Model Theory and Machine Learning

by

Hunter Sato Chase
B.S., The University of Chicago, 2014
M.S., University of Illinois at Chicago, 2016

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mathematics
in the Graduate College of the
University of Illinois at Chicago, 2020

Chicago, Illinois

Defense Committee:
James Freitag, Chair and Advisor
David Marker
Lev Reyzin
György Turán
Michael Chris Laskowski, University of Maryland, College Park

For my parents.

ACKNOWLEDGMENTS

Throughout the course of any significant undertaking, one naturally accumulates a large number of those particular debts that cannot properly be repaid, and instead can only be acknowledged. I have received an incredible amount of support from many people, and my success would not have otherwise been possible.

I must first thank Jim Freitag, my advisor. His support and encouragement have been indispensable. Moreover, Jim and I worked together on the projects which comprise this thesis, and the results are his as much as they are mine.

My work has been shaped for the better by many useful ideas, comments, and suggestions from a great number of people. These include Siddharth Bhaskar, Artem Chernikov, Gabe Conant, Dimitrios Diochnos, Kyle Gannon, Vincent Guingona, Cameron Hill, Alex Kruckman, Chris Laskowski, Maryanthe Malliaris, Dave Marker, Dhruv Mubayi, Lev Reyzin, Caroline Terry, György Turán, and the anonymous referee of [12]. In particular, I would like to recognize Chris Laskowski, Dave Marker, Lev Reyzin, and György Turán for serving on my committee and reviewing my thesis.

I am deeply indebted to the many fantastic math teachers I have had, both formal and informal, from elementary school through graduate school. They instilled and cultivated an excitement and passion for mathematics, without which I could not have undertaken such an endeavor. Of special mention is Maryanthe Malliaris, whose undergraduate logic course

inspired me (and many others) to study logic. Also of note is Stephanie Scheffler, who was both an exquisite teacher and, more recently, a useful source of life advice.

I have received a lot of support from my fellow graduate students, past and present. Having some compatriots with whom to commiserate on what is a long and brutal slog makes it a little easier to weather. Chief among them are Will Adkisson, Tom Dean, Matt DeVilbiss, Nathan Lopez, Keaton Quinn, Sam Shideler, and Jonathan Wolf, but there are far too many others to enumerate in full.

Mathematics is its own strange little world, which one enters at their peril. I am deeply grateful to my friends outside of mathematics who have kept me grounded, given me perspective, and helped me breathe when I needed air. In particular, I must thank Quinn Colter, Kristy Hwang, Rachel Kulikoff, Hannah Mark, Jason McCreery, Kevin Rose, Vidur Sood, and Mallory VanMeeter, although again, there are too many to name. In much the same vein, I would like to thank Nikki Falk for helping me work through a lot of the issues that pop up when the atmosphere of mathematics clouds one's thinking.

Finally, I thank Michael Chase and Patricia Sato, my parents. They have been steadfast and unwavering in their love and support. I love them deeply and owe them everything.

HSC

CONTRIBUTIONS OF AUTHORS

Chapter 2 represents section 4 of [14], co-authored with James Freitag.

Chapter 3 represents sections 2–4 and a portion of section 1 of the preprint [13], co-authored with James Freitag.

Chapter 4 represents section 5 and a portion of section 1 of the preprint [13], co-authored with James Freitag.

Chapter 5 represents the preprint [12], co-authored with James Freitag.

Except as noted, all content in these chapters, including introduction, definitions, theorems, and writing of the various manuscripts was done jointly with James Freitag.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

LIST OF FIGURES

SUMMARY

We study a connection between model theory and machine learning by way of common combinatorial properties. In various settings of machine learning, combinatorial properties of the concept class being learned dictate whether the class is learnable, and if so, how long or how much data is required. In model theory, combinatorial properties can give rise to dividing lines, which create classes of theories in which a structure theory can be developed to varying degrees. Model theory and machine learning share several common combinatorial properties. The first known connection was between PAC learning and NIP theories by way of VC dimension, which led to subsequent interaction between the two fields.

In this thesis, we describe a broad connection between stability and several forms of exact learning. The key combinatorial property is Littlestone dimension. This property has been known to both model theory and machine learning for decades (although by a different name in model theory), although it had not been previously pointed out that the connection existed. Finite Littlestone dimension classifies online learning, in which a learner classifies sequentially presented data. Finite Littlestone dimension also classifies stable theories, a model-theoretic class in which a rich structure theory has been developed. We extend this connection to query learning, in which the learner attempts to identify the target concept by guessing it exactly and receiving feedback to its guesses. We also consider compression schemes, where sets must be encoded by a bounded number of its elements and reconstructed by one of several reconstruction functions.

SUMMARY (Continued)

The boundary between finite and infinite Littlestone dimension that corresponds to learnability and non-learnability has many similarities to the boundary between finite and infinite VC dimension. In both settings, one can define a counting function called the shatter function, and the relevant dimension controls the growth rate of this function. In particular, the appropriate Sauer-Shelah Lemma gives a polynomial bound if the relevant dimension is finite, while the function is exponential if the relevant dimension is infinite. We develop a framework for proving bounds in a uniform way and apply it in similar settings.

CHAPTER 1

INTRODUCTION

Model theory uses logical systems (usually first-order logic) to study mathematical structures. Given a structure, one can consider the theory of that structure in an appropriate language; that is, the collection of sentences in that language which are true of that structure. For example, taking the first-order theory of the complex numbers in the language $\mathcal{L} = \{0, 1, +, -, \cdot\}$ yields ACF_0, the first-order theory of algebraically closed fields of characteristic 0. Model theory studies the relationship between first-order theories and the structures (models) that satisfy those theories.[1] Model theory also considers the structure of definable sets. Given a structure $\mathcal{M}$ and a formula $\phi(x)$ with free variable x, ϕ defines the set

$$\phi(M) := \{a \in M \mid \mathcal{M} \models \phi(a)\}$$

by picking out those elements of M which satisfy the formula. For formulas with parameters, one can further consider the family of definable sets uniformly generated by that formula. A structure $\mathcal{M}$ and a formula $\phi(x, y)$ give rise to the uniform family of ϕ-definable sets

$$\mathcal{F}_\phi := \{\phi(M; b) \mid b \in M\}.$$

---

[1] For a reference in model theory, see, e.g. [31], [41].

One could further consider the family of externally ϕ-definable sets by allowing the parameters b to range over an $\mathcal{M}$-saturated elementary extension $\mathcal{N}$, rather than $\mathcal{M}$.

A major program in model theory is to develop a structure theory for definable sets. This is not possible in general, but there are several *dividing lines* that establish broad classes of theories with common properties, where structure theory can be developed to varying degrees. Typically, a dividing line will have a tame side, where one can obtain general structural results about definable sets, and a wild side, where one can obtain results about the lack of structure.

Many dividing lines can be characterized in several ways. For example, *stable theories* can be classified by counting types, by the collapse of indiscernible sequences to indiscernible sets, by the absence of a formula with the order property, or by the absence of a formula with infinite Littlestone dimension (and this is not a complete list). The latter two characterizations are local combinatorial properties, which examine a single formula at a time. The presence or absence of a certain combinatorial property in a family of definable sets is a common way of characterizing dividing lines.

Machine learning is the study of algorithms that use sample data to learn or make predictions. In the general setup, a learner is given a concept class $\mathcal{C}$ consisting of subsets of a base set X. The learner is asked to learn a target concept $A \in \mathcal{C}$ based on sample data. There are many versions of machine learning. A particular type of learning will specify what it means to learn A, how the data is received, and any other requirements or restrictions. For example, the learner may be asked to identify A exactly, or approximate it up to a small error with respect to some measure. The sample data may be randomly drawn or may be obtained by queries. For

any given type of learning, we say that $\mathcal{C}$ is learnable if every $A \in \mathcal{C}$ can be learned. We wish to characterize those classes $\mathcal{C}$ which are learnable in a given setting. Remarkably, in several variants of machine learning, learnability is characterized by a combinatorial property that also characterizes a dividing line in model theory.

Work on the interaction between model theory and machine learning began when Laskowski observed that *NIP theories* were characterized by every formula having finite VC dimension, another local combinatorial property [24]. Under reasonable measurability conditions, finite VC dimension also characterizes those set systems which are probably approximately correct (PAC) learnable, a kind of learning where sample data is drawn randomly, and the learner must identify the target concept up to a small error with high probability. Moreover, model theory provides a wealth of examples. A partitioned formula $\phi(x; y)$ has finite VC dimension precisely when the uniform family of ϕ-definable sets in some model $\mathcal{M}$ has finite VC dimension.

The connection between NIP theories and PAC learning by way of finite VC dimension has formed the basis of most of the interaction between model theory and machine learning. It has led to work on topics such as compression schemes, uniformly definable types over finite sets, and honest definitions of types. This has contributed to the development of the dividing line between NIP and IP theories, which is the second-most prominent and developed dividing line in model theory.

The present work concerns the *most* prominent dividing line, that between stable and unstable theories. Recall that stable theories are characterized by each formula having finite Littlestone dimension. The importance of Littlestone dimension has been known to both ma-

chine learning and model theory (under the alias of Shelah 2-rank); however, in [14], we point out that Littlestone dimension was common to *both* fields. In machine learning, a set system is *online-learnable* if a learner, presented with a sequence of elements and tasked with determining membership of those elements in the target concept, can make a uniformly bounded number of mistakes. A set system is online-learnable if and only if it has finite Littlestone dimension. In particular, stable formulas are precisely those formulas that generate set systems that are online-learnable.

We also explore further connections between several kinds of query learning and model theory. In query learning, a learner obtains data by submitting two kinds of queries. A learner can submit an equivalence query, submitting a guess of the target concept and receiving a counterexample, or a membership query, asking about the membership of a specific element. The connection makes use of both Littlestone dimension and consistency dimension, the latter of which roughly corresponds with formulas without the finite cover property. Stable nfcp formulas can be used to generate concept classes along with corresponding hypothesis classes necessary for learning to succeed.

A related notion is that of compression schemes. Given a concept class $\mathcal{C}$, a compression scheme is a method of encoding the traces of concepts in $\mathcal{C}$ on finite sets. Given some $C \in \mathcal{C}$ and a finite $F \subseteq X$, a d-compression returns d elements from F. Subsequently, one of several reconstruction functions recovers the entire behavior of C on F from the behavior of C on the d-element compression. In the case where $\mathcal{C}$ has finite Littlestone dimension,

we use Littlestone dimension to bound the size of the encoding together with the number of reconstruction functions necessary.

A fundamental tool tying VC dimension to machine learning is the Sauer-Shelah Lemma, which establishes a divide between polynomial and exponential growth rates of the shatter function of a class $\mathcal{C}$, depending on whether $\mathcal{C}$ has finite or infinite VC dimension. A nearly identical result for Littlestone dimension was formulated by Bhaskar [11]. We develop a tool that provides the flexibility to prove such results in a uniform way. In particular, we apply the tool to the op-rank setting of Guingona and Hill [20], which seeks to generalize Littlestone dimension and other notions of dimension arising in model theory. We use our tool to show that the growth rate of the corresponding shatter function is controlled by the *op-dimension* of the set system.

1.0.1 Organization

- In Chapter 2 we describe a connection between online learning and stable formulas.
- In Chapter 3 we study several types of query learning and make a connection with stable formulas without the finite cover property.
- In Chapter 4 we study compression schemes.
- In Chapter 5 we describe a framework that can be used to prove several variants of the Sauer-Shelah Lemma.

Within each chapter, we provide additional background to the material within that chapter.

CHAPTER 2

STABILITY AND ONLINE LEARNING

This chapter represents section 4 of [14], co-authored with James Freitag.

Minor edits have been made for consistency with the rest of the thesis.

The initial setting of online learning which we describe is due to Littlestone [25]; the particular setting received relatively little attention, perhaps due to the very strong assumptions ([25] is in fact famous for several other contributions). Littlestone's work was generalized in various ways in the ensuing years, with the assumptions being significantly weakened. We will begin with the original setup of [25], and eventually describe two settings laid out in [10]. First, we set up some of the combinatorial notions pertinent in each of the settings we consider.

The next several definitions follow the notation and terminology of Bhaskar [11], although we prefer "Littlestone dimension" instead of Bhaskar's "thicket dimension," and use "stable" to describe the general setting.

Definition 2.0.1. A *binary element tree of height* h, denoted by $\mathcal{T}_h$, is a rooted complete binary tree of height h whose non-leaf vertices are labeled by elements of the set X and whose leaves are labeled by elements of $\mathcal{C}$ (see Figure 1).

For the following definitions, fix a binary element tree of height h.

Definition 2.0.2. A vertex v_1 is *below* a vertex v_2 if v_2 lies on the (unique) path from v_1 to the root of the tree. We say that v_1 is *left-below* v_2 if v_1 is below v_2 and the first edge along the path from v_2 to v_1 goes down and to the left. The notion of *right-below* is defined analogously. When a vertex labeled by b is left-below a vertex labeled by a, we write $a <_L b$. Similarly, when a vertex labeled by b is right-below a vertex labeled by a, we write $a <_R b$.

Definition 2.0.3. A leaf, labeled by $A \in \mathcal{C}$ is said to be *well-labeled* if for each vertex above Y, say labeled by a,

$$a \in A \text{ if and only if } a <_R A.$$

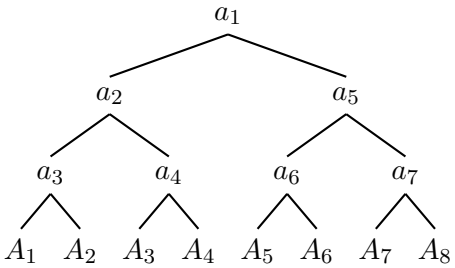

Figure 1: A binary element tree of height three. Here $a_i \in X$ and $A_i \in \mathcal{C}$. The leaf labeled with A_4 is well-labeled if and only if $a_1 \notin A_4$ and $a_2, a_4 \in A_4$. For all other a_i, there is no requirement about membership in A_4.

Definition 2.0.4. The *stable shatter function* $\rho_{\mathcal{F}} : \mathbb{Z}^{\geq 0} \to \mathbb{Z}^{\geq 0}$ is defined by letting $\rho_{\mathcal{F}}(n)$ be the maximum number of well-labeled leaves on a binary element tree of height n, $\mathcal{T}_n$, whose

leaves are labeled with elements of $\mathcal{F}$. The *Littlestone dimension* $\mathrm{Ldim}(\mathcal{F})$ is the maximum integer n such that $\rho_{\mathcal{F}}(n) = 2^n$, or else $\mathrm{Ldim}(\mathcal{F}) = \infty$.

Littlestone dimension has appeared in several other contexts under different names; in fact, Bhaskar [11] was aware of the terminology and definitions of [38], which we reproduce next:

Definition 2.0.5. Let $\mathcal{M}$ be a monster model of a complete $\mathcal{L}$-theory. Fix a consistent partial type $\pi(x)$ and a partitioned formula $\phi(x; y)$. Then the ordinal $R(\pi, \phi, 2)$, called the Shelah 2-rank, is defined as follows:

- $R(\pi, \phi, 2) \geq 0$.
- For any limit ordinal λ, $R(\pi, \phi, 2) \geq \lambda$ if $R(\pi, \phi, 2) \geq \alpha$ for all $\alpha < \lambda$.
- For any ordinal α, $R(\pi, \phi, 2) \geq \alpha+1$ if there is some $\phi(x, a)$ such that $R(\pi \cup \{\phi(x, a)\}, \phi, 2) \geq \alpha$ and $R(\pi \cup \{\neg\phi(x, a)\}, \phi, 2) \geq \alpha$.

In general, $R(\pi, \Delta, 2)$ can also be defined for a finite collection of formulas Δ, but this case can be shown to reduce to the case of a single formula. The formula $\phi(x, y)$ is *stable* if and only if $R(\{x = x\}, \phi, 2)$ is finite [38]; a theory is stable if every formula is stable. It is reasonably clear that the $R(\pi, \phi, 2)$ is the Littlestone dimension of the set system on $\mathcal{M}^{|y|}$ given by the collection of sets $\{\phi(b, \mathcal{M}) \mid b \in \pi(\mathcal{M})\}$; for more details, see [11].

Littlestone dimension also appears for the first time in the context of learning theory in [25].

2.0.1 The realizable case

Fix a set system $\mathcal{C}$ on a set X. Assume that $Y = Y' = \{0, 1\}$ and the loss function for a prediction $\hat{y}$ and concept (that is, a set) A on input x is given by $|\hat{y} - 1_A(x)|$. Over all possible

algorithms, we seek to minimize our loss, that is, the number of mistakes we make over n rounds of predictions. In the *realizable* case, we assume that $A \in \mathcal{C}$, so that the true concept is among the set of concepts $\mathcal{C}$ accessible to the learner. There are *no* assumptions on the choices of the instances x_t. The goal is to minimize the worst-case number of mistakes made by our predictions over all possible samples of the instances and choice of the concept. So, we seek to bound

$$M = \max_{A \in \mathcal{C}} \max_{\bar{x}=(x_1,\ldots,x_n)} \sum_{t=1}^{n} |\hat{y}_t - 1_A(x_t)|,$$

where $\hat{y}_t$ is chosen by some deterministic algorithm.

For applications and purposes of discussing the bounds, one often views the entity selecting the instances $\bar{x}$ as antagonistic to the learner—and in our current simplified setting, bounding the worst-case number of mistakes bounds the actual number of mistakes made when the antagonistic sampling entity has perfect information about the prediction process.

Theorem 2.0.6. *[25] The worst-case number of mistakes of any deterministic algorithm in the online learning setting with concept class $\mathcal{C}$ is at least the Littlestone dimension of $\mathcal{C}$, and there is an algorithm that makes at most this many mistakes.*

Remark 2.0.7. The algorithm which minimizes the number of worst-case mistakes in the above setting is referred to as the Standard Optimal Algorithm, and we describe it briefly here. Begin with $V_0 = \mathcal{C}$. At each stage, the learner inductively defines V_i. At stage t, the learner receives x_t, and sets, for $r = 0, 1$,

$$V_t^{(r)} := \{A \in V_{t-1} \,|\, 1_A(x_t) = r\}.$$

The learner predicts $\hat{y}_t = r$ which maximizes the Littlestone dimension of $V_t^{(r)}$ (ties are predicted in some fixed manner, say $\hat{y}_t = 0$ in the case of a tie). Then the learner gets the value of $1_A(x_t)$ and realizes whether a mistake has been made. At this point, set $V_t = V_t^{1_A(x_t)}$.

The essential point here is that if a mistake is made, it must be the case that the Littlestone dimension of V_t is strictly less than the Littlestone dimension of V_{t-1} (proving this is an easy exercise). Of course, this bounds the total number of mistakes which the algorithm can ever make under any choice of $\bar{x}$ by the Littlestone dimension.

Where $\mathcal{C}$ is generated by a stable formula $\phi(z, x)$, say $\mathcal{C} = \{\phi(b, \mathcal{M}) \mid b \in \mathcal{M}\}$, the algorithm equivalently functions as follows. Begin with the partial type $\pi_0(z) = \{z = z\}$, and inductively define $\pi_i(z)$. When the learner receives x_t, the learner predicts $\hat{y}_t = r$, where r maximizes $R(\pi_{t-1} \cup \{\phi(z, x_t)^r\}, \phi, 2)$, where $\phi(z, x)^1 = \phi(z, x)$ and $\phi(z, x)^0 = \neg\phi(z, x)$. Upon receiving $1_A(x_t)$, set $\pi_t(z) = \pi_{t-1}(z) \cup \{\phi(z, x_t)^{1_A(x_t)}\}$. Again, a mistake on x_t will mean $R(\pi_t, \phi, 2) < R(\pi_{t-1}, \phi, 2)$.

