuracy and confidence parameters for the verification.
- $k = \lceil 8\log(4/\delta)/\varepsilon \rceil$.
- $m_V = \Theta(\sqrt{s}\log(bk/\delta)\tau^{-2} + \log(k/\delta)\varepsilon^{-2})$.
- $m_P = \Theta(s^3 \log(sbk/\delta\tau)\tau^{-2})$.
- $S_V, S'_V \sim \mathcal{D}^{m_V}$, $S_P \sim \mathcal{D}^{m_P}$ are independent sets of i.i.d. samples.
- $S_V = (z_1^V, \ldots, z_{m_V}^V)$, $S'_V = (z_1^{V'}, \ldots, z_{m_V}^{V'})$, $S_P = (z_1^P, \ldots, z_{m_P}^P)$.

```
$\textsc{Verifier}(S_V, S'_V)$:
    for $i \in [k]$:
        $h_i \leftarrow \textsc{VerifierIteration}(S_V)$        ▷ Protocol C.3
    $i^* \leftarrow \arg\min_{i \in [k]} L_{S'_V}(h_i)$
    output $h_{i^*}$
```

```
$\textsc{Prover}(S_P)$:
    loop forever:
        $q \leftarrow$ receive query from verifier
        $v \leftarrow \frac{1}{m_P} \sum_{i \in [m_P]} q\left(z_i^P\right)$
        send $v$ to verifier
```

Protocol C.2: A PAC verification protocol for statistical query algorithms.

Assumptions: As in Protocol C.2.

```
VerifierIteration($S_V$):
  for $t \leftarrow 1, 2, \ldots$:
      simulate $A$ until it sends a batch of queries or produces an output
      if $A$ sends a batch of queries $\mathbf{q}_t$:
          if $t \geq b$:
              output reject and terminate
          $\mathbf{a}_t \leftarrow \mathsf{Atoms}(\sigma(\mathbf{q}_t))$
          send $\mathbf{a}_t$ to prover
          receive $\tilde{\mathbf{p}}_t$ from prover
          IdentityTest($S_V, \mathbf{a}_t, \tilde{\mathbf{p}}_t, \tau$)
          $\tilde{\mathbf{v}}_t \leftarrow$ evaluations for $\mathbf{q}_t$ induced by $\tilde{\mathbf{p}}_t$
          send $\tilde{\mathbf{v}}_t$ to $A$
      else if $A$ produces output $h$:
          return $h$

IdentityTest($S_V, \mathbf{a}_t, \tilde{\mathbf{p}}_t, \tau$):
  for $j \in [m_V]$:
     $i_j \leftarrow i \in [|\mathbf{a}_t|]$ such that $a_t^i(z_j^V) = 1$
  execute the distribution identity tester of Theorem 5.4.1
       with sample $I = (i_1, \ldots, i_{m_V})$ to distinguish with
       probability at least $1 - \varepsilon\delta/4b$ between

           $\mathsf{TV}(\tilde{\mathbf{p}}_t, \mathbf{p}_t) \leq \frac{\tau}{2\sqrt{|\mathbf{a}_t|}}, \quad \text{and} \quad \tau \leq \mathsf{TV}(\tilde{\mathbf{p}}_t, \mathbf{p}_t)$

       where $\mathbf{p}_t$ is the distribution that generated $I$
  if identity tester rejects:
      output reject and terminate
```

Protocol C.3: A subroutine of Protocol C.2.