2.0.2 Learning from experts

The case in which we assume that the learner has access to true concept $A \in \mathcal{C}$ is often referred to as the *realizable* case of online learning. For various applications, this assumption is too strong (as are other assumptions from the previous subsection which we will deal with in later sections). In this section, we will explain a context of online learning which removes the realizability assumption.

The goal again is to minimize mistakes, but here, the minimization will be relative to a particular class of $\{0, 1\}$-valued functions, which we will call $\mathcal{H}$. That is, we wish to minimize,

for any sampling of instances, $\bar{x} = (x_1, \ldots, x_T)$, the difference between the number of mistakes made by the learner and the minimal number of mistakes made by any of the functions in $\mathcal{H}$. So, in this case, the loss function is taken to be

$$\sum |\hat{y}_t - y_t| - \min_{h \in \mathcal{H}} \sum |h(x_t) - y_t|.$$

Here one often thinks intuitively that the functions in $\mathcal{H}$ are experts making predictions, and the learner's job is to choose which expert's prediction to believe.

Littlestone and Warmuth [27] consider this problem in the case that $\mathcal{H}$ is finite via a probabilistic weighted majority algorithm. We will now describe their algorithm. At the outset, each of the N many experts $\{f_i\}_{i=1}^N = \mathcal{H}$ is assigned weight 1, and the weight of expert i at stage t will be denoted by w_i^t. We fix the learning rate $\eta > 0$, which dictates how much we discount the weight of an expert for providing incorrect advice. At each stage, the learner receives the expert advice, $(f_1(x_t), \ldots, f_N(x_t))$, a tuple in $\{0,1\}^N$. The learner predicts 1 with probability

$$p_t = \frac{1}{\sum_{i=1}^N w_i^{t-1}} \sum_{i=1}^N w_i^{t-1} f_i(x_t).$$

Then once the actual value y_t is revealed, the weights are updated via: $w_i^t = w_i^{t-1} e^{-\eta \cdot |f_i(x_t) - y_t|}$. That is, those experts who were wrong see their weight drop by a factor of $e^{-\eta}$.

The expected value of the loss function of their algorithm with a sample of size T is

$$\sum_{t=1}^T E(|\hat{y}_t - y_t|) - \min_{h \in \mathcal{H}} \sum_{t=1}^T |h(x_t) - y_t| \leq \sqrt{\frac{1}{2} \ln(N) T}.$$

Here, the assumption that $\mathcal{H}$ is finite is often too strong for applications, however, [10] generalize the setup to the case in which $\mathcal{H}$ is infinite, but of finite Littlestone dimension, proving:

Theorem 2.0.8. *There is an algorithm such that for all $h \in \mathcal{H}$ and any sequence of instances $\bar{x} = (x_1, \ldots, x_T)$,*

$$\sum_{t=1}^{T} E(|\hat{y}_t - y_t|) - \min_{h \in \mathcal{H}} \sum_{t=1}^{T} |h(x_t) - y_t| \leq \sqrt{\frac{1}{2} \operatorname{Ldim}(\mathcal{H}) \cdot T \ln(T)}.$$

In [10] it is also shown that no algorithm (even allowing randomization) can achieve an expected bound better than $\sqrt{\frac{1}{8} \operatorname{Ldim}(\mathcal{H}) T}$. Closing the gap between the lower and upper bounds for the loss function (sometimes called *regret* in this context) is one of the main open problems mentioned in [10], where the authors remark that there are few known interesting examples of infinite classes with finite Littlestone dimension. Certainly, the model theory provides a large array of mathematically interesting examples of such classes which may be useful in providing examples which improve various bounds discussed above.

2.0.3 Bounded stochastic noise

Suppose that we work in the general setup from the previous section (again, not assuming realizability), but with a difference in the way we generate labels and measure mistakes. Suppose that there is a function $h \in \mathcal{H}$ such that the labels $y_1, \ldots, y_T$ are independent $\{0, 1\}$-valued random variables with the property that for all t, $Pr(h(x_t) \neq y_t) \leq \gamma$ with $\gamma \in (0, \frac{1}{2})$. This value γ will be called the noise rate.

In this setting, one seeks to minimize the difference between the predictions and the output of the noisy function on the samples:

$$E\left(\sum_{t=1}^{T}|\hat{y}_t - y_t|\right).$$

Note here that there are two sources of randomness—the choices of the algorithm may be randomized and the labels y_t are random variables. The expectation is taken with respect to both of these.

Theorem 2.0.9. *For any concept class $\mathcal{H}$, and any $\gamma \in [0, \frac{1}{2})$, there is an algorithm (possibly randomized) so that for any $h \in \mathcal{H}$, and a sequence of examples $(x_1, y_1), \ldots, (x_T, y_T)$ with each y_t a random variable as described above,*

$$E\left(\sum_{t=1}^{T}|\hat{y}_t - h(x_t)|\right) \leq \frac{\mathrm{Ldim}(\mathcal{H}) \cdot \ln(T)}{1 - 2\sqrt{\gamma(1-\gamma)}}.$$

That is, the expected number of mistakes grows only logarithmically in the sample size. In [10], the authors give an example of a class $\mathcal{H}$ which shows that the left-hand side of the inequality in the theorem is bounded below by $\Omega(\mathrm{Ldim}(\mathcal{H}) \cdot \ln(T))$.

CHAPTER 3

QUERY LEARNING

This chapter represents sections 2–4 along with a portion of section 1 of the preprint [13], co-authored with James Freitag.

3.1 Introduction

Fix a set X and denote by $\mathcal{P}(X)$ the collection of all subsets of X. A *concept class*[1] $\mathcal{C}$ on X is a subset of $\mathcal{P}(X)$. In the equivalence query (EQ) learning model, a learner attempts to identify a target set $A \in \mathcal{C}$ by means of a series of data requests called *equivalence queries*. The learner has full knowledge of $\mathcal{C}$, as well as a hypothesis class $\mathcal{H}$ with $\mathcal{C} \subseteq \mathcal{H} \subseteq \mathcal{P}(X)$. An *equivalence query* consists of the learner submitting a hypothesis $B \in \mathcal{H}$ to a teacher, who either returns *yes* if $A = B$, or a counterexample $x \in A \triangle B$. In the former case, the learner has learned A, and in the latter case, the learner uses the new information to update and submit a new hypothesis. In sections 3.2 and 3.3, the teacher may be assumed to be adversarial and the worst-case number of queries required to learn any concept is analyzed. In section 3.4, we consider the case in which the teacher selects counterexamples randomly according to a fixed but arbitrary distribution.

We will also consider learning with equivalence and membership queries (EQ+MQ). In a membership query, a learner submits a single element x from the base set X to the teacher, who

---

[1] We will also sometimes call $\mathcal{C}$ a set system on X.

returns the value $A(x)$, where A is the target concept. In this setting, the learner may choose to make either type of query at any stage, submitting any $x \in X$ for a membership query or submitting any $B \in \mathcal{H}$ for an equivalence query. The learner learns the target concept A when they submit A as an equivalence query.

With Theorems 3.2.6 and 3.2.24, we give upper bounds for the number of queries required for EQ and EQ+MQ learning a class $\mathcal{C}$ with hypotheses $\mathcal{H}$ in terms of the *Littlestone dimension of* $\mathcal{C}$, denoted $\mathrm{Ldim}(\mathcal{C})$, and the *consistency dimension of* $\mathcal{C}$ *with respect to* $\mathcal{H}$, denoted $\mathrm{C}(\mathcal{C}, \mathcal{H})$. We also give lower bounds for the number of required queries in terms of these quantities. In the EQ+MQ setting, the bounds are tight enough to completely characterize when a problem is efficiently learnable. Littlestone dimension is well-known in learning theory [25] and model theory.[1]

Consistency dimension and the related notion of strong consistency dimension are more subtle, which we detail in section 3.2. When $\mathcal{H}$ is taken to be $\mathcal{P}(X)$, $\mathrm{C}(\mathcal{C}, \mathcal{H}) = 1$; for various examples of set systems with $\mathcal{H} = \mathcal{C}$, one has $\mathrm{C}(\mathcal{C}, \mathcal{H}) = \infty$. In 3.2.2, we define a new invariant, the consistency threshold of $\mathcal{C}$, and provide a construction (for arbitrary $\mathcal{C}$) of a hypothesis class $\mathcal{H}$ which is not much more complicated than $\mathcal{C}$ (of the same Littlestone dimension as $\mathcal{C}$) such that $\mathrm{C}(\mathcal{C}, \mathcal{H}) \leq \mathrm{Ldim}(\mathcal{C}) + 1$. In 3.2.3, we compare our bounds and invariants to those previously appearing in the literature.

---

[1] In model theory, Littlestone dimension is called Shelah 2-rank, see [14] for additional details.

Theorems 3.2.6 and 3.2.24 can be used to establish efficient learnability in specific applied settings *if* one can obtain appropriate bounds on Littlestone dimension and consistency dimension. Let $(\mathcal{C}_n, \mathcal{H}_n)$ be a collection of concept and hypothesis classes which depends on some parameter n. Typically, we are thinking of finite classes which grow with n. We prove that whenever $\mathcal{C}_n$ can be learned by an algorithm using polynomially many membership queries and equivalence queries from $\mathcal{H}_n$, there must be polynomial bounds on Littlestone and consistency dimension. Moreover, whenever such an algorithm exists, the algorithm given in Theorem 3.2.24 accomplishes this.

Finally, to close section 3.2, we explain the connection between strong consistency dimension and a model theoretic property called the finite cover property (fcp), or rather its negation, referred to henceforth as the nfcp. We show that if $\mathcal{C}$ is the set system given by uniform instances of a fixed first order formula ϕ, and $\mathcal{H}$ is the collection of externally ϕ-definable sets, then $(\mathcal{C}, \mathcal{H})$ has finite strong consistency dimension if and only if ϕ has the nfcp.

In section 3.3 we demonstrate the practicality of our approach by providing simple and fast proofs of the efficient learnability of regular languages and certain ω-languages, reproving results of [1; 5; 17; 16]. Besides the conceptual simplicity of the approach, the bounds in learning complexity resulting from our algorithm have some novel aspects. For instance, our bounds have no dependence on the length of the strings provided to the learner as counterexamples, in contrast to existing algorithms.

In section 3.4 we turn to a randomized variant of EQ-learning in which the teacher is required to choose counterexamples randomly from a known probability distribution on X. [4]

show that for a concept class of size n, there is an algorithm in which the expected number of queries to learn any concept is at most $\log_2(n)$. It is natural to wonder whether there is a notion of dimension which can be used to bound the expected number of queries. In fact, Angluin and Dohrn [4, Theorem 25] already consider this, and show that the VC dimension of the concept class is a lower bound on the number of expected queries. However, [4, Theorem 26], using an example of [25], shows that the VC dimension *cannot* provide an upper bound for the number of queries. We show that the Littlestone dimension provides such an upper bound; we give an algorithm which yields a bound which is linear in the Littlestone dimension for the expected number of queries needed to learn any concept.

3.2 A combinatorial characterization of EQ-learnability

Often, one assumes that X is finite, and the emphasis is placed on finding bounds on the number of queries it may take to learn any $A \in \mathcal{C}$. We also consider the case where X is infinite, for which we give the following definition.

Definition 3.2.1. Let $\mathcal{C}$ and $\mathcal{H}$ be set systems on a set X. $\mathcal{C}$ is *learnable with equivalence queries* from $\mathcal{H}$ if there exists some $n < \omega$ and some algorithm to submit hypotheses from $\mathcal{H}$ such that any concept $A \in \mathcal{C}$ is learnable in at most n equivalence queries, given any teacher returning counterexamples. Let $\mathrm{LC}^{EQ}(\mathcal{C}, \mathcal{H})$ be the least such n if $\mathcal{C}$ is learnable with equivalence queries from $\mathcal{H}$, and $\mathrm{LC}^{EQ}(\mathcal{C}, \mathcal{H}) = \infty$ otherwise.

$\mathrm{LC}^{EQ}(\mathcal{C}, \mathcal{H})$ is called the *learning complexity*, representing the optimal number of queries needed in the worst-case scenario.

Similarly, $\mathcal{C}$ is learnable with equivalence queries from $\mathcal{H}$ and membership queries if there exists some $n < \omega$ and some algorithm to submit membership queries from X or equivalence queries from $\mathcal{H}$ such that any concept $A \in \mathcal{C}$ is learnable in at most n equivalence queries. The learning complexity is defined similarly and is denoted by $\mathrm{LC}^{EQ+MQ}(\mathcal{C}, \mathcal{H})$.

3.2.1 EQ-learnability from Littlestone and consistency dimension

Proposition 3.2.2. *[25, Theorems 5 and 6] If* $\mathrm{LC}^{EQ}(\mathcal{C}, \mathcal{H}) \leq d + 1$, *then* $\mathrm{Ldim}(\mathcal{C}) \leq d$. *If* $\mathcal{H} = \mathcal{P}(X)$, *then the converse holds.*

Proof. Suppose $\mathrm{Ldim}(\mathcal{C}) \geq d + 1$. We show that we can force the learner to use at least $d + 2$ equivalence queries. Construct a binary element tree of height $d + 1$ with proper labels from $\mathcal{C}$ witnessing $\mathrm{Ldim}(\mathcal{C}) \geq d+1$. Given the first hypothesis H_0 from the learner, return the element on the 0th level on the tree as a counterexample. Continue this, returning the element on the ith level along the path consistent with previous counterexamples as the counterexample to hypothesis H_i. We will return $d + 1$ counterexamples, and the learner still requires one more hypothesis to identify the concept. Since this will occur for one of the proper labels A of the binary element tree, we have forced the learner to use at least $d+2$ equivalence queries for some $A \in \mathcal{C}$.

Suppose $\mathrm{Ldim}(\mathcal{C}) = d < \infty$. Let $\mathcal{C}_0 = \mathcal{C}$. Inductively define $\mathcal{C}_i$, $i = 1, \ldots, d$ as follows. Given $\mathcal{C}_i$, for any $x \in X$ and $j \in \{0, 1\}$, let

$$\mathcal{C}_i^{(x,j)} := \{A \in \mathcal{C}_i \mid \chi_A(x) = j\},$$

where χ_A is the characteristic function on A. Let

$$B_i := \{x \in X \mid \mathrm{Ldim}(\mathcal{C}_i^{(x,1)}) \geq \mathrm{Ldim}(\mathcal{C}_i^{(x,0)})\}.$$

Submit B_i as the hypothesis. If B_i is correct, we are done. Otherwise, we receive a counterexample x_i. Set

$$\mathcal{C}_{i+1} := \{A \in V_i \mid \chi_A(x_i) \neq \chi_{B_i}(x_i)\}$$

to be the concepts which have the correct label for x_i. Observe that at each stage, $\mathrm{Ldim}(\mathcal{C}_{i+1}) < \mathrm{Ldim}(\mathcal{C}_i)$. Therefore, if we make d queries without correctly identifying the target, then we must have $\mathrm{Ldim}(\mathcal{C}_d) = 0$. Then V_d is a singleton, which must be the target concept. □

Notice in particular that if $\mathrm{Ldim}(\mathcal{C}) = \infty$, then $\mathcal{C}$ cannot be learned with equivalence queries, even with $\mathcal{H} = \mathcal{P}(X)$. The assumption that $\mathcal{H} = \mathcal{P}(X)$ makes learning straightforward, but this may be too strong for many settings. However, without some additional hypotheses on $\mathcal{H}$, learnability may already be hopeless, even for *very simple* set systems. For instance, let $\mathcal{C}$ be the set of singletons. If $\mathcal{H} = \mathcal{C}$, then we may take as long as $|X|$ to learn if X is finite, or never learn at all if X is infinite. However, if the learner is allowed to guess $\emptyset$, this forces the teacher to identify the target singleton.

The strategy of Proposition 3.2.2 permeates both learnability and non-learnability proofs; identifying a specific set amounts to reducing the Littlestone dimension of the family of possible concepts to 0; actually submitting the target concept before the Littlestone dimension reaches 0 can be thought of as a best-case scenario that we cannot rely on. Non-learnability then

amounts to an inability to reduce the Littlestone dimension of the family of possible concepts to 0 through a series of finitely many equivalence queries. The main purpose of this section is to give precise conditions on $\mathcal{H}$ and $\mathcal{C}$ which *characterize* learnability.

Definition 3.2.3. Given a set X, a *partially specified subset* A of X is a partial function $A : X \to \{0, 1\}$.

- Say $x \in A$ if $A(x) = 1$, $x \notin A$ if $A(X) = 0$, and membership of x is unspecified otherwise. The *domain* of A, $\mathrm{dom}(A)$, is $A^{-1}(\{0, 1\})$. Call A *total* if $\mathrm{dom}(A) = X$. We identify subsets $A \subseteq X$ with total partially specified subsets. The *size* of A, $|A|$, is the cardinality of $\mathrm{dom}(A)$.
- Given two partially specified subsets A and B, write $A \sqsubseteq B$ if A and B agree on $\mathrm{dom}(A)$; call A a *restriction* of B and B an *extension* of A.
- Given a set $Y \subseteq \mathrm{dom}(A)$, the restriction $A|_Y$ of A to Y is the partial function where $A|_Y(x) = A(x)$ for all $x \in Y$, and is unspecified otherwise.
- Given a set system $\mathcal{C}$ on X, A is *n-consistent* with $\mathcal{C}$ if every size n restriction of A has an extension in $\mathcal{C}$. Otherwise, say A is *n-inconsistent*. A is *finitely consistent* with $\mathcal{C}$ if every restriction of A of finite size has an extension in $\mathcal{C}$—that is, A is n-consistent with $\mathcal{C}$ for all $n < \omega$.

The following definition is a translation into set systems of a definition that first appeared in [8].

Definition 3.2.4. The *consistency dimension* of $\mathcal{C}$ with respect to $\mathcal{H}$, denoted $\mathrm{C}(\mathcal{C}, \mathcal{H})$, is the least integer n such that for every subset $A \subseteq X$ (viewed as a total partially specified subset), if A is n-consistent with $\mathcal{C}$, then $A \in \mathcal{H}$. If no such n exists, then say $\mathrm{C}(\mathcal{C}, \mathcal{H}) = \infty$.

Observe that $\mathrm{C}(\mathcal{C}, \mathcal{H}) = 1$ iff $\mathcal{H}$ shatters[1] the set of all elements $x \in X$ such that there are A_0 and A_1 in $\mathcal{C}$ such that $x \notin A_0$ but $x \in A_1$. In this case, it is possible to learn any concept in $\mathcal{C}$ in at most $\mathrm{Ldim}(\mathcal{C}) + 1$ equivalence queries, using the method of Proposition 3.2.2. So we may assume that $\mathrm{C}(\mathcal{C}, \mathcal{H}) > 1$.

Lemma 3.2.5. *Suppose that for each $i < n$, $\mathcal{C}_i$ is a concept class on X and $\mathcal{H}_i$ is a hypothesis class on X. Suppose that $\mathrm{LC}^{EQ}(\mathcal{C}_i, \mathcal{H}_i) = m_i$. Then $\mathrm{LC}^{EQ}(\mathcal{C}, \mathcal{H}) \leq \sum_{i<n} m_i$, where $\mathcal{C} := \cup_{i<n} \mathcal{C}_i$ and $\mathcal{H} := \cup_{i<n} \mathcal{H}_i$.*

Proof. We give the proof for $n = 2$; then the result for $n > 2$ follows easily by induction.

To learn a target concept $A \in \mathcal{C} = \mathcal{C}_0 \cup \mathcal{C}_1$ with hypotheses from $\mathcal{H} = \mathcal{H}_0 \cup \mathcal{H}_1$, begin by assuming that $A \in \mathcal{C}_0$. Attempt to learn A by making guesses from $\mathcal{H}_0$, according to the procedure by which any concept in $\mathcal{C}_0$ is learnable in at most m_0 many queries. If, after making m_0 many queries, we have failed to learn A, then we conclude that $A \notin \mathcal{C}_0$, whence $A \in \mathcal{C}_1$. We can then learn A in at most m_1 many additional queries with guesses from $\mathcal{H}_1$. □

We can now give an upper bound for the learning complexity in terms of Littlestone dimension and consistency dimension.

---

[1]Recall that a set system $\mathcal{C}$ shatters a set A if, for all $B \subseteq A$, there is $C \in \mathcal{C}$ such that $C \cap A = B$.

Theorem 3.2.6. *Suppose* $\mathrm{Ldim}(\mathcal{C}) = d < \infty$ *and* $1 < \mathrm{C}(\mathcal{C}, \mathcal{H}) = c < \infty$. *Then* $\mathrm{LC}^{EQ}(\mathcal{C}, \mathcal{H}) \leq c^d$.

Proof. We proceed by induction on d. The base case, $d = 0$, is trivial, as then $\mathcal{C}$ is a singleton.

Suppose there is some element x such that $\mathrm{Ldim}(\mathcal{C} \cap x) < d + 1$ and $\mathrm{Ldim}(\mathcal{C} \setminus x) < d + 1$, where $\mathcal{C} \cap x := \{A \in \mathcal{C} \,|\, x \in A\}$ and $\mathcal{C} \setminus x := \{A \in \mathcal{C} \,|\, x \notin A\}$. Then by induction, any concept in $\mathcal{C} \cap x$ can be learned in at most c^d queries with guesses from $\mathcal{H}$, and the same is true for $\mathcal{C} \setminus x$. Then by Lemma 3.2.5, any concept in $\mathcal{C}$ can be learned in at most $2c^d \leq c^{d+1}$ equivalence queries.

If no such x exists, then for all x, either $\mathrm{Ldim}(\mathcal{C} \cap x) = d + 1$ or $\mathrm{Ldim}(\mathcal{C} \setminus x) = d + 1$. Let B be such that $x \in B$ iff $\mathrm{Ldim}(\mathcal{C} \cap x) = d + 1$.

If $B \in \mathcal{H}$, then we submit B as our query. If we are incorrect, then by choice of B, the class $\mathcal{C}'$ of concepts consistent with the counterexample x_0 will have Littlestone dimension $\leq d$. By induction, any concept in $\mathcal{C}'$ can be learned in at most c^d many queries, and so we learn a in at most $c^d + 1 \leq c^{d+1}$ queries.

If $B \notin H$, then, since $\mathrm{C}(\mathcal{C}, \mathcal{H}) = c$, there are some $x_0, \ldots, x_{c-1}$ such that there is no $A \in \mathcal{C}$ such that $B|_{\{x_0,\ldots,x_{c-1}\}} \sqsubseteq A$. Then, with notation as in the proof of Proposition 3.2.2,

$$\mathcal{C} = (\mathcal{C}^{(x_0, 1-B(x_0))}) \cup \ldots \cup (\mathcal{C}^{(x_{c-1}, 1-B(x_{c-1}))}),$$

and $\mathrm{Ldim}(\mathcal{C}^{(x_i,1-B(x_i))}) \leq d$ for each i. Then, by induction, for each i, any concept in $\mathcal{C}^{(x_i,1-B(x_i))}$ can be learned in at most c^d many queries with guesses from $\mathcal{H}$. By Lemma 3.2.5, any concept in $\mathcal{C}$ can be learned in at most c^{d+1} many queries with guesses from $\mathcal{H}$. □

On the other hand, Proposition 3.2.2 gives a lower bound of $\mathrm{Ldim}(\mathcal{C}) + 1 \leq \mathrm{LC}^{EQ}(\mathcal{C}, \mathcal{H})$. There is also a lower bound for learning complexity in terms of consistency dimension:

Proposition 3.2.7. *[8, Theorem 2] Suppose there is some partially specified subset A which is n-consistent with $\mathcal{C}$ but which does not have a total extension in $\mathcal{H}$. Then $n < \mathrm{LC}^{EQ}(\mathcal{C}, \mathcal{H})$.*

Proof. By hypothesis, given any equivalence query H, the teacher can find some $x \in \mathrm{dom}(A)$ such that $H(x) \neq A(x)$. Moreover, since A is n-consistent with $\mathcal{C}$, the teacher is able to return a counterexample of this form for the first n equivalence queries. Thus $\mathcal{C}$ cannot be learned with fewer than $n + 1$ equivalence queries from $\mathcal{H}$. □

In particular, if $\mathrm{C}(\mathcal{C}, \mathcal{H}) \geq c$, then there is some subset A which is $(c-1)$-consistent with $\mathcal{C}$ but which does not belong to $\mathcal{H}$. Then $c \leq \mathrm{LC}^{EQ}(\mathcal{C}, \mathcal{H})$. So $\mathrm{C}(\mathcal{C}, \mathcal{H}) \leq \mathrm{LC}^{EQ}(\mathcal{C}, \mathcal{H})$. In fact, we will obtain a stronger bound using strong consistency dimension in section 3.2.3.

Furthermore, if $\mathrm{C}(\mathcal{C}, \mathcal{H}) = \infty$, then $\mathcal{C}$ cannot be learned with equivalence queries from $\mathcal{H}$. Combining Theorem 3.2.6 and Propositions 3.2.2 and 3.2.7, we obtain the following:

Theorem 3.2.8. *$\mathcal{C}$ is learnable with equivalence queries from $\mathcal{H}$ iff $\mathrm{Ldim}(\mathcal{C}) < \infty$ and $\mathrm{C}(\mathcal{C}, \mathcal{H}) < \infty$.*

3.2.2 Obtaining finite consistency dimension

We have established that finite consistency dimension is essential for EQ-learning. The central question we answer in this subsection is: given $\mathcal{C}$, can one obtain a hypothesis class $\mathcal{H}$ which is not much more complicated than $\mathcal{C}$ with the property that $\mathrm{C}(\mathcal{C}, \mathcal{H})$ is finite?

Definition 3.2.9. Fix a set system $\mathcal{C}$ on a set X. $\mathcal{C}$ has *consistency threshold* $n < \infty$ if, given any hypothesis class $\mathcal{H} \supset \mathcal{C}$, we have that

$$\mathrm{C}(\mathcal{C}, \mathcal{H}) < \infty \quad \text{iff} \quad \mathrm{C}(\mathcal{C}, \mathcal{H}) \leq n.$$

Lemma 3.2.10. *Suppose A is a partially specified subset finitely consistent with $\mathcal{C}$. Then there is a total extension $A' \sqsupseteq A$ finitely consistent with $\mathcal{C}$.*

Proof. Let $X = \{x_\alpha \,|\, \alpha < |X|\}$ be a well-ordering of X. Let $A_0 = A$. We inductively define a $\sqsubseteq$-chain of partially specified subsets A_α, where each A_α is defined on $\mathrm{dom}(A) \cup \{x_\xi \,|\, \xi < \alpha\}$ and is finitely consistent with $\mathcal{C}$. For α a limit ordinal, set $A_\alpha = \cup_{\xi<\alpha} A_\xi$. It is clear that A_α is finitely consistent with $\mathcal{C}$ if all A_ξ for $\xi < \alpha$ are.

At any successor stage $\alpha + 1$, if $x_\alpha \in \mathrm{dom}(A_\alpha)$, set $A_{\alpha+1} = A_\alpha$. Otherwise, we must extend A_α to x_α while remaining finitely consistent with $\mathcal{C}$. Assume for contradiction that neither $B_0 := A_\alpha \cup \{x_\alpha \mapsto 0\}$ nor $B_1 := A_\alpha \cup \{x_\alpha \mapsto 1\}$ are finitely consistent with $\mathcal{C}$. Then there are finite sets $Y_0, Y_1 \subseteq \mathrm{dom}(A_\alpha)$ such that $B_0|_{Y_0 \cup \{a_\alpha\}}$ and $B_1|_{Y_1 \cup \{a_\alpha\}}$ have no extension in $\mathcal{C}$. But $A_\alpha|_{Y_0 \cup Y_1}$ has an extension B in $\mathcal{C}$, and B must be an extension of either $B_0|_{Y_0 \cup \{a_\alpha\}}$ or

$B_1|_{Y_1 \cup \{a_\alpha\}}$, a contradiction. So A_α has a finitely consistent extension to x_α, and we set $A_{\alpha+1}$ to be such an extension.

We then take $A' = \cup_{\xi<|X|} A_\xi$. □

Proposition 3.2.11. *Let $\mathcal{C}, \mathcal{H}$ be set systems and let A be a partially specified subset. The following are equivalent:*

(i) *A is finitely consistent with $\mathcal{C}$.*

(ii) *If $\mathrm{C}(\mathcal{C}, \mathcal{H}) < \infty$, then there is a total extension $A' \sqsupseteq A$ in $\mathcal{H}$.*

Proof. (i) ⇒ (ii): Let $A' \sqsupseteq A$ be a total extension finitely consistent with $\mathcal{C}$. If $\mathrm{C}(\mathcal{C}, \mathcal{H}) < \infty$, then $A' \in \mathcal{H}$.

(ii) ⇒ (i): We show the contrapositive. Suppose that A is not finitely consistent with $\mathcal{C}$, witnessed by some size n restriction A_0, which is a $\sqsubseteq$-minimal such restriction. We find some $\mathcal{H}$ such that $\mathrm{C}(\mathcal{C}, \mathcal{H}) < \infty$ but $\mathcal{H}$ contains no total extension of A. Let $\mathcal{H}$ be the collection of all (total partially specified) subsets which are not extensions of A_0. So A has no total extension in $\mathcal{H}$. We claim that $\mathrm{C}(\mathcal{C}, \mathcal{H}) \leq n$. Indeed, observe that given any (total partially specified) subset B that is n-consistent with $\mathcal{C}$, we have $A_0 \not\sqsubseteq B$, and then $B \in \mathcal{H}$.

□

In particular, if $\mathrm{C}(\mathcal{C}, \mathcal{H}) < \infty$, then $\mathcal{H}$ contains all finitely consistent subsets. That is, extensions of all finitely consistent partially specified subsets (equivalently, by Lemma 3.2.10, all finitely consistent total partially specified subsets) are necessary to obtain $\mathrm{C}(\mathcal{C}, \mathcal{H}) < \infty$. Consistency threshold classifies when this is a sufficient condition.

Proposition 3.2.12. *The following are equivalent:*

(i) *$\mathcal{C}$ has consistency threshold $\leq n < \infty$.*

(ii) *For all (total partially specified) subsets A, if A is n-consistent with $\mathcal{C}$, then A is finitely consistent with $\mathcal{C}$.*

(iii) *If $\mathcal{H}$ contains all finitely consistent (total partially specified) subsets, then $\mathrm{C}(\mathcal{C}, \mathcal{H}) \leq n$.*

Proof. (i) $\Rightarrow$ (ii): Assume for contradiction that there is some total A which is n-consistent but not finitely consistent. Let m be minimal such that A is m-inconsistent. Then there is a size m restriction $A' \sqsubseteq A$ that has no extension in $\mathcal{C}$. Then let $\mathcal{H}$ contain all subsets which do not extend A'.

We claim that $\mathrm{C}(\mathcal{C}, \mathcal{H}) = m$. Note that A witnesses that $\mathrm{C}(\mathcal{C}, \mathcal{H}) \geq m$. On the other hand, observe that given any partially specified subset B that is m-consistent with $\mathcal{C}$, we have $A' \not\sqsubseteq B$, and then it is easy to see that B has a total extension in $\mathcal{H}$.

(ii) $\Rightarrow$ (iii): If $\mathcal{H}$ contains all finitely consistent subsets, and all n-consistent subsets are finitely consistent, then $\mathrm{C}(\mathcal{C}, \mathcal{H}) \leq n$ holds immediately.

(iii) $\Rightarrow$ (i): By Proposition 3.2.11, if $\mathrm{C}(\mathcal{C}, \mathcal{H}) < \infty$, then $\mathcal{H}$ already has all finitely consistent subsets. Then $\mathrm{C}(\mathcal{C}, \mathcal{H}) \leq n$. □

In particular, if $\mathcal{C}$ has finite consistency threshold, then $\mathrm{C}(\mathcal{C}, \mathcal{H}) < \infty$ iff $\mathcal{H}$ contains all finitely consistent subsets.

Corollary 3.2.13. *Suppose $\mathcal{C}$ does not have finite consistency threshold. Then for arbitrarily large n, there is some total subset A_n which is n-consistent but not $(n+1)$-consistent with $\mathcal{C}$.*

Finite consistency threshold is not strictly necessary to provide a positive answer to the central question of this subsection; nevertheless, it does identify a clear qualitative dividing line. When $\mathcal{C}$ has finite consistency threshold, $\mathcal{H}$ only needs to contain all finitely consistent subsets; letting $\mathcal{H}_\infty$ be the set of all finitely consistent subsets, we obtain a minimum hypothesis class such that learning is possible.

Where $\mathcal{C}$ does not have finite consistency threshold, more is required; we must add some hypotheses which are inconsistent with the concepts in $\mathcal{C}$, and there is no minimal $\mathcal{H}$ such that learning is possible. However, for each m, we can replace "finitely consistent" with "m-consistent" to obtain a class $\mathcal{H}_m$ such that $\mathrm{C}(\mathcal{C}, \mathcal{H}_m) \leq m$—let $\mathcal{H}_m$ be the collection of all subsets which are m-consistent with $\mathcal{C}$. Note that $\mathcal{H}_m$ is clearly the minimum hypothesis class such that $\mathrm{C}(\mathcal{C}, \mathcal{H}) \leq m$.

Note that for all m, $\mathcal{H}_\infty \subseteq \mathcal{H}_m$. By Proposition 3.2.12, if $\mathcal{C}$ has consistency threshold n, then for all $m \geq n$, $\mathcal{H}_m = \mathcal{H}_n = \mathcal{H}_\infty$. If $\mathcal{C}$ does not have finite consistency threshold, there is no minimal $\mathcal{H}$ such that $\mathrm{C}(\mathcal{C}, \mathcal{H}) < \infty$; by Corollary 3.2.13, if $\mathrm{C}(\mathcal{C}, \mathcal{H}) = m$, then there is $m' \geq m$ such that $\mathcal{H}_{m'} \subsetneq \mathcal{H}$.

By choosing m appropriately, given any $\mathcal{C}$, we can find a hypothesis class such that $\mathrm{C}(\mathcal{C}, \mathcal{H}) < \infty$ without increasing the Littlestone dimension; that is, $\mathrm{Ldim}(\mathcal{H}) = \mathrm{Ldim}(\mathcal{C})$.

Theorem 3.2.14. *Suppose* $\mathrm{Ldim}(\mathcal{C}) = d < \infty$. *Then there is* $\mathcal{H}$ *such that* $\mathrm{C}(\mathcal{C}, \mathcal{H}) < \infty$ *and* $\mathrm{Ldim}(\mathcal{H}) = \mathrm{Ldim}(\mathcal{C})$. *Furthermore, we can find such an* $\mathcal{H}$ *such that* $\mathrm{C}(\mathcal{C}, \mathcal{H}) \leq \mathrm{Ldim}(\mathcal{C}) + 1$.

Proof. Fix some $m > d = \mathrm{Ldim}(\mathcal{C})$. Let $\mathcal{H}_m$ be the collection of all subsets which are m-consistent with $\mathcal{C}$. It is immediate that $\mathrm{C}(\mathcal{C}, \mathcal{H}_m) \leq m < \infty$.

Assume for contradiction that $\mathrm{Ldim}(\mathcal{H}_m) > \mathrm{Ldim}(\mathcal{C})$. Consider a binary element tree of height $d+1$ that can be properly labeled with elements of $\mathcal{H}_m$; in particular, there is some leaf which cannot be labeled with an element of $\mathcal{C}$. Consider such a leaf. The path through the binary element tree to this leaf defines a partially specified subset A that is $(d+1)$-inconsistent with $\mathcal{C}$. In particular, any total extension is $(d+1)$-inconsistent, so m-inconsistent, and so does not belong to $\mathcal{H}_m$. This contradicts our ability to label the leaf with an element of $\mathcal{H}_m$.

In particular, recall that when $\mathcal{C}$ has finite consistency threshold n, A is n-consistent with $\mathcal{C}$ iff it is finitely consistent with $\mathcal{C}$. So setting $\mathcal{H}_m$ as above with m at least the finite consistency threshold amounts to setting $\mathcal{H}_m$ to be the collection of all finitely consistent partially specified subsets. In this case, $\mathrm{Ldim}(\mathcal{H}_m) = \mathrm{Ldim}(\mathcal{C})$ even if $m \leq d$, as increasing the Littlestone dimension requires adding something inconsistent with $\mathcal{C}$.

Regardless of whether $\mathcal{C}$ has finite consistency dimension, we can let $m = d+1$. Then $\mathrm{C}(\mathcal{C}, \mathcal{H}_m) \leq m = d+1$. □

3.2.3 From consistency to strong consistency

From an algorithms perspective, the result of Theorem 3.2.6 is unsatisfactory, since it is exponential in $\mathrm{Ldim}(\mathcal{C})$. We give an example to show that, without modification, we cannot expect a significant improvement.

Example 3.2.15. Fix $c > 2$ and d. Let $\{a_\tau \mid \tau \in [c]^i, 1 \leq i \leq d\}$ be distinct elements indexed by finite nonempty sequences of length at most d from $[c]$. For $\sigma \in [c]^d$, let $B_\sigma = \{a_\tau \mid \tau \subseteq \sigma\}$. Let $\mathcal{C} = \{B_\sigma \mid \sigma \in [c]^d\}$. Then $\mathrm{Ldim}(\mathcal{C}) = d$.

If we take $\mathcal{C}$ to also be our hypothesis class, then $\mathrm{C}(\mathcal{C},\mathcal{C}) = c+1$. Indeed, the (total partially specified) subset $A = \{a_0\}$ is c-consistent but not $(c+1)$ consistent with $\mathcal{C}$, witnessed by the restriction of A to $\{a_0, a_{0,0}, \ldots, a_{0,c-1}\}$, so $\mathrm{C}(\mathcal{C},\mathcal{C}) \geq c+1$. On the other hand, if A is a subset $(c+1)$-consistent with $\mathcal{C}$, then, by induction on the length of τ, for each $1 \leq i \leq d$, A contains exactly one a_τ with $\tau = i$, so $A \in \mathcal{C}$.

However, it may take as long as c^d many equivalence queries to learn; if the teacher returns a_σ as a counterexample to hypothesis A_σ, then the learner can only eliminate A_σ.

The most promising modification is the following variant of consistency dimension, which also appeared in [8] in a slightly different form.

Definition 3.2.16. The *strong consistency dimension* of $\mathcal{C}$ with respect to $\mathcal{H}$, denoted $\mathrm{SC}(\mathcal{C},\mathcal{H})$, is the least integer n such that for every partially specified subset A, if A is n-consistent with $\mathcal{C}$, then A has an extension in $\mathcal{H}$. If no such n exists, then say $\mathrm{SC}(\mathcal{C},\mathcal{H}) = \infty$.

We therefore make the stronger requirement that all partially specified subsets that are n-consistent be consistent, rather than just all totally partially specified subsets. It is immediate from the definition that $\mathrm{C}(\mathcal{C},\mathcal{H}) \leq \mathrm{SC}(\mathcal{C},\mathcal{H})$. At the smallest levels, consistency dimension and strong consistency dimension are equal.

Proposition 3.2.17. *If* $\mathrm{C}(\mathcal{C},\mathcal{H}) = 1$*, then* $\mathrm{SC}(\mathcal{C},\mathcal{H}) = 1$*. If* $\mathrm{C}(\mathcal{C},\mathcal{H}) = 2$*, then* $\mathrm{SC}(\mathcal{C},\mathcal{H}) = 2$*.*

Proof. Observe that $\mathrm{C}(\mathcal{C},\mathcal{H}) = 1$ iff $\mathrm{SC}(\mathcal{C},\mathcal{H}) = 1$ iff $\mathcal{H}$ shatters the set of all elements $x \in X$ such that there are A_0 and A_1 in $\mathcal{C}$ such that $x \notin A_0$ but $x \in A_1$.

Suppose that $\mathrm{C}(\mathcal{C}, \mathcal{H}) = 2$. Let A be a partially specified subset that is 2-consistent with $\mathcal{C}$. We wish to find a total extension of A in $\mathcal{H}$. It suffices to find a total extension $B \sqsupseteq A$ that is 2-consistent with $\mathcal{C}$.

Let $X = \{x_\alpha \,|\, \alpha < |X|\}$ be a well-ordering of X. Let $A_0 = A$. We inductively define a $\sqsubseteq$-chain of partially specified subsets A_α, where each A_α is defined on $\mathrm{dom}(A) \cup \{x_\xi \,|\, \xi < \alpha\}$ and is 2-consistent with $\mathcal{C}$. For α a limit ordinal, set $A_\alpha = \cup_{\xi<\alpha} A_\xi$. It is clear that A_α is 2-consistent with $\mathcal{C}$ if all A_ξ for $\xi < \alpha$ are.

At any successor stage $\alpha + 1$, if $x_\alpha \in \mathrm{dom}(A_\alpha)$, set $A_{\alpha+1} = A_\alpha$. Otherwise, we must extend A_α to x_α while remaining 2-consistent with $\mathcal{C}$. Assume for contradiction that neither $B_0 := A_\alpha \cup \{x_\alpha \mapsto 0\}$ nor $B_1 := A_\alpha \cup \{x_\alpha \mapsto 1\}$ are 2-consistent with $\mathcal{C}$. Then there are y_0, $y_1 \in \mathrm{dom}(A_\alpha)$ such that $B_0|_{\{y_0,x_\alpha\}}$ and $B_1|_{\{y_1,x_\alpha\}}$ have no extension in $\mathcal{C}$. But $A_\alpha|_{\{y_0,y_1\}}$ has an extension B in $\mathcal{C}$, and B must be an extension of either $B_0|_{\{y_0,x_\alpha\}}$ or $B_1|_{\{y_1,x_\alpha\}}$, a contradiction. So A_α has a 2-consistent extension to x_α, and we set $A_{\alpha+1}$ to be such an extension.

We then take $\cup_{\xi<|X|} A_\xi$ to be our total extension. □

As the following examples show, consistency dimension and strong consistency dimension may differ when $\mathrm{C}(\mathcal{C}, \mathcal{H}) \geq 3$.

Example 3.2.18. Let $X = \{a, b, c, d, e\}$. Let

$$\mathcal{C} = \mathcal{H} = \{\{a, b, c\}, \{a, b, d\}, \{a, c, d, e\}, \{b, c, d, e\}\}.$$

One can verify that $\mathrm{C}(\mathcal{C}, \mathcal{H}) = 3$, but the partially specified subset $\{a, b, c, d\}$ with e unspecified witnesses that $\mathrm{SC}(\mathcal{C}, \mathcal{H}) > 3$.

Example 3.2.19. Continuing Example 3.2.15, observe that $\mathrm{SC}(\mathcal{C}, \mathcal{C}) = c^d$. In particular, the partially specified subset A' given by

$$A'(a_\tau) = \begin{cases} 0 & |\tau| = d \\ \text{undefined} & \text{otherwise} \end{cases}$$

witnesses that $\mathrm{SC}(\mathcal{C}, \mathcal{C}) > c^d - 1$. Then we learn in at most $\mathrm{SC}(\mathcal{C}, \mathcal{C})$ many queries. Moreover, this demonstrates that consistency dimension and strong consistency dimension can differ by an arbitrarily large amount (allowing $\mathrm{Ldim}(\mathcal{C})$ to vary), and that strong consistency dimension may even be exponentially larger than consistency dimension.

Strong consistency dimension, like consistency dimension, categorizes equivalence query learning:

Theorem 3.2.20. *$\mathcal{C}$ is learnable with equivalence queries from $\mathcal{H}$ iff $\mathrm{Ldim}(\mathcal{C}) \leq \infty$ and $\mathrm{SC}(\mathcal{C}, \mathcal{H}) < \infty$. In particular, $\mathrm{SC}(\mathcal{C}, \mathcal{H}) \leq \mathrm{LC}^{EQ}(\mathcal{C}, \mathcal{H})$.*

Proof. For the reverse direction, use Theorem 3.2.6 and the observation that $\mathrm{C}(\mathcal{C}, \mathcal{H}) \leq \mathrm{SC}(\mathcal{C}, \mathcal{H})$.

For the forward direction, use Propositions 3.2.2 and 3.2.7. In particular, if $\mathrm{SC}(\mathcal{C}) \geq c$, then there is a partially specified subset A that is $(c-1)$-consistent with $\mathcal{C}$ but which has no total extension in $\mathcal{H}$. Then, by Proposition 3.2.7, $c \leq \mathrm{LC}^{EQ}(\mathcal{C}, \mathcal{H})$. □

Corollary 3.2.21. *Suppose $\mathrm{Ldim}(\mathcal{C}) < \infty$. Then $\mathrm{C}(\mathcal{C}, \mathcal{H}) < \infty$ iff $\mathrm{SC}(\mathcal{C}, \mathcal{H}) < \infty$.*

The distinction between consistency dimension and strong consistency dimension is subtle, and many previous results hold with little to no modification if one replaces consistency dimension with strong consistency dimension. On the other hand, our work in section 3.3 will reveal the practical difficulties associated with strong consistency dimension in complicated concept classes.

We have already seen in Theorem 3.2.20 that strong consistency dimension provides a better lower bound for learning complexity. It is also known in the finite case that strong consistency dimension also gives a stronger upper bound for learning complexity:

Theorem 3.2.22. *[8, Theorem 2] Suppose $\mathcal{C}$ is finite. Then* $\mathrm{LC}^{EQ}(\mathcal{C},\mathcal{H}) \leq \lceil SC(\mathcal{C},\mathcal{H}) \cdot \ln|\mathcal{C}| \rceil$.

Proof. As this was originally framed in the setting where concepts were represented by strings, we give an abbreviated translation of the original proof into the language of set systems. This proof demonstrates the utility of constructing a partial hypothesis and taking some complete extension.

Let $c = \mathrm{SC}(\mathcal{C},\mathcal{H})$. At stage i, let $\mathcal{C}_i \subseteq \mathcal{C}$ be the set of remaining possible target concepts. Let A_i be the partially specified subset given by

$$A(x) = \begin{cases} 1 & x \text{ belongs to more than } \frac{c-1}{c}|\mathcal{C}_i| \text{ many } C \in \mathcal{C}_i \\ 0 & x \text{ belongs to less than } \frac{1}{c}|\mathcal{C}_i| \text{ many } C \in \mathcal{C}_i \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Observe that A is c-consistent with $\mathcal{C}$—given any $Y := \{x_0, \ldots, x_{c-1}\} \subseteq \mathrm{dom}(A)$, for each j, less than $\frac{1}{c}|\mathcal{C}_i|$ many remaining concepts disagree with A on x_j, so less than $c\frac{1}{c}|\mathcal{C}_i| = |\mathcal{C}_i|$ many concepts disagree with A on some x_j. So some concept agrees with A on Y. So A is c-consistent.

So we can find some $B \in \mathcal{H}$ such that $B \sqsupseteq A$, and we submit B as our hypothesis. By choice of A, if we receive a counterexample, we will have $|\mathcal{C}_{i+1}| \leq \frac{c-1}{c}|\mathcal{C}_i|$. Repeating this $\lceil c \cdot \ln |\mathcal{C}| \rceil$ many times is enough to identify and submit the target concept. □

In light of Example 3.2.19, one hopes that improved bounds on learning can be found in terms of strong consistency dimension and Littlestone dimension when $\mathcal{C}$ is infinite. We are unable to show this presently, but offer some evidence in this direction:

Proposition 3.2.23. *Suppose* $\mathrm{Ldim}(\mathcal{C}) = d < \infty$ *and* $\mathrm{SC}(\mathcal{C}, \mathcal{H}) = 2 < \infty$. *Then* $\mathrm{LC}^{EQ}(\mathcal{C}, \mathcal{H}) = d + 1$.

Proof. We know by Proposition 3.2.2 that $d + 1$ is a lower bound. We show that it is also an upper bound.

Let $V_0 = \mathcal{C}$. Inductively define V_i, $i = 1, \ldots, d$ as follows. Given V_i, for any $x \in X$ and $j \in \{0, 1\}$, let

$$V_i^{(x,j)} := \{B \in V_i \mid \chi_B(x) = j\},$$

where χ_B is the characteristic function on B. Construct the partially specified subset A_i where

$$A_i(x) = \begin{cases} 0 & \text{Ldim}(V_i^{(x,0)}) = \text{Ldim}(V_i) \\ 1 & \text{Ldim}(V_i^{(x,1)}) = \text{Ldim}(V_i) \\ \text{undefined} & \text{otherwise.} \end{cases} \tag{3.1}$$

We claim that A_i has an extension in H. By our assumption that $\text{SC}(\mathcal{C}, \mathcal{H}) = 2$, it suffices to check that A is 2-consistent with V_i. Suppose for contradiction that there are $a_0, a_1 \in \text{dom}(A_i)$ such that, without loss of generality, $A_i(a_0) = A_i(a_1) = 0$, but there is no extension of $A_i|_{\{a_0,a_1\}}$ in V_i. Then observe that $V_i^{(x_0,0)} \subseteq V_i^{(x_1,1)}$, whence

$$\text{Ldim}(V_i) \geq \text{Ldim}(V_i^{(x_1,1)}) \geq \text{Ldim}(V_i^{(x_0,0)}) = \text{Ldim}(V_i),$$

so $\text{Ldim}(V_i^{(x_1,1)}) = \text{Ldim}(V_i)$. But we also have $\text{Ldim}(V_i^{(x_1,0)}) = \text{Ldim}(V_i)$, a contradiction, as we could then construct a binary element tree with proper labels from V_i of height $\text{Ldim}(V_i)+1$ with x_1 at the root.

Let $B_i \in \mathcal{H}$ be a total extension of A_i. Submit B_i as the hypothesis. If B_i is correct, we are done. Otherwise, we receive a counterexample x_i. Set

$$V_{i+1} := \{B \in V_i \,|\, \chi_B(x_i) \neq \chi_{B_i}(x_i)\}.$$

Observe that at each stage, $\mathrm{Ldim}(V_{i+1}) < \mathrm{Ldim}(V_i)$. Therefore, if we make d queries without correctly identifying the target, then we must have $\mathrm{Ldim}(V_d) = 0$. Then V_d is a singleton, which must be the target concept.

□

The proof of Proposition 3.2.23 uses strong consistency in a key way, as the hypothesis is generated by extending a certain partially specified subset. Nevertheless, the conclusion holds under the assumption that $C(\mathcal{C}, \mathcal{H}) = 2$, due to Proposition 3.2.17.

3.2.4 Adding membership queries and efficient learning of finite classes

Consistency dimension was originally derived from the notion of polynomial certificates, which was used to characterize learning with equivalence and membership queries in the finite case by [21]. The following is an improvement of the upper bound on EQ+MQ learning complexity of $\lceil \mathrm{C}(\mathcal{C}, \mathcal{H}) \log_2 |\mathcal{C}| \rceil$ implicit in the proof of Theorem 3.1.1 in [21] (stated explicitly in [8]). Our bound replaces $\log_2 |\mathcal{C}|$ with $\mathrm{Ldim}(\mathcal{C})$.

Theorem 3.2.24. *Suppose* $\mathrm{Ldim}(\mathcal{C}) = d < \infty$ *and* $\mathrm{C}(\mathcal{C}, \mathcal{H}) = c < \infty$. *Then* $\mathrm{LC}^{EQ+MQ}(\mathcal{C}, \mathcal{H}) \leq c'd + 1$, *where* $c' = \max\{1, c - 1\}$.

Proof. [1] We proceed by induction on d. The base case, $d = 0$, is trivial, as then $\mathcal{C}$ is a singleton. Suppose there is some element x such that $\mathrm{Ldim}(\mathcal{C} \cap x) < d + 1$ and $\mathrm{Ldim}(\mathcal{C} \setminus x) < d + 1$, where $\mathcal{C} \cap x := \{A \in \mathcal{C} \mid x \in A\}$ and $\mathcal{C} \setminus x := \{A \in \mathcal{C} \mid x \notin A\}$. Then by induction, any concept in $\mathcal{C} \cap x$

---

[1] The algorithm is similar to that of Theorem 3.2.6. However, the applications of Lemma 3.2.5 are replaced with membership queries.

can be learned in at most $c'd+1$ queries with guesses from $\mathcal{H}$, and the same is true for $\mathcal{C} \setminus x$. Submit x as a membership query. This tells us whether the target concept lies in $\mathcal{C} \cap x$ or $\mathcal{C} \setminus x$, and then we require at most $c'd+1$ many queries, for a total of $c'd+2 \leq c'(d+1)+1$ many queries.

If no such x exists, then for all x, either $\mathrm{Ldim}(\mathcal{C} \cap x) = d+1$ or $\mathrm{Ldim}(\mathcal{C} \setminus x) = d+1$. Let B be such that $x \in B$ iff $\mathrm{Ldim}(\mathcal{C} \cap x) = d+1$.

If $B \in \mathcal{H}$, then we submit B as our query. If we are incorrect, then by choice of B, the class $\mathcal{C}'$ of concepts consistent with the counterexample x_0 will have Littlestone dimension $\leq d$. By induction, any concept in $\mathcal{C}'$ can be learned in at most $c'd+1$ many queries, and so we learn the target in at most $c'd+2 \leq c'(d+1)+1$ queries.

If $B \notin H$, then, since $\mathrm{C}(\mathcal{C}, \mathcal{H}) = c$, there are some $x_0, \ldots, x_{c-1}$ such that there is no $A \in \mathcal{C}$ such that $B|_{\{x_0,\ldots,x_{c-1}\}} \sqsubseteq A$. (Observe that this cannot happen when $c = 1$. In fact, Proposition 3.2.17 and the proof of Proposition 3.2.23 imply that this cannot even happen when $c = 2$. In particular, $c' = c-1$.) Then, with notation as in the proof of Proposition 3.2.2,

$$\mathcal{C} = (\mathcal{C}^{(x_0,1-B(x_0))}) \cup \ldots \cup (\mathcal{C}^{(x_{c-1},1-B(x_{c-1}))}),$$

and $\mathrm{Ldim}(\mathcal{C}^{(x_i,1-B(x_i))}) \leq d$ for each i. By induction, any concept in each $\mathcal{C}^{(x_i,1-B(x_i))}$ can be learned in at most $c'd+1$ many queries. By submitting $x_0, \ldots, x_{c-2}$ as membership queries, we can determine some i such that the target belongs to $\mathcal{C}^{(x_i,1-B(x_i))}$ (if the result of each

membership query on x_j is $B(x_j)$, then we know that $i = c - 1$). We therefore learn in at most $c'd + 1 + (c - 1) = c'(d + 1) + 1$ many queries. □

We have a lower bound on learning complexity in terms of consistency dimension in this setting analogous to Proposition 3.2.7:

Proposition 3.2.25. *Suppose there is some (total) subset A which is n-consistent with $\mathcal{C}$ but which does not have a total extension in $\mathcal{H}$. Then $n < \mathrm{LC}^{EQ+MQ}(\mathcal{C}, \mathcal{H})$. In particular, $\mathrm{C}(\mathcal{C}, \mathcal{H}) \leq \mathrm{LC}^{EQ+MQ}(\mathcal{C}, \mathcal{H})$.*

Proof. We first show that $n < \mathrm{LC}^{EQ+MQ}(\mathcal{C}, \mathcal{H})$. If the learner submits x as a membership query, the teacher returns $A(x)$ if possible, that is, if there is a concept $B \in \mathcal{C}$ which agrees with the previous data and satisfies $B(x) = A(x)$.

By hypothesis, given any equivalence query H, the teacher can find some $x \in \mathrm{dom}(A)$ such that $H(x) \neq A(x)$, and the teacher returns a counterexample of this form if possible, that is, if there is a concept $B \in \mathcal{C}$ which agrees with the previous data and satisfies $B(x) = A(x)$.

Moreover, since A is n-consistent with $\mathcal{C}$, the teacher is able to return data of this form for the first n queries. Thus $\mathcal{C}$ cannot be learned with fewer than $n + 1$ equivalence queries from $\mathcal{H}$.

From this, it follows that $\mathrm{C}(\mathcal{C}, \mathcal{H}) \leq \mathrm{LC}^{EQ+MQ}(\mathcal{C}, \mathcal{H})$. □

Finally, putting together the various upper and lower bounds from this section we give a characterization of those problems efficiently learnable by equivalence and membership queries:

Theorem 3.2.26. *Let $(\mathcal{C}_n, \mathcal{H}_n)_{n\in\mathbb{N}}$ be a family of concept classes and hypothesis classes, respectively. Let $c_n = \mathrm{C}(\mathcal{C}_n, \mathcal{H}_n)$. Let $d_n = \mathrm{Ldim}(\mathcal{C}_n)$. The following are equivalent:*

(i) $\mathrm{LC}^{EQ+MQ}(\mathcal{C}_n, \mathcal{H}_n)$ *is bounded by a polynomial in n.*

(ii) *c_n and d_n are bounded by a polynomial in n.*

(iii) *The algorithm from Theorem 3.2.24 learns $\mathcal{C}_n$ in at most polynomially in n many membership queries and equivalence queries in $\mathcal{H}_n$.*

Proof. (ii) ⇒ (iii) follows immediately from Theorem 3.2.24, and (iii) ⇒ (i) follows by definition of learning complexity.

(i) ⇒ (ii): In Proposition 3.2.25, we showed that $\mathrm{LC}^{EQ+MQ}(\mathcal{C}, \mathcal{H}) \geq \mathrm{C}(\mathcal{C}, \mathcal{H})$, so it follows that if c_n is not polynomially bounded then neither is $\mathrm{LC}^{EQ+MQ}(\mathcal{C}_n, \mathcal{H}_n)$.

Now suppose that d_n is not polynomially bounded. By [7, Theorem 2.1] [1] we have

$$\mathrm{LC}^{EQ+MQ}(\mathcal{C}, \mathcal{H}) \geq \mathrm{LC}^{EQ+MQ}(\mathcal{C}, \mathcal{P}(X)) \geq \log\left(\frac{4}{3}\right) \cdot \mathrm{LC}^{EQ}(\mathcal{C}, \mathcal{P}(X)).$$

By [25, Theorems 5 and 6], we can replace $\mathrm{LC}^{EQ}(\mathcal{C}, \mathcal{P}(X))$ with $\mathrm{Ldim}(\mathcal{C})$. Thus:

$$\mathrm{LC}^{EQ+MQ}(\mathcal{C}_n, \mathcal{H}_n) \geq \log\left(\frac{4}{3}\right) \cdot d_n,$$

from which it follows that $\mathrm{LC}^{EQ+MQ}(\mathcal{C}_n, \mathcal{H}_n)$ is not polynomially bounded. □

---

[1] The inequality of [7] gives a lower bound for LC^{EQ+MQ} which improved on the lower bound of $\frac{\mathrm{LC}^{EQ}(\mathcal{C},\mathcal{P}(X))}{\log(1+\mathrm{LC}^{EQ}(\mathcal{C},\mathcal{P}(X)))}$ from [28, Theorem 3]. In fact, Theorem 3 of [28] actually suffices for our purposes.

Finally, the upper and lower bounds of this section also yield a characterization of which infinite classes are learnable in finitely many equivalence and membership queries.

Corollary 3.2.27. $\mathrm{LC}^{EQ+MQ}(\mathcal{C},\mathcal{H}) < \infty$ *iff* $\mathrm{Ldim}(\mathcal{C}) < \infty$ *and* $\mathrm{C}(\mathcal{C},\mathcal{H}) < \infty$.

3.2.5 The negation of the finite cover property

One can compare set systems with finite strong consistency dimension, to the model-theoretic classes of formulas and theories without the *finite cover property*, which we define below. Informally, the negation of the finite cover property allows for a specific quantitative bound for applications of compactness.

Definition 3.2.28. Fix a first order theory T. A formula $\phi(x;y)$ in the language of T *does not have the finite cover property (ncfp)* if there is $n = n(\phi)$ such that for all $\mathcal{M} \models T$, and every $p \subseteq \{\phi(x;a), \neg\phi(x;a) \,|\, a \in M\}$, the following holds: if every $q \subseteq p$ of size n is consistent, then p is consistent. We let $n(\phi)$ denote the minimal such n.

T does not have the finite cover property if all formulas $\phi(x;y)$ do not have the finite cover property.[1]

Consider the setting where $\mathcal{C}$ is generated by a formula $\phi(x;y)$, that is, $\mathcal{M} \models T$ and

$$\mathcal{C} = \mathcal{C}_\phi = \{\phi(M;b) \,|\, b \in M\}.$$

---

[1]The definition of nfcp on the formula level given here is stronger than original formulation in [39], but it gives an equivalent characterization on the level of theories.

That is, $\mathcal{C}_\phi$ consists of the ϕ-definable sets. Suppose $\phi^{opp}(y;x) = \phi(x;y)$ does not have the finite cover property, witnessed by some $n = n(\phi^{opp})$. Then, given any disjoint $A_0, A_1 \subseteq X$, if every size n subset of

$$p(y) := \{\phi^{opp}(y;a) \,|\, a \in A_1\} \cup \{\neg\phi^{opp}(y;a) \,|\, a \in A_0\}$$

is consistent, then $p(y)$ is consistent. We can identify this partial type with the partially specified subset A where

$$A(x) = \begin{cases} 0 & x \in A_0 \\ 1 & x \in A_1 \\ \text{unspecified} & \text{otherwise.} \end{cases}$$

By passing to an $|M|^+$-saturated extension $\mathcal{N} \succ \mathcal{M}$ to obtain a larger parameter set, we can find $b' \in \mathcal{N}$ satisfying $p(y)$. Then $\phi(M, b')$ is a total extension of A.

Supposing we have passed to an $|\mathcal{M}|^+$ saturated extension $\mathcal{N}$, we can let

$$\mathcal{H} = \mathcal{H}_\phi := \{\phi(M;b') \,|\, b' \in N\}.$$

That is, $\mathcal{H}_\phi$ consists of all externally ϕ-definable subsets of M, as N contains realizations of all consistent partial ϕ^{opp}-types over M. By the compactness theorem, this means that N contains realizations of all finitely consistent partial ϕ^{opp}-types over M. Having identified partially specified subsets of M with their corresponding ϕ^{opp}-type, this amounts to observing that

$\mathcal{H}_\phi$ contains total extensions of all finitely consistent partially specified subsets, equivalently, contains all finitely consistent total subsets.

This gives a model-theoretic motivation to the strategy suggested by Proposition 3.2.12. Adding all finitely consistent subsets to $\mathcal{H}$ amounts to saturating $\mathcal{N}$ so as to realize all ϕ^{opp}-types over $\mathcal{M}$.

If ϕ^{opp} has nfcp with $n(\phi) = n$, then the finitely consistent partial types are exactly the n-consistent types. Then $\mathcal{H}_\phi$ contains total extensions of all n-consistent partially specified subsets, so $SC(\mathcal{C}_\phi, \mathcal{H}_\phi) = n$. Note that ϕ^{opp}-types witnessing that n is the minimal such n at which ϕ^{opp} has nfcp give partially specified subsets witnessing that $\mathrm{SC}(\mathcal{C}_\phi, \mathcal{H}_\phi) \not< n$. This reflects a variant of Proposition 3.2.12 for strong consistency dimension.

In particular, formulas ϕ such that ϕ^{opp} has nfcp provide a rich family of examples where $\mathcal{C}_\phi$ has finite (strong) consistency threshold. That is, for such ϕ, it is necessary and sufficient for $\mathcal{H}$ to contain all externally ϕ-definable subsets (that is, all total finitely consistent partially specified subsets) to obtain $\mathrm{SC}(\mathcal{C}, \mathcal{H}) < \infty$. On the other hand, when ϕ^{opp} has the finite cover property, the externally definable sets are no longer sufficient, and one must venture beyond the sets ϕ is capable of cutting out to obtain $\mathrm{SC}(\mathcal{C}, \mathcal{H}) < \infty$ (that is, by adding some sets which are inconsistent).

Furthermore, Littlestone dimension of $\phi(x; y)$ (that is, the Littlestone dimension of $\mathcal{C}_\phi$) is expressible as a first-order property. So we will have $\mathrm{Ldim}(\mathcal{C}) = \mathrm{Ldim}(\mathcal{H})$. So when the context is a set system $\mathcal{C}$ generated by a stable formula $\phi(x; y)$ with $\phi^{opp}(y; x)$ nfcp, we can obtain a set system $\mathcal{H}$ such that $\mathrm{SC}(\mathcal{C}, \mathcal{H}) < \infty$, but $\mathcal{H}$ is not much more complicated than the original set

system - $\mathcal{H}$ has the same Littlestone dimension as $\mathcal{C}$. This is essentially the content of Theorem 3.2.14 when $\mathcal{C}$ has finite consistency threshold.

We give an example from model theory where ϕ^{opp} has the fcp.

Example 3.2.29. Let $\mathcal{M}$ be a structure in the language $\{E\}$, where E is an equivalence relation with one class of size n for each $n \in \mathbb{N}$, possibly with some infinite classes. Let

$$\phi(x; y) \quad \text{be} \quad E(x, y) \wedge x \neq y$$

and let $\mathcal{C} = \mathcal{C}_\phi$.

Suppose $a_1, \ldots, a_d$ are the elements belonging to the equivalence class of size d. Then the ϕ^{opp}-type $\{\phi(a_i, y) \,|\, i \leq d\}$ $(d-1)$-consistent but d inconsistent. Since there are equivalence classes of arbitrarily large size, these witness that ϕ^{opp} is not nfcp. One can check that $\mathrm{Ldim}(\mathcal{C}_\phi) = 2$.

In any $|M|^+$-saturated elementary extension $\mathcal{N}$ of $\mathcal{M}$, no additional elements are added to the finite equivalence classes already present in $\mathcal{M}$, though $\mathcal{N}$ adds new infinite classes and new elements to any existing infinite classes.

An attempt to learn $\mathcal{C}_\phi$ by equivalence queries following the strategy of Theorem 3.2.6 would be as follows. We are attempting to identify some $c \in M$. Letting a_0 be an element in a new infinite equivalence class in $\mathcal{N}$, we guess $\phi(M, a_0) = \emptyset$. Then any counterexample will identify an element belonging to the equivalence class of c. If c belongs to an infinite class, then we can find some $a_1 \in N$ which is a new element of this class. Then $\phi(M, a_1) = \{b \in M \,|\, E(b, c)\}$.

Then c is the only available counterexample, and we submit the correct concept $\phi(M, c)$ at our next turn. However, if c belongs to the finite class of size n, then N has no new elements in this class. Then the relevant queries, which are of the form $\phi(M, a)$ for a in the class of c, are already present in $\mathcal{C}_\phi$. Then we are essentially attempting to identify a singleton from a set of size n, and it is clear that the process could take up to n additional guesses.

3.3 Efficient learnability of regular languages

In a seminal paper, [1] showed that regular languages are efficiently learnable with equivalence queries plus membership queries, and in this subsection, we will use Theorem 3.2.24 to give an alternate short proof of this fact.[1] Let $\mathcal{L}_{n,m}$ be the class of binary regular languages on strings of length at most m specified by a deterministic finite automaton on at most n nodes. The $\mathcal{L}^*$ algorithm of [1] specifically uses $\mathcal{O}(n)$ equivalence queries and $\mathcal{O}(mn^2)$ membership queries. We let $DFA_2(n)$ denote the collection of (equivalence classes of) deterministic finite automata accepting binary strings and having at most n nodes. The proof of the next proposition is straightforward.

Proposition 3.3.1. *The Littlestone dimension of $DFA_2(n)$ is at most $\mathcal{O}(n \log n)$.*

Proof. In [22, Proposition 1], it is shown that $|DFA_2(n)| \leq \frac{n^{2n} 2^n n}{n!} \leq 2^{\mathcal{O}(n \log n)}$. From this, it follows that the Littlestone dimension of $DFA_2(n)$ is at most $\mathcal{O}(n \log n)$. □

---

[1] In the following sections, we only make use of *proper equivalence queries*, that is, $\mathcal{H} = \mathcal{C}$. We shall therefore let $\mathrm{C}(\mathcal{C}) := \mathrm{C}(\mathcal{C}, \mathcal{C})$, which we will call the consistency dimension of $\mathcal{C}$ (with analogous notation for strong consistency dimension).

The proof of the following proposition reveals the connection between consistency and the Myhill-Nerode theorem.

Proposition 3.3.2. $\mathrm{C}(DFA_2(n)) \leq 2\binom{n+1}{2} = n(n+1)$.

Proof. Fix a subset C of binary strings and x, y binary strings. We say that z is a (C-) distinguishing extension of x and y if $xz \in C$ but $yz \notin C$ or vice versa. If x and y have no distinguishing extension, then we say x and y are C-equivalent, and write $x \sim_C y$. The Myhill-Nerode theorem [34] says that a subset of binary strings of length m is the accept set of a finite automaton with at most n nodes if and only if the number of $\sim_C$ classes is at most n. Thus, given any subset C of the binary strings of length m which is not a regular language recognized by an automaton with at most n nodes, there are at least $n+1$ $\sim_C$-classes of elements. Pick representatives $x_0, \ldots, x_n$ from $n+1$ classes, and for each $i < j$, pick some z_{ij} that is a distinguishing extension of x_i and x_j. Then restricting C to the partial assignment on $\{x_k z_{ij} \mid i < j,\, k = i, j\}$, a domain of size $2\binom{n+1}{2} = n(n+1)$ that witnesses that $x_i \not\sim_C x_j$ for all $i \neq j$, we can see that this restriction is inconsistent with the class of regular languages recognized by automata with at most n nodes. Therefore $\mathrm{C}(DFA_2(n)) \leq n(n+1)$. [1] □

Now, by Theorem 3.2.24 and the previous two results, it follows that:

Theorem 3.3.3. *The class $\mathcal{L}_{n,m}$ is learnable in at most $\mathcal{O}(n \log n)$ equivalence queries and at most $\mathcal{O}(n \log n)(n(n+1))$ membership queries.*

---

[1] Note that the same proof shows that the consistency dimension of $DFA_m(n)$ is also at most $n(n+1)$.

It is interesting to note that contrary to $\mathcal{L}^*$, when using the algorithm from Theorem 3.2.24, there is no dependence on m, the length of the binary strings which the teacher is allowed to provide as counterexamples[1].

Theorem 3.2.6 now implies that $\mathcal{L}_{n,m}$ is learnable in at most $(n(n+1))^{O(n \log n)}$ equivalence queries. Theorem 3.2.22 shows that a finite class $\mathcal{C}$ is learnable in at most $\lceil \mathrm{SC}(\mathcal{C}) \cdot \ln |\mathcal{C}| \rceil$ equivalence queries. Since [3] showed that $\mathcal{L}_{n,m}$ is not learnable in polynomially many equivalence queries, it follows that $\mathrm{SC}(\mathcal{L}_{n,m})$ cannot be polynomial in n, m.

3.3.1 Learning ω-languages

In this section, we consider the natural extension to languages on infinite strings indexed by ω, called ω-languages. For an alphabet Σ, we denote by Σ^ω, the strings of symbols from Σ of order type ω. Similar to the previous section, we consider an automaton, which consists of the collection $\mathbb{A} = (\Sigma, Q, q_0, \delta)$, where Q is a finite collection of states, q_0 is the initial state, and $\delta : Q \times \Sigma \to 2^Q$ is a transition rule. To form a language, an automaton is equipped with an acceptance criterion.[2] Fix a subset $F \subseteq Q$. A run of a *Büchi automaton* is accepting if and only if it visits the set F infinitely often. An ω-language is ω-regular if it is recognized by a non-deterministic Büchi automaton. A run of a *co-Büchi automaton* is accepting if and only if it visits F only finitely often. Let $\psi : Q \to \{1, \ldots, k\}$ be a function, which we think of as a

---

[1]We should also note that $\mathcal{L}^*$ was improved by Schapire to give a better bound on membership queries (still depending on m). [37].

[2]Numerous acceptance criteria have been extensively studied in the literature, and we refer the reader to [5; 17; 16] for overviews.

coloring of the states of the automaton. Let c be the minimum color which is visited infinitely often. A run of a *parity automaton* is accepting if and only if c is odd.

Two ω-regular languages are equivalent if they agree on the set of periodic words [32], which allows for the possibility of recognizing the ω-language using finitary automata. This is the approach of [5; 17], whose notation we follow closely. A *family of DFAs* (FDFA) $\mathcal{F}$ is a pair (Q, P) where Q is a DFA with $|Q|$ states and P is a collection of $|Q|$ many DFAs, which we refer to as *progress DFAs* - one DFA P_q for each state q of Q. Given a pair of finite words, (u, v), a run of our family of DFAs consists of running Q on u, then running $P_{Q(u)}$ on v where $Q(u)$ is the ending state of Q on u. The pair (u, v) can be used to represent an infinite periodic word uv^ω.

Let $FDFA(n, m)$ be the class of families of deterministic finite automata where the leading automaton has at most n nodes and the progress automata each have at most m nodes. It is *not* quite true that once an ω-regular language has been reduced to an FDFA that one can use $\mathcal{L}^*$ directly to learn the various DFAs in the family [5, section 4]. It is also not completely obvious what the bounds for Littlestone and consistency dimension are in terms of the DFAs in the family, but the next two results give such bounds which imply the efficient learnability of ω-regular languages.

Proposition 3.3.4. *The class $FDFA(n, m)$ has Littlestone dimension at most $\mathcal{O}(n \log n + nm \log m)$.*

Proof. The number of FDFAs of size (n, m) is clearly at most $|DFA_2(n)| \cdot |DFA_2(m)|^n$. That is

$$|FDFA(n, m)| \leq |DFA_2(n)| \cdot |DFA_2(m)|^n.$$

It follows that

$$\mathrm{Ldim}(FDFA(n, m)) \leq \log(|DFA_2(n)| \cdot |DFA_2(m)|^n)$$

and using [22, Proposition 1], the desired bound follows. □

Proposition 3.3.5. $\mathrm{C}(FDFA(n, m)) \leq 2^{\binom{n(m+1)}{2}} = \mathcal{O}(n^2m^2)$.

Proof. A run of an FDFA on (u, v) can be simulated by the run of an appropriate automaton in the class $DFA_3(n \cdot (m + 1))$. To see this, input word $u\$v$ where \$ is a new symbol (recall we are assuming u, v are binary) to a DFA which has the same diagram as the FDFA but with an edge labeled with \$ from each state of the leading automaton to the initial state of the corresponding progress DFA. Now it follows by Proposition 3.3.2 that the consistency dimension of $FDFA(n, m)$ is at most $2^{\binom{n(m+1)}{2}}$. □

Using the previous two results together with Theorem 3.2.24, one can deduce the efficient learnability of $FDFA(n, m)$:

Theorem 3.3.6. *The class $FDFA(n, m)$ is learnable in at most $\mathcal{O}(n \log n + nm \log m)$ equivalence queries and at most $\mathcal{O}((\log n + m \log m) \cdot (n^3m^2))$ membership queries.*

We have formulated our bounds in terms of the number of states in the FDFA corresponding to a given ω-language. In [5; 17] bounds on the number of states of FDFAs in terms of the

number of states of automata for ω-languages with various acceptors are given. Specifically, the following bounds hold:

1. When $\mathcal{A}$ is a deterministic Büchi (DBA) or co-Büchi (DCA) automaton with n states, there is an equivalent FDFA of size at most $(n, 2n)$ [17, 5.3].
2. When $\mathcal{A}$ is a deterministic parity automaton (DPA) with n states and k colors, there is an equivalent FDFA of size at most (n, kn) [17, 5.4].
3. When $\mathcal{A}$ is a nondeterministic Büchi automaton (NBA) with n states, there is an equivalent FDFA of size at most $(2^{\mathcal{O}(n \log n)}, 2^{\mathcal{O}(n \log n)})$.

Any NBA can be translated into a DPA, and so 2) yields the efficient learnability of ω-regular languages *in terms of the number of states in a DPA* (this translation also yields 3). However, the translation from NBA to DPA is known to require an exponential increase in the number of states in general [36]. From an FDFA of size at most (n, k) there is a translation into an NBA with at most $\mathcal{O}(n^2k^3)$ states [17, Theorem 5.8], and so it follows that the exponential increase in states in moving from NBAs to FDFAs is necessary [17, Theorem 5.6].

Finally, we mention that [6] define restricted classes of ω-languages for which right-congruence is *fully informative*, and isolate numerous classes (e.g. for each type of acceptor from the previous subsection) of ω-languages for which an infinitary invariant of the Myhill-Nerode theorem holds. This variant of Myhill-Nerode is sufficient to bound the consistency dimension (and thus establish the learnability) of the classes in terms of the number of right equivalence classes of $\sim_{\mathcal{L}}$ similar to the proof of Proposition 3.3.2.

3.4 Random counterexamples and EQ-learning

In section 3.2 we characterized learnability by equivalence queries in terms of Littlestone dimension and strong consistency dimension. The setting of equivalence query learning [2] as described in section 3.2 deals with worst-case bounds for algorithmic identification of concepts by a learner. In this section, we follow [4] and analyze a slightly different situation, in which the teacher selects the counterexamples at random, and we seek to bound the *expected* number of queries. [4] worked specifically with concept classes coming from boolean matrices, which was convenient for their notation. Our formulation is equivalent, but we use slightly different notation.

Throughout this section, let X be a finite set, let $\mathcal{C}$ be a set system on X, and let μ be a probability measure on X. For $A, B \in \mathcal{C}$, let $\Delta(A, B) = \{x \in X \mid A(x) \neq B(x)\}$ denote the symmetric difference of A and B.

Definition 3.4.1. We denote, by $\mathcal{C}_{\bar{x}=\bar{i}}$ for $\bar{x} \in X^n$ and $\bar{i} \in \{0,1\}^n$, the set system $\{A \in \mathcal{C} \mid A(x_j) = i_j,\ j = 1, \ldots, n\}$. For $A \in \mathcal{C}$ and $a \in X$, we let

$$u(A, a) = \mathrm{Ldim}(\mathcal{C}) - \mathrm{Ldim}(\mathcal{C}_{a=A(a)}).$$

For any $a \in X$, either $\mathcal{C}_{a=1}$ or $\mathcal{C}_{a=0}$ has Littlestone dimension strictly less than that of $\mathcal{C}$ and so:

Lemma 3.4.2. *For $A, B \in \mathcal{C}$ and $a \in X$ with $A(a) \neq B(a)$,*

$$u(A,a) + u(B,a) \geq 1.$$

Next, we define a directed graph which is similar to the *elimination graph* of [4].

Definition 3.4.3. We define the *thicket query graph* $G_{TQ}(\mathcal{C}, \mu)$ to be the weighted directed graph on vertex set $\mathcal{C}$ such that the directed edge from A to B has weight $d(A,B)$ equal to the expected value of $\mathrm{Ldim}(\mathcal{C}) - \mathrm{Ldim}(\mathcal{C}_{x=B(x)})$ over $x \in \Delta(A,B)$ with respect to the distribution $\mu|_{\Delta(A,B)}$. [1]

Definition 3.4.4. The *query rank* of $A \in \mathcal{C}$ is defined as: $\inf_{B \in \mathcal{C}}(d(A,B))$.

Lemma 3.4.5. *For any $A \neq B \in \mathcal{C}$, $d(A,B) + d(B,A) \geq 1$.*

Proof. Noting that $\Delta(A,B) = \Delta(B,A)$, and using Lemma 3.4.2:

$$\begin{aligned} d(A,B) + d(B,A) &= \sum_{a \in \Delta(A,B)} \frac{\mu(a)}{\mu(\Delta(A,B))}(u(A,a) + u(B,a)) \\ &\geq \sum_{a \in \Delta(A,B)} \frac{\mu(a)}{\mu(\Delta(A,B))} = 1. \end{aligned}$$

□

Definition 3.4.6. [4, Definition 14] Let G be a weighted directed graph and $l \in \mathbb{N}$, $l > 1$. A *deficient l-cycle* in G is a sequence $v_0, \ldots v_{l-1}$ of distinct vertices such that for all $i \in [l]$, $d(v_i, v_{(i+1) \pmod l}) \leq \frac{1}{2}$ with strict inequality for at least one $i \in [l]$.

---

[1] Here one should think of the query by the learner as being A, and the actual hypothesis being B. The teacher samples from $\Delta(A,B)$, and the learner now knows the value of the hypothesis on x.

The next result is similar to Theorems 16 (the case $l = 3$) and Theorem 17 (the case $l > 3$) of [4], but our proof is rather different (note that the case $l = 2$ follows easily from Lemma 3.4.5).

Theorem 3.4.7. *The thicket query graph $G_{TQ}(\mathcal{C}, \mu)$ has no degenerate l-cycles for $l \geq 2$.*

The analogue of Theorem 16 can be adapted in a very similar manner to the technique employed by [4]. However, the analogue of the proof of Theorem 17 falls apart in our context; the reason is that Lemma 3.4.2 is analogous to Lemma 6 of [4] (and Lemma 3.4.5 is analogous to Lemma 13 of [4]), but our lemmas involve inequalities instead of equations. The inductive technique of [4, Theorem 17] is to shorten degenerate cycles by considering the weights of a particular edge in the elimination graph along with the weight of the edge in the opposite direction. Since one of those weights being large forces the other to be small (by the *equalities* of their lemmas), the induction naturally separates into two useful cases. In our thicket query graph, things are much less tightly constrained - one weight of an edge being large does not force the weight of the edge in the opposite direction to be small. However, the technique employed in our proof seems to be flexible enough to adapt to prove Theorems 16 and 17 of [4].

Proof. Suppose the vertices in the degenerate l-cycle are $A_0, \ldots, A_{l-1}$.

By the definition of degenerate cycles and $d(-,-)$, we have, for each $i \in \mathbb{Z}/l\mathbb{Z}$, that

$$\sum_{a \in \Delta(A_i, A_{i+1})} \frac{\mu(a)}{\mu(\Delta(A_i, A_{i+1}))} u(A_i, a) \leq \frac{1}{2},$$

so, clearing the denominator, we have

$$\sum_{a \in \Delta(A_i, A_{i+1})} \mu(a) u(A_i, a) \leq \frac{1}{2} \mu(\Delta(A_i, A_{i+1})). \tag{3.2}$$

Note that throughout this argument, the coefficients are being calculated modulo l. Notice that for at least one value of i, the inequality in Equation 3.2 must be strict.

Let G, H be a partition of

$$\mathcal{X} = \{A_1, \ldots, A_l\}.$$

Now define

$$D(G,H) := \{a \in X \,|\, \forall A_1, B_1 \in G,\, \forall A_2, B_2 \in H,\, A_1(a) = B_1(a),\, A_2(a) = B_2(a), A_1(a) \neq A_2(a)\}.$$

The following fact follows from the definition of $\Delta(A, B)$ and $D(-,-)$.

Fact 3.4.8. *The set $\Delta(A_i, A_{i+1})$ is the disjoint union, over all partitions of $\mathcal{X}$ into two pieces G, H such that $A_i \in G$ and $A_{i+1} \in H$ of the sets $D(G,H)$.*

Now, take the sum of the inequalities Equation 3.2 as i ranges from 1 to l. On the left-hand of the resulting sum, we obtain

$$\sum_{i=1}^{l} \left(\sum_{G,H \text{ a partition of } \mathcal{X},\, A_i \in G, A_{i+1} \in H} \left(\sum_{a \in D(G,H)} \mu(a) u(A_i, a) \right) \right).$$

On the right-hand side of the resulting sum we obtain

$$\frac{1}{2}\sum_{i=1}^{l}\left(\sum_{G,H \text{ a partition of } \mathcal{X},\, A_i\in G, A_{i+1}\in H}\left(\sum_{a\in D(G,H)}\mu(a)\right)\right).$$

Given a partition G, H of $\{A_1, \ldots, A_l\}$ we note that the term $D(G,H) = D(H,G)$ appears exactly once as an element of the above sum for a fixed value of i exactly when $A_i \in G$ and $A_{i+1} \in H$ or $A_i \in H$ and $A_{i+1} \in G$.

Consider the partition G, H of $\mathcal{X}$. Suppose that $A_j, A_{j+1}, \ldots, A_k$ is a block of elements each contained in G, and that A_{j-1}, A_{k+1} are in H. Now consider the terms $i = j-1$ and $i = k$ of the above sums (each of which where $D(G,H)$ appears).

On the left-hand side, we have $\sum_{a\in D(G,H)}\mu(a)u(A_{j-1},a))$ and $\sum_{a\in D(G,H)}\mu(a)u(A_k,a))$. Note that for $a \in D(G,H)$, we have $a \in \Delta(A_{j-1}, A_k)$. So, by Lemma 3.4.2, we have

$$\sum_{a\in D(G,H)}\mu(a)u(A_{j-1},a)) + \sum_{a\in D(G,H)}\mu(a)u(A_k,a)) \geq \sum_{a\in D(G,H)}\mu(a).$$

On the right-hand side, we have

$$\frac{1}{2}(\sum_{a\in D(G,H)}\mu(a) + \sum_{a\in D(G,H)}\mu(a)) = \sum_{a\in D(G,H)}\mu(a).$$

For each G, H a partition of X, the terms appearing in the above sum occur in pairs as above by Fact 3.4.8, and so, we have the left-hand side is at least as large as the right-hand side of the

sum of inequalities Equation 3.2, which is impossible, since one of the inequalities must have been strict by our degenerate cycle. □

Theorem 3.4.9. *There is at least one element $A \in \mathcal{C}$ with query rank at least $\frac{1}{2}$.*

Proof. If not, then for every element $A \in \mathcal{C}$, there is some element $B \in \mathcal{C}$ such that $d(A, B) < \frac{1}{2}$. So, pick, for each $A \in \mathcal{C}$, an element $f(A)$ such that $d(A, f(A)) < \frac{1}{2}$. Now, fix $A \in \mathcal{C}$ and consider the sequence of elements of $\mathcal{C}$ given by $(f^i(A))$; since $\mathcal{C}$ is finite, at some point the sequence repeats itself. So, take a list of elements $B, f(B), \ldots, f^n(B) = B$. By construction, this yields a bad cycle, contradicting Theorem 3.4.7. □

3.4.1 The thicket max-min algorithm

In this subsection we show how to use the lower bound on query rank proved in Theorem 3.4.9 to give an algorithm which yields the correct concept in linearly (in the Littlestone dimension) many queries from $\mathcal{C}$. The approach is fairly straightforward—essentially the learner repeatedly queries the highest query rank concept. The approach is similar to that taken in [4, Section 5] but with query rank in place of their notion of *informative*.

Now we informally describe the thicket max-min-algorithm. At stage i, the learner is given information of a concept class $\mathcal{C}_i$. The learner picks the query

$$A = \arg\max_{A \in \mathcal{C}_i} \left(\min_{B \in \mathcal{C}_i} d_{\mathcal{C}_i}(A, B) \right).$$

The algorithm halts if the learner has picked the actual concept C. If not, the teacher returns a random element $a_i \in \Delta(A, C)$ at which point the learner knows the value of $C(a_i)$. Then

$$\mathcal{C}_{i+1} = (\mathcal{C}_i)_{a_i = C(a_i)}.$$

Let $T(\mathcal{C})$ be the expected number of queries before the learner correctly identifies the target concept.

Theorem 3.4.10. *The expected number of queries to learn a concept in a class $\mathcal{C}$ is less than or equal to* $2\,\mathrm{Ldim}(\mathcal{C})$.

Proof. The expected drop in the Littlestone dimension of the concept class induced by any query before the algorithm terminates is at least $\frac{1}{2}$ by Theorem 3.4.9; so the probability that the drop in the Littlestone dimension is positive is at least $\frac{1}{2}$ for any given query. So, from $2n$ queries, one expects at least n drops in Littlestone dimension. □

We give a rough bound on the probability that the algorithm has not terminated after a certain number of queries. Since a query can reduce the Littlestone dimension of the induced concept class by at most $\mathrm{Ldim}(\mathcal{C})$ and the expected drop is at least $\frac{1}{2}$, the probability that a query reduces the Littlestone dimension is at least $\frac{1}{2\,\mathrm{Ldim}(\mathcal{C})}$. Then the probability that the Littlestone dimension of the induced concept class after n queries is positive is at most the probability of fewer than $\mathrm{Ldim}(\mathcal{C})$ many successes in the binomial distribution with probability

$\frac{1}{2\,\mathrm{Ldim}(\mathcal{C})}$ and n trials. It follows by Hoeffding's inequality that the probability that the algorithm has not terminated after n steps is at most

$$e^{-2\frac{\left(\frac{n}{2\,\mathrm{Ldim}(\mathcal{C})}-\mathrm{Ldim}(\mathcal{C})\right)^2}{n}}.$$

CHAPTER 4

COMPRESSION SCHEMES AND STABILITY

This chapter represents section 5 along with a portion of section 1 of the preprint [13], co-authored with James Freitag.

In this chapter, we introduce compression schemes for concept classes. Specifically, the notion we work with is equivalent to d-compression with b extra bits (of Floyd and Warmuth [18]). In [23], Laskowski and Johnson proved that the concept class corresponding to a stable formula has an extended d-compression for some d. Later, a result of Laskowski appearing as [19, Theorem 4.1.3] in fact showed that one could take d equal to the Shelah 2-rank (Littlestone dimension) and uses 2^d many reconstruction functions. We show that $d+1$ many reconstruction functions suffice.

In this chapter, we follow the notation and definitions given in [19] on *compression schemes*, a notion due to Littlestone and Warmuth [26]. Roughly speaking, $\mathcal{C}$ admits a *d-dimensional compression scheme* if, given any finite subset F of X and some $f \in \mathcal{C}$, there is a way of encoding the set F with only d-many elements of F in such a way that F *can be recovered*. We will give a formal definition, but we note that numerous variants of this idea appear throughout the literature. For instance:

- Size d-array compression [9].
- Extended compression schemes with b extra bits [18].

The next definition, which is the notion of compression we will work with in this section is equivalent to the notion of a d-compression with b extra bits (of Floyd and Warmuth) [23, see Proposition 2.1].

Definition 4.0.1. We say that a concept class $\mathcal{C}$ has a *d-compression* if there is a compression function $\kappa : \mathcal{C}_{fin} \to X^d$ and a finite set $\mathcal{R}$ of reconstruction functions $\rho : X^d \to 2^X$ such that for any $f \in \mathcal{C}_{fin}$

1. $\kappa(f) \subseteq dom(f)$
2. $f = \rho(\kappa(f))|_{dom(f)}$ for at least one $\rho \in \mathcal{R}$.

We work with the above notion mainly because it is the notion used in [19], and our goal is to improve a result of Laskowski appearing there [19, Theorem 4.1.3]. In [23], Laskowski and Johnson prove that the concept class corresponding to a stable formula has an extended d-compression for some d. The precise value of d is not determined, but was conjectured to be the Littlestone dimension. A later unpublished result of Laskowski appearing as [19, Theorem 4.1.3] in fact showed that one could take d equal to the Shelah 2-rank (Littlestone dimension) and uses 2^d many reconstruction functions. In Theorem 4.0.4, we will show that $d + 1$ many reconstruction functions suffice.

The question of Johnson and Laskowski is the analogue (for Littlestone dimension) of a well-known open question from VC theory [18]: is there a bound $A(d)$ linear in d such that every class of VC dimension d has a compression scheme of size at most $A(d)$? In general, there is known to be bound which is at most exponential in d [33].

Definition 4.0.2. Suppose $\mathrm{Ldim}(\mathcal{C}) = d$. Given a partial function f, say that f is *exceptional* for $\mathcal{C}$ if for all $a \in \mathrm{dom}(f)$,

$$\mathcal{C}_{(a,f(a))} := \{g \in \mathcal{C} \mid g(a) = f(a)\}$$

has Littlestone dimension d.

Definition 4.0.3. Suppose $\mathrm{Ldim}(\mathcal{C}) = d$. Let $f_{\mathcal{C}}$ be the partial function given by

$$f_{\mathcal{C}}(x) = \begin{cases} 0 & \mathrm{Ldim}(\mathcal{C}_{(x,0)}) = d \\ 1 & \mathrm{Ldim}(\mathcal{C}_{(x,1)}) = d \\ \text{undefined} & \text{otherwise.} \end{cases}$$

It is clear that $f_{\mathcal{C}}$ extends any partial function exceptional for $\mathcal{C}$.

Theorem 4.0.4. *Any concept class $\mathcal{C}$ of Littlestone dimension d has an extended d-compression with $(d+1)$-many reconstruction functions.*

Proof. If $d = 0$, then $\mathcal{C}$ is a singleton, and one reconstruction function suffices. So we may assume $d \geq 1$.

Fix some $f \in \mathcal{C}_{fin}$ with domain F. We will run an algorithm to construct a tuple of length at most d from F by adding one element at each step of the algorithm. During each step of the algorithm, we also have a concept class $\mathcal{C}_i$, with $\mathcal{C}_0 = \mathcal{C}$ initially.

If f is exceptional in $\mathcal{C}_{i-1}$, then the algorithm halts. Otherwise, pick either:

- $a_i \in F$ such that $f(a_i) = 1$ and

$$(\mathcal{C}_{i-1})_{(a_i,1)} := \{g \,|\, g \in \mathcal{C}_{i-1},\, g(a_i) = 1\}$$

  has Littlestone dimension less than $\mathrm{Ldim}(\mathcal{C}_{i-1})$. In this case, set $\mathcal{C}_i := (\mathcal{C}_{i-1})_{(a_i,1)} = \{g \,|\, g \in \mathcal{C}_{i-1},\, g(a_i) = 1\}$.

- $d_i \in F$ such that $f(d_i) = 0$ and

$$(\mathcal{C}_{i-1})_{(d_i,0)} := \{g \,|\, g \in \mathcal{C}_{i-1},\, g(d_i) = 0\}$$

  has Littlestone dimension less than $\mathrm{Ldim}(\mathcal{C}_{i-1})$. In this case, set $\mathcal{C}_i := (\mathcal{C}_{i-1})_{(d_i,0)}$.

We allow the algorithm to run for at most d steps. There are two distinct cases. If our algorithm has run for d steps, let $\kappa(f)$ be the tuple $(\bar{a}, \bar{d})$ of all of the elements a_i as above followed by all of the elements d_i as above for $i = 1, \ldots, d$. By choice of a_i and d_i, this tuple consists of d distinct elements. By construction the set

$$\mathcal{C}_{(\bar{a},\bar{d})} := \{g \in \mathcal{C} |\, g(a_i) = 1,\, g(d_i) = 0\}$$

has Littlestone dimension 0, that is, there is a unique concept in this class. So, given $(c_1, c_2, \ldots, c_n) \in X^d$ consisting of distinct elements, for $i = 0, \ldots, d$, we let $\rho_i(c_1, \ldots, c_n)$ be some g belonging to

$$\{g \in \mathcal{C} \,|\, g(c_j) = 1 \text{ for } j \leq i,\, g(c_j) = 0 \text{ for } j > i\},$$

if such a g exists. By construction, for some i, the Littlestone dimension of the concept class $\{g \in \mathcal{C} \cap F \mid g(c_j) = 1 \text{ for } j \leq i,\ g(c_j) = 0 \text{ for } j > i\}$ is zero, and so g is uniquely specified and will extend f.

We handle cases where the algorithm halts early by augmenting two of the reconstruction functions ρ_0 and ρ_1 defined above. Because ρ_0 and ρ_1 have so far only been defined for tuples consisting of d distinct elements, we can extend these to handle exceptional cases by generating tuples with duplicate elements.

If the algorithm stops at some step $i > 1$, then it has generated a tuple of length $i - 1$ consisting of some elements a_j and some elements d_k. Let $\bar{a}$ consist of the elements a_j chosen during the algorithm, and let $\bar{d}$ consist of the elements d_k chosen during the running of the algorithm. Observe that f is exceptional for $\mathcal{C}_{(\bar{a},\bar{d})}$.

If $\bar{a}$ is not empty, with initial element a', then let $\kappa(f) = (\bar{a}, a', \bar{d}, a', \ldots, a') \in F^d$. From this tuple, one can recover $(\bar{a}, \bar{d})$ (assuming $\bar{a}$ is nonempty), so we let $\rho_1(\bar{a}, a', \bar{d}, a', \ldots, a')$ be some total function extending $f_{\mathcal{C}_{(\bar{a},\bar{d})}}$, which itself extends f. So $\rho_1(\bar{a}, \bar{d})$ extends f whenever the algorithm halts before step d is completed *and* some a_i was chosen at some point. If $\bar{a}$ is empty, then let $\kappa(f) = (\bar{d}, d', \ldots, d') \in F^d$, where d' is the initial element of $\bar{d}$. From this tuple, one can recover $(\emptyset, \bar{d})$ (assuming $\bar{a}$ is empty), so we let $\rho_0(\bar{d}, d', \ldots, d')$ be total function extending $f_{\mathcal{C}_{(\emptyset,\bar{d})}}$, which itself extends f. Finally, if the algorithm terminates during step 1, then it has generated the empty tuple. In this case, let $\kappa(f) = (c, \ldots, c)$ for some $c \in F$. Then $\mathrm{Ldim}(\mathcal{C}) = \mathrm{Ldim}(\mathcal{C}_{(c,l)})$ for some $l \in \{0, 1\}$. In particular, if we have defined $\kappa(f') = (c, \ldots, c)$ above for some f' where the algorithm only returns c (rather than the empty tuple), then

$1 - l = f'(c) \neq f(c)$, and so any such f' is handled by ρ_{1-l}. So we may overwrite ρ_l to set $\rho(c, \ldots, c)$ to be a total function extending $f_{\mathcal{C}}$, which itself extends f. For any tuple output by our algorithm, one of the reconstruction functions produces an extension of the original concept.

□

CHAPTER 5

BANNED SEQUENCE PROBLEMS AND THE SAUER-SHELAH LEMMA

This chapter represents the preprint [12], co-authored with James Freitag.

5.1 Introduction

A single combinatorial notion called *VC dimension* determines important dividing lines in both machine learning (PAC learnability of a class) and model theory (the independence/non-independence dichotomy, IP/NIP) [24], and the finiteness of this quantity plays an essential role in the development of various structural results in theories without the independence property and in machine learning. Often at the root of these developments is the Sauer-Shelah Lemma, which for a formula $\phi(x;y)$ without the independence property, gives a polynomial bound on the shatter function associated with ϕ—that is, the number of consistent ϕ-types over finite sets. Without NIP, however, the number of ϕ-types can grow exponentially in the size of the finite parameter set. In a recent paper, Bhaskar [11] noticed that when the formula ϕ is actually stable, that is, ϕ has finite Shelah 2-rank (also called Littlestone dimension or thicket dimension in the context of set systems), one can relax the way in which the ϕ-types are constructed, allowing for trees of parameters (explained below) while still proving polynomial bounds on the resulting collection of consistent ϕ-types. Again, in the absence of stability the number of types formed in this manner can grow exponentially in the height of the tree. Following Bhaskar, we refer

to this growth dichotomy theorem as the Stable Sauer-Shelah Lemma. In [14], we notice that stability also determines an important dividing line in machine learning; stability determines learnability in various settings of *online learning*. In these settings of learning, various results at their core rely on the polynomial growth of the stable shatter function.

In both settings described above, the growth of the number of types being polynomially bounded or exponential is completely determined by whether a simple combinatorial notion of dimension is finite, and the upper bound (which is tight in general) on the number of such types (in terms of the appropriate notion of dimension) is identical in both cases. In light of this, Bhaskar naturally asks if there is a single combinatorial principle which explains both the Sauer-Shelah Lemma and the stable variant. The main purpose of this chapter is to set up a general context in which one can prove Sauer-Shelah type results into which both of the above contexts fit, answering Bhaskar's question as well as proving new results. Our solution to the problem is quite general and deals with what we call *banned sequence problems*.

Our general setup of banned sequence problems is an interesting combinatorial setting in its own right, and we will roughly describe the simplest context here. Suppose that you consider the collection of all binary sequences of length n, and for each subset of the indices of size k, there is at least one "banned subsequence" of length k. How many binary sequences of length n are there which avoid each of the banned sequences on all subsets of the indices of size k? Subject to some very mild assumptions on how the banned sequences are chosen, we show that there are at most

$$\sum_{i=0}^{k-1} \binom{n}{i}$$

such sequences. This bound is the bound of the Sauer-Shelah Lemma. Without the mild assumptions, we show that this bound can be violated. The generality of our setup covers both the settings mentioned above as well as yielding some new results.

We give a slight improvement of a result of Malliaris and Terry [30] regarding sizes of cliques and independent sets in stable graphs. Essentially, their result uses the finiteness of a certain combinatorial dimension, *tree rank*, in order to establish polynomial bounds strong enough to get a version of the Erdős-Hajnal conjecture, among other results (Malliaris and Terry also develop further structural properties of graphs which we will not touch on in this thesis). We examine tree rank in the general context of banned sequence problems, and as a result, give a slight improvement to their bounds.

In the last part of the chapter, we turn to the setting of op-ranks. For each $s \in \mathbb{N}$, Guingona and Hill [20] define a rank of partial types, op_s-rank. For instance, when $s = 1$, op_1-rank is equal to the Shelah 2-rank. Working with set systems of finite op_s-rank, we establish a new variant of the Sauer-Shelah Lemma using our banned sequence setup.

We note that not every known variant of the Sauer-Shelah Lemma seems to fit into the context of banned sequence problems; the main results of [15] establish a variant of Sauer-Shelah for n-dependent theories which does not seem to easily fit into our context of banned sequence problems. Is there a general setup which also covers the known Sauer-Shelah style results for n-dependent theories? This seems reasonable to ask because n-dependent theories generalize NIP theories in a way similar to how theories with finite op_s-rank generalize stable theories.

5.1.1 Organization

In section 5.2, we give the necessary preliminary notation for our results. In section 5.3, we lay out the basic theory of banned sequence problems along with some applications. In section 5.4, we generalize our banned sequence problems. In section 5.4.2, we apply generalized banned problems to the op-rank setting.

5.2 Preliminaries

Our primary combinatorial tool applies to theorems surrounding VC dimension and Littlestone dimension (also known as Shelah's 2-rank in model theory or thicket dimension in [11]), and we recall those definitions and relevant theorems. The next several definitions can be found in various sources, e.g. [40].

Throughout, any indexing starts at 0, and $[n] := \{0, 1, \ldots, n-1\}$. By $\binom{[n]}{k}$ we mean the collection of all subsets of $[n]$ of size k.

Recall that a set system $(X, \mathcal{F})$ (often referred to as $\mathcal{F}$ when X is understood) consists of a set X and a collection $\mathcal{F} \subseteq \mathcal{P}(X)$ of subsets of X. For $Y \subseteq X$, the projection of $\mathcal{F}$ onto Y is the set system with base set Y and collection of subsets

$$\mathcal{F}_Y := \{F \cap Y \mid F \in \mathcal{F}\}.$$

VC dimension measures the ability of a set system to pick out subsets of a set of a given size.

Definition 5.2.1. A set system $(X, \mathcal{F})$ *shatters* a set Y if $\mathcal{F}_Y = \mathcal{P}(Y)$. The VC dimension of $\mathcal{F}$ is the largest $k < \omega$ such that $\mathcal{F}$ shatters some set of size k, or is infinite if $\mathcal{F}$ shatters arbitrarily large sets. The *shatter function*

$$\pi_{\mathcal{F}}(n) := \sup_{Y \subseteq X, |Y|=n} |\mathcal{F}_Y|$$

is given by the supremum of the size of the projection onto subsets of a given size.

If a set system has finite VC dimension, then we obtain a polynomial bound on the shatter function.

Theorem 5.2.2 (Sauer-Shelah Lemma)**.** *Let $\mathcal{F}$ be a set system of VC dimension k. Then the maximum size of a projection from $\mathcal{F}$ onto a set $A = \{a_0, \ldots, a_{n-1}\}$ of size n is $\sum_{i=0}^{k} \binom{n}{i}$. In particular,*

$$\pi_{\mathcal{F}}(n) \leq \sum_{i=0}^{k} \binom{n}{i}.$$

Several proofs of the Sauer-Shelah Lemma can be found in various sources, e.g. [40; 35].

Littlestone dimension is a variant of VC dimension; our development follows [11]. (Bhaskar calls Littlestone dimension "thicket dimension"—we prefer to use Littlestone dimension, or use "stable" to describe the general setting.) Given a set from the set system, elements are presented sequentially, with the element presented depending on membership of previous elements.

Definition 5.2.3. A *binary element tree* of height n with labels from X is a function $T : 2^{<n} \to X$. A *node* is a binary sequence $\sigma \in 2^{<n}$ along with its label, $a_\sigma := T(\sigma)$. A *leaf* is a binary sequence of length n, $\tau : [n] \to \{0, 1\}$. A leaf τ is properly labeled by a set A if for all $m < n$,

$$a_{\tau|_{[m]}} \in A \quad \text{iff} \quad \tau(m) = 1.$$

Definition 5.2.4. The *Littlestone dimension* of a set system $(X, \mathcal{F})$ is the largest $k < \omega$ such that there is a binary element tree of height k with labels from X such that every leaf can be properly labeled by elements of $\mathcal{F}$, or is infinite if there are such trees of arbitrary height. The *stable shatter function* (what Bhaskar calls the "thicket shatter function") $\rho_{\mathcal{F}}(n)$ is the maximum number of leaves properly labeled by elements of $\mathcal{F}$ in a binary element tree of height n.

Theorem 5.2.5 (Stable (Thicket) Sauer-Shelah Lemma [11])**.** *Let $\mathcal{F}$ be a set system of Littlestone dimension k. Then the maximum number of properly labeled leaves in a binary element tree of height n is $\sum_{i=0}^{k} \binom{n}{i}$. In particular,*

$$\rho_{\mathcal{F}}(n) \leq \sum_{i=0}^{k} \binom{n}{i}.$$

VC dimension and the (VC) shatter function can be viewed in the context of binary element trees where every node of the same height is labeled with the same element, i.e. $a_\sigma = a_{\sigma'}$ whenever $|\sigma| = |\sigma'|$.

There are dual notions of both VC dimension and Littlestone dimension, and their corresponding shatter functions, where the roles of elements and sets are reversed.

Definition 5.2.6. Given a set system $(X, \mathcal{F})$, the dual set system $(X, \mathcal{F})^*$, or just $\mathcal{F}^*$, is the set system with base set $\mathcal{F}$ where the subsets are given by

$$\{F \mid F \in \mathcal{F}, x \in F\}$$

for each $x \in X$. The dual VC (resp., Littlestone) dimension of $\mathcal{F}$ is the VC (resp., Littlestone) dimension of $\mathcal{F}^*$.

Dual Littlestone dimension can be calculated by examining *binary decision trees*, where nodes are labeled by sets in the set system, and leaves are labeled by elements. Dual VC dimension can be calculated similarly.

In model theory, given a model $\mathcal{M}$, the VC (resp., Littlestone) dimension of a partitioned formula $\phi(x; y)$ is the VC (resp., Littlestone) dimension of the set system

$$(M^{|x|}, \{\phi(M^{|x|}, b) \mid b \in M^{|y|}\}).$$

These combinatorial notions encode model-theoretic dividing lines. A formula is NIP iff it has finite VC dimension, and is stable iff it has finite Littlestone dimension.

5.3 The combinatorics of banned sequences

The binary element tree structure used to define Littlestone dimension allows us to identify a leaf of the tree with the binary sequence corresponding to the path through the tree to that leaf. Then counting properly-labeled leaves amounts to counting the corresponding binary sequences. We establish a framework for counting binary sequences under certain conditions reflecting the tree structure, from which we will obtain a unified proof of the Sauer-Shelah lemmas.

5.3.1 Banned binary sequences and Sauer-Shelah lemmas

Our framework for counting binary sequences will reflect the height of the tree as well as the dimension (either Littlestone or VC) of the set system. We find it easier to count banned sequences. Having Littlestone dimension $k-1$ says that in a tree of height k, there are some leaves which cannot be properly labeled, and those leaves correspond to sequences that we ban.

Definition 5.3.1. A *k-fold banned binary sequence problem (BBSP) of length n*, for $0 \leq k \leq n$ is a function

$$f : \binom{[n]}{k} \times 2^{n-k} \to \mathcal{P}(2^k) \setminus \{2^k\}.$$

Informally, for each k-subset of $[n]$ and each binary sequence of length $n-k$, the binary sequences of length k not selected by f are banned, and we ban at least one such sequence. Sometimes we will refer to the sequences omitted by the function f as *banned subsequences.*

Remark 5.3.2. It will be convenient to view binary sequences as functions, where the domain is the appropriate set of indices. Given $S \in \binom{[n]}{k}$, let $\bar{S} := [n] \setminus S$. When we consider $f(S, Y)$

for some fixed S, we view $Y \in 2^{n-k}$ as a function $Y : \bar{S} \to \{0,1\}$, and elements of $f(S,Y)$ as functions $Z : S \to \{0,1\}$, identifying 2^{n-k} with $2^{\bar{S}}$ and 2^k with 2^S.

Given $X : [n] \to \{0,1\}$ and $S \subseteq [n]$, let X_S denote the restriction $X|_S$ of X to S, that is, the subsequence obtained by restricting to the indices in S.

We shall denote the union of two binary sequences Y and Z with disjoint domains as $Y \sqcup Z$. For example, if Y has domain $\{0,2\}$, with $Y(0) = Y(2) = 0$, and Z has domain $\{1\}$ with $Z(1) = 1$, then $Y \sqcup Z$ is the binary sequence 010. When we wish to extend a sequence by appending some $j \in \{0,1\}$, we will merely write $Y \sqcup j$, with the index of j usually understood from the context.

For a fixed $S \in \binom{[n]}{k}$, we denote the elements of S by $\{s_0, \ldots, s_{k-1}\}$, where $s_0 < s_1 < \ldots < s_{k-1}$.

Definition 5.3.3. A *solution* to a k-fold banned binary sequence problem f of length n is a binary sequence $X \in 2^n$ such that for any $S \in \binom{[n]}{k}$,

$$X_S \in f(S, X_{\bar{S}}).$$

A sequence which is not a solution is *banned*.

Intuitively, a solution to a banned binary sequence problem is a sequence which avoids every banned subsequence. In applications to binary element trees, properly labeled leaves will correspond to solutions of a certain banned binary sequence problem.

Without further assumptions, the number of solutions of a BBSP can grow exponentially in n for a fixed k.

Proposition 5.3.4. *A k-fold BBSP f of length n has at most $(2^k - 1)2^{n-k}$ solutions.*

Proof. Fix $S \in \binom{n}{k}$. For $Y : \bar{S} \to \{0,1\}$ and $Z : S \to \{0,1\}$, $Y \sqcup Z$ can only be a solution if $Z \in f(S, Y)$, and for each of 2^{n-k} many such Y's, there are at most $2^k - 1$ many Z's. □

We observe that to obtain this bound, and so have only 2^{n-k} banned sequences, we must be able to find a collection $\mathcal{B}$ of 2^{n-k} sequences $X : [n] \to \{0,1\}$ such that for all $S \in \binom{[n]}{k}$ and all $Y : \bar{S} \to \{0,1\}$, there is some $X \in \mathcal{B}$ such that $Y \subseteq X$. Then we can set $f(S, Y) := \{X_S\}$, and then every $X \in 2^n \setminus \mathcal{B}$ is a solution. In general this is not possible. It is possible for $k = n$, where we simply pick a sequence of length n to ban, $k = n - 1$, where $\mathcal{B}$ can consist of, say, the two constant sequences, $k = 1$, given below, and $k = 0$, which is trivial. But this condition already cannot be met for $k = 2$ and $n = 4$. In this case, one can verify that the minimum size of $\mathcal{B}$ to satisfy the above condition is 5, and so a 2-fold BBSP of length 4 can have at most 11 solutions.

Example 5.3.5. Let f be the 1-fold BBSP of length n given by

$$f(\{s\}, Y) = \begin{cases} 1 & Y \text{ has an even number of 1s} \\ 0 & Y \text{ otherwise.} \end{cases}$$

Then f has 2^{n-1} solutions, given by those binary sequences which have an even number of 1s.

We therefore need stronger hypotheses in order to bound the number of solutions by the Sauer-Shelah bound.

Definition 5.3.6. A k-fold banned binary sequence problem f of length n is *not hereditary* if there is $S \in \binom{[n]}{k}$ and a function $g : 2^S \to 2^{\bar{S}}$ such that

- for all $Z : S \to \{0,1\}$, we have $Z \in f(S, g(Z))$, and
- for all $Z_\alpha \neq Z_\beta$, the first index at which $g(Z_\alpha) \sqcup Z_\alpha$ and $g(Z_\beta) \sqcup Z_\beta$ differ is in S.

Otherwise, say f is *hereditary*.

One can think of the second condition as stating that g is continuous in the sense that for any $t \in \bar{S}$, $g(Z_\alpha)(t) = g(Z_\beta)(t)$ whenever $(Z_\alpha)_{S\cap[t]} = (Z_\beta)_{S\cap[t]}$, that is, $g(Z)(t)$ depends only on $Z_{S\cap[t]}$.

We will usually suppress the function g, and instead use indices to indicate the mapping—given $Z_\alpha : S \to \{0,1\}$, we let $Y_\alpha := g(Z_\alpha)$. Then being not hereditary amounts to finding $S \in \binom{[n]}{k}$ such that for all $Z_\alpha : S \to \{0,1\}$, we can associate a $Y_\alpha : \bar{S} \to \{0,1\}$ such that $Z_\alpha \in f(S, Y_\alpha)$, and for any $Z_\alpha \neq Z_\beta$, the first index at which $Y_\alpha \sqcup Z_\alpha$ and $Y_\beta \wedge Z_\beta$ differ is in S.

For our purposes, being hereditary is the desirable property; hereditary BBSPs allow us to obtain the Sauer-Shelah bound on the number of solutions. We can also study binary element trees using hereditary BBSPs, and thus derive the corresponding Sauer-Shelah lemmas. We choose to call these BBSPs hereditary because proving the Sauer-Shelah bound on the number of solutions uses derivative BBSPs in the inductive step, and being hereditary is preserved in these derivative problems.

Theorem 5.3.7. *Any hereditary k-fold banned binary sequence problem of length n has at most $\sum_{i=0}^{k-1} \binom{n}{i}$ solutions.*

The proof is by induction. We will make use of the recursive property of binomial coefficients,

$$\binom{n}{i} = \binom{n-1}{i-1} + \binom{n-1}{i},$$

from which it follows that

$$\sum_{i=0}^{k-1} \binom{n}{i} = \sum_{i=0}^{k-2} \binom{n-1}{i} + \sum_{i=0}^{k-1} \binom{n-1}{i}.$$

In particular, we use the inductive strategy suggested by these equalities. The base cases will be $k = 0$ and $k = n$. In the inductive step, given a hereditary k-fold banned binary sequence problem f of length n, we seek two derivative problems: a $(k-1)$-fold banned binary sequence problem of length $n-1$, and a k-fold banned binary sequence problem of length $n-1$, both of which are hereditary. We will use banned sequences of these derivative problems to construct banned sequences of the original problem.

Definition 5.3.8. Let f be a k-fold banned binary sequence problem of length n, for $1 \leq k \leq n-1$.

- Let $\hat{f}$ be the $(k-1)$-fold banned binary sequence problem of length $n-1$ given as follows: for all $T \in \binom{[n-1]}{k-1}$, all $Y \in 2^{n-k}$, and all $Z \in 2^{k-1}$, let

$$Z \notin \hat{f}(T, Y) \qquad \text{iff} \qquad \exists j \in \{0,1\}\ Z \sqcup j \notin f(T \sqcup \{n-1\}, Y).$$

- Let f' be the k-fold banned binary sequence problem of length $n-1$ given as follows: for all $S \in \binom{[n-1]}{k}$, all $Y \in 2^{n-k-1}$, and all $Z \in 2^k$, let

$$Z \notin f'(S,Y) \qquad \text{iff} \qquad \forall j \in \{0,1\}\ Z \notin f(S, Y \sqcup j)$$

That is, the banned subsequences in $\hat{f}(T,Y)$ are those subsequences which can be extended by a particular j to a banned subsequence in $f(T \cup \{n-1\}, Y)$. In particular, any banned sequence of $\hat{f}$ has some extension which is a banned sequence of f.

The banned subsequences in $f'(S,Y)$ are those subsequences which are banned subsequences in $f(S, Y \sqcup j)$ for any extension of Y by j. In particular, any extension of any banned sequence of f' is a banned sequence of f.

Lemma 5.3.9. *Suppose f is a hereditary k-fold banned binary sequence problem of length n, for $1 \leq k \leq n-1$. Then both $\hat{f}$ and f' are also hereditary.*

Proof. Suppose for contradiction that $\hat{f}$ is not hereditary. Then there exists $T \in \binom{[n-1]}{k-1}$ such that for each $Z_\alpha : T \to \{0,1\}$, there is $Y_\alpha : \bar{T} \to \{0,1\}$ such that $Z_\alpha \in \hat{f}(T, Y_\alpha)$, and for any $Z_\alpha \neq Z_\beta$, the first index at which $Y_\alpha \sqcup Z_\alpha$ and $Y_\beta \sqcup Z_\beta$ differ belongs to T. Note that for some Z_α and some $j \in \{0,1\}$, we have that

$$Z_\alpha \sqcup j \notin f(T \cup \{n-1\}, Y_\alpha),$$

or else associating each $Z_\alpha \sqcup j$ with Y_α would witness that f itself is not hereditary. Then, by definition of $\hat{f}$, $Z_\alpha \notin \hat{f}(T, Y_\alpha)$, a contradiction. So $\hat{f}$ is hereditary.

Suppose for contradiction that f' is not hereditary. Then there exists $S \in \binom{[n-1]}{k}$ such that for all $Z_\alpha : S \to \{0,1\}$, there is $Y_\alpha : \bar{S} \to \{0,1\}$ such that $Z_\alpha \in f'(S, Y_\alpha)$, and for any $Z_\alpha \neq Z_\beta$, the first index at which $Y_\alpha \sqcup Z_\alpha$ and $Y_\beta \sqcup Z_\beta$ differ belongs to S. By definition of f', for each Z_α, there is $j_\alpha \in \{0,1\}$ such that

$$Z_\alpha \in f(S, Y_\alpha \sqcup j_\alpha).$$

Let Y'_α be $Y_\alpha \sqcup j_\alpha$. Then associating Z_α with Y'_α witnesses that f is not hereditary, a contradiction. So f' is hereditary. □

Proof of Theorem 5.3.7. We prove the result by induction on n and k. Let f be a hereditary k-fold banned binary sequence problem of length n. Let $B(f)$ denote the number of sequences banned by f. It suffices to prove that

$$B(f) \geq 2^n - \sum_{i=0}^{k-1} \binom{n}{i}.$$

The base cases are $k = n$ and $k = 0$. When $k = n$, we have $2^n - \sum_{i=0}^{k-1} \binom{n}{i} = 1$, and any BBSP has at least one banned sequence. When $k = 0$, for all $Y \in 2^n$, we have $f(\emptyset, Y) = \emptyset$. Then for all $X \in 2^n$, we have $X_\emptyset = \emptyset \notin f(\emptyset, X_{[n]})$. So all $X \in 2^n$ are banned, and f has no solutions.

Otherwise, we proceed by induction. We show

$$B(f) \geq B(\hat{f}) + B(f').$$

For each sequence $\hat{X}$ that is banned by $\hat{f}$, there is at least one extension X which is banned by f, and we pick one such extension. For each sequence X' banned by f', at most one extension of X' was already obtained by extending a sequence $\hat{X}$ banned by $\hat{f}$. So there is at least one extension X of X' which is banned by f (by definition of f') but was not obtained by extending banned sequences for $\hat{f}$. Therefore these banned sequences of f constructed from f' and $\hat{f}$ have no common members, and so we have

$$B(f) \geq B(\hat{f}) + B(f'),$$

as desired. By induction, we have that

$$\begin{aligned} B(f) &\geq \left(2^{n-1} - \sum_{i=0}^{k-2} \binom{n-1}{i}\right) + \left(2^{n-1} - \sum_{i=0}^{k-1} \binom{n-1}{i}\right) \\ &\geq 2^n - \sum_{i=0}^{k-1} \binom{n}{i}. \end{aligned}$$

Thus f has at most $\sum_{i=0}^{k-1} \binom{n}{i}$ solutions. □

It shall be useful to identify a stronger banned binary sequence problem, namely those in which $f(S, Y)$ depends only on S.

Definition 5.3.10. A banned binary sequence problem f is *independent* if $f(S,Y) = f(S,Y')$ for any $Y, Y' : \bar{S} \to 0,1$. When f is independent, we write $f(S)$.

Corollary 5.3.11. *Any independent k-fold banned binary sequence problem f of length n has at most $\sum_{i=0}^{k-1} \binom{n}{i}$ solutions.*

Proof. We check that f is hereditary. If not, then there is $S \in \binom{[n]}{k}$ such that for all $Z_\alpha : S \to \{0,1\}$, there is $Y_\alpha : \bar{S} \to \{0,1\}$ with $Z_\alpha \in f(S, Y_\alpha) = f(S)$. But then $f(S) = 2^k$, a contradiction. The result follows from Theorem 5.3.7. □

Banned binary sequence problems provide a common framework to prove Sauer-Shelah type bounds.

Proof of Theorem 5.2.2. We obtain a $k+1$-fold independent BBSP f of length n as follows. Given $S = \{a_{s_0}, \ldots a_{s_k}\} \in \binom{A}{k+1}$, let $f(S)$ be the set of binary sequences Z of length $k+1$ such that there is some $F \in \mathcal{F}$ such that $a_{s_i} \in F$ iff $Z(i) = 1$. We have that $f(S) \neq 2^{k+1}$ since the VC dimension of $\mathcal{F}$ is k, and f is clearly independent. Then a subset B of A is in the projection from $\mathcal{F}$ onto A iff the characteristic sequence of B (i.e. the sequence where the jth entry is 1 iff $a_j \in B$) is a solution to f. The result follows from Corollary 5.3.11. □

Proof of Theorem 5.2.5. Let T be a binary element tree of height n, with nodes a_σ for $\sigma \in 2^{<n}$. We obtain a $k+1$-fold hereditary BBSP of length n, f, as follows. Given $S = \{s_0, \ldots, s_k\} \in \binom{[n]}{k+1}$ where $s_0 < s_1 < \cdots < s_k$ and $Y : \bar{S} \to \{0,1\}$, we obtain a binary element tree of height $k+1$ by taking all paths $\tau \in 2^n$ through T such that $Y \subseteq \tau$. Any two such paths first differ at some node a_σ where $|\sigma| \in S$, so removing all other nodes gives us the binary element tree $T_{S,Y}$

of height $k+1$. Since $\mathcal{F}$ has Littlestone dimension k, not all leaves of $T_{S,Y}$ can be properly labeled, so let $f(S,Y)$ be the set of all sequences whose corresponding leaves in $T_{S,Y}$ can be properly labeled. Then a leaf in T can only be properly labeled if the corresponding sequence is a solution to f.

We now show that f as constructed above is hereditary. Fix $S = \{s_0, \ldots, s_k\}$, and suppose for contradiction that this choice of S witnesses that f is *not* hereditary. Then, for each $Z_\alpha : S \to \{0,1\}$, there is $Y_\alpha : \bar{S} \to \{0,1\}$ such that $Z_\alpha \in f(S, Y_\alpha)$. We obtain a complete binary tree of height $k+1$ specified by each path $Y_\alpha \sqcup Z_\alpha$ constructed in this manner, restricted to S. In particular, any two paths constructed in this manner first differ at some index in S, as the first index at which $Y_\alpha \sqcup Z_\alpha$ and $Y_\beta \sqcup Z_\beta$ differ is in S. Since each Z_α is not banned, we have a complete binary tree of height $k+1$ in which every leaf can be properly labeled, a contradiction.

The result then follows from Theorem 5.3.7. □

5.3.2 An application to type trees

Banned binary sequence problems can be applied to other problems with a tree structure. We use this to improve a result of Malliaris and Terry [30].

Definition 5.3.12. Given a graph $G = (V, E)$ on n vertices and $A \subseteq 2^{<n}$, closed under initial segments, we say that a labeling $V = \{a_\eta \mid \eta \in A\}$ is a *type tree* if for each $\eta \in A$:

1. If $\eta \sqcup 0 \in A$, then $a_{\eta \sqcup 0}$ is nonadjacent to a_η. If $\eta \sqcup 1 \in A$, then $a_{\eta \sqcup 1}$ is adjacent to a_η.
2. If $\eta \subsetneq \eta' \subsetneq \eta''$, then a_η is adjacent to $a_{\eta'}$ if and only if a_η is adjacent to $a_{\eta''}$.

A type tree has height h if $A \subseteq 2^{<h}$ but $A \not\subseteq 2^{<h-1}$.

More generally, given a model $\mathcal{M}$, a finite set $B \subseteq M$, a finite collection Δ of partitioned formulas closed under cycling of the variables, and $A \subseteq \omega^{<\omega}$ closed under initial segments, a type tree is a labeling $B = \{b_\eta \mid \eta \in A\}$ such that, for any $\eta, \eta' \in A$, b_η and $b_{\eta'}$ have the same Δ-type over their common predecessors $\{b_\zeta \mid \zeta \subsetneq \eta, \beta \subsetneq \eta'\}$ iff $\eta \subseteq \eta'$ or $\eta' \subseteq \eta$. Type trees are used in more generality in [29], but we restrict our attention to type trees of graphs.

Definition 5.3.13. The *tree rank* of a graph $G = (V, E)$ is the largest integer t such that there is a subset $V' \subset V$ and some indexing $V' = \{a_\eta \mid \eta \in 2^{<t}\}$ which is a type tree for the induced graph on V', i.e. the type tree of V' is a full binary tree of height t.

The main interest in type trees for graphs lies in the fact that if we have a branch of length h for a graph (V, E) with tree rank t, there is a clique or independent set of size at least $\max\{\frac{h}{2}, t\}$ [30, Lemma 4.4]. More generally, branches through a type tree can be used to extract indiscernible sequences [29, Theorem 3.5]. In both cases, stability establishes the length of long branches through the type tree. For graphs, this is by way of tree rank—observe that the edge relation having Littlestone dimension k implies that the tree rank is at most $k + 1$. We use banned binary sequence problems to improve the bounds from [30, Theorem 4.6]. The improvement is modest, but it demonstrates how banned binary sequence problems accommodate the combinatorics of type trees, at least in the case of the graph edge relation.

Theorem 5.3.14. *Let $G = (V, E)$ be a graph with $|V| = n$ and tree rank $t \geq 2$. Suppose $A \subseteq 2^{<n}$ and $V = \{a_\eta \mid \eta \in A\}$ is a type tree with height h, where $h \geq 2t$. Then*

$$h \geq (n \cdot (t-2)!)^{\frac{1}{t}} + 1.$$

The assumptions on t and h are not restrictive if our aim is to obtain cliques or independent sets. If $t = 1$, then there is no branching, and we obtain a clique or independent set of size $\frac{n}{2}$. If $h < 2t$, then the largest clique or independent set guaranteed by [30, Lemma 4.4] is just the tree rank t.

Proof. We will associate a hereditary t-fold banned binary sequence problem of length $h-1$ with the type tree. Fix any subset $S = \{s_0, \ldots, s_{t-1}\}$ in $\binom{[h-1]}{t}$ and any $Y : \bar{S} \to \{0,1\}$. Let $f(S,Y)$ consist of all $Z : S \to \{0,1\}$ such that $(Y \sqcup Z)_{[s_{t-1}+1]}$ is an element of $2^{<h}$ which is in the index set A of the type tree.

Suppose for contradiction that $f(S,Y) = 2^t$. For each $\eta \in 2^{<t+1}$, we identify η with a partial function $Z_\eta : S \rightharpoonup \{0,1\}$, where $\eta(i) = Z_\eta(s_i)$. For each $i < t$ and each $\eta : [i] \to \{0,1\}$ in $2^{<t+1} \setminus 2^t$, let $b_\eta = a_{(Y \sqcup Z_\eta)_{[s_i]}}$. For each $\eta : [t] \to \{0,1\}$ in 2^t, let $b_\eta = a_{(Y \sqcup Z_\eta)_{[s_{t-1}+1]}}$. Note that each b_η is well-defined—in particular, for $\eta \in 2^t$, if $b_\eta = a_{(Y \sqcup Z_\eta)_{[s_{t-1}+1]}}$ was not an element of the type tree, then we would have $Z_\eta \notin f(S,Y)$. The rest of the elements are well-defined since the index set of a type tree is closed under initial segments. Then the b_η define a full binary type tree of height $t+1$, contradicting our assumption that the tree rank of G is t. So f is a t-fold BBSP of length $h-1$.

We check that f is hereditary. Suppose for contradiction that f is not hereditary, witnessed by some $S \in \binom{[h-1]}{t}$. So for each $Z_\alpha : S \to \{0,1\}$, there is $Y_\alpha : \bar{S} \to \{0,1\}$ such that $Z_\alpha \in f(S, Y_\alpha)$, and for $\alpha \neq \beta$, the first index at which $Y_\alpha \sqcup Z_\alpha$ and $Y_\beta \sqcup Z_\beta$ differ is in S. Identify each $\eta \in 2^{<t+1}$ with Z_η as above. For each $i < t$ and each $\eta : [i] \to \{0,1\}$, let $b_\eta = a_{(Y_\alpha \sqcup Z_\alpha)_{[s_i]}}$ for any α such that $Z_\eta \subseteq Z_\alpha$. For each $\eta : [t] \to \{0,1\}$, let $b_\eta = a_{(Y_\eta \sqcup Z_\eta)_{[s_{t-1}+1]}}$.

These b_η are defined since $Z_\eta \notin f(S, Y_\eta)$ by hypothesis. All other b_η, for $\eta : [i] \to \{0,1\}$, $i < t$, are defined since type trees are closed under initial segments, and well-defined since if $Z_\eta \subseteq Z_\alpha, Z_\beta$, then the first index at which $Y_\alpha \sqcup Z_\alpha$ and $Y_\beta \sqcup Z_\beta$ differ is in S and is at least s_i. Then the b_η form a type tree of height $t+1$, a contradiction.

Thus a type tree of height h gives a hereditary t-fold banned binary sequence problem of length $h-1$. Now, by Theorem 5.3.7, the number of nodes at level h_0, $h_0 = 0, \ldots, h-1$, is at most

$$\sum_{i=0}^{t-1} \binom{h_0}{i}.$$

Thus, the total number of nodes of a type tree of height h and tree rank t is at most

$$\begin{aligned}
\sum_{h_0=0}^{h-1} \sum_{i=0}^{t-1} \binom{h_0}{i} &= 1 + \sum_{h_0=1}^{h-1} \sum_{i=0}^{t-1} \binom{h_0}{i} \\
&= 1 + \sum_{h_0=1}^{h-1} \left(1 + \sum_{i=1}^{t-1} \binom{h_0}{i} \right) \\
&\leq \sum_{h_0=1}^{h-1} \sum_{i=1}^{t-1} \binom{h-1}{i} \qquad (5.1) \\
&\leq \sum_{h_0=1}^{h-1} \sum_{i=1}^{t-1} \frac{(h-1)^{t-1}}{(t-1)!} \qquad (5.2) \\
&\leq \sum_{h_0=1}^{h-1} \frac{(h-1)^{t-1}}{(t-2)!} \\
&\leq \frac{(h-1)^t}{(t-2)!},
\end{aligned}$$

where estimates in (Equation 5.1) and (Equation 5.2) follow from hypotheses on t and h. Then

$$\frac{(h-1)^t}{(t-2)!} \geq n,$$

so

$$h \geq (n \cdot (t-2)!)^{\frac{1}{t}} + 1.$$

□

Under the hypotheses of Theorem 5.3.14, applying [30, Lemma 4.4] gives us a clique or independent set of size at least

$$\frac{(n \cdot (t-2)!)^{\frac{1}{t}} + 1}{2}.$$

This is an improvement of the lower bound given by Malliaris and Terry [30, Corollary 4.7].

5.4 Generalized banned sequence problems and applications

In this section we generalize Theorem 5.3.7 to the setting of j-ary sequences and apply the resulting combinatorics to prove Sauer-Shelah type lemmas in the op-rank context [20].

5.4.1 Banned j-ary sequence problems

Definition 5.4.1. A *k-fold banned j-ary sequence problem of length n*, for $0 \leq k \leq n$, is a function

$$f : \binom{[n]}{k} \times j^{n-k} \to \mathcal{P}(j^k) \setminus \{j^k\}.$$

A *solution* to g is a j-ary sequence $X \in j^n$ such that for any $S \in \binom{[n]}{k}$,

$$X_S \in f(S, X_{\bar{S}}).$$

As before, for a fixed $S \in \binom{[n]}{k}$,, we denote the elements of S by $\{s_0, \ldots, s_{k-1}\}$, where $s_0 < s_1 < \ldots < s_{k-1}$. When we consider $f(S, Y)$, we view $Y \in j^{n-k}$ as a function $Y : \bar{S} \to [j] = \{0, 1, \ldots, j-1\}$, and elements of $f(S, Y)$ as functions $Z : S \to [j]$, identifying j^{n-k} with $j^{\bar{S}}$ and j^k with j^S.

Definition 5.4.2. A k-fold banned j-ary sequence problem (j-ary BSP) f of length n is *not hereditary* if there is $S \in \binom{[n]}{k}$ and a function $g : j^S \to j^{\bar{S}}$ such that

- for all $Z : S \to [j]$, we have $Z \in f(S, g(Z))$, and
- for all $Z_\alpha \neq Z_\beta$, the first index at which $g(Z_\alpha) \sqcup Z_\alpha$ and $g(Z_\beta) \sqcup Z_\beta$ differ is in S.

Otherwise, say f is *hereditary*.

As before, we suppress g and use indices to indicate the mapping, letting Y_α denote Z_α.

Theorem 5.4.3. *Any hereditary k-fold banned j-ary sequence problem of length n has at most $\sum_{i=0}^{k-1}(j-1)^{n-i}\binom{n}{i}$ solutions.*

The proof is similar to the proof of Theorem 5.3.7. We use the generalized versions of the derivative problems for the induction.

Definition 5.4.4. Let f be a k-fold banned j-ary sequence problem of length n, for $1 \leq k \leq n-1$.

- Let $\hat{f}$ be the $(k-1)$-fold banned j-ary sequence problem of length $n-1$ given as follows: for all $T \in \binom{[n-1]}{k-1}$, all $Y \in j^{n-k}$, and all $Z \in j^{k-1}$, let

$$Z \notin \hat{f}(T,Y) \qquad \text{iff} \qquad \exists l \in [j]\ Z \sqcup l \notin f(T \sqcup \{n-1\}, Y).$$

- Let f' be the k-fold banned j-ary sequence problem of length $n-1$ given as follows: for all $S \in \binom{[n-1]}{k}$, all $Y \in j^{n-k-1}$, and all $Z \in j^{k}$, let

$$Z \notin f'(S,Y) \qquad \text{iff} \qquad \forall l \in [j]\ Z \notin f(S, Y \sqcup l)$$

Lemma 5.4.5. *Suppose f is a hereditary k-fold banned j-ary sequence problem of length n, for $1 \leq k \leq n-1$. Then both $\hat{f}$ and f' are also hereditary.*

The proof is a straightforward generalization of Lemma 5.3.9.

Proof of Theorem 5.4.3. The proof is by induction on n and k. Let f be a hereditary k-fold banned j-ary sequence problem of length n.

Let $B(f)$ denote the number of sequences banned by f. It suffices to prove that

$$B(f) \geq j^n - \sum_{i=0}^{k-1} (j-1)^{n-i} \binom{n}{i}.$$

The base cases are $k = n$ and $k = 0$. When $k = n$, $j^n - \sum_{i=0}^{k-1}(j-1)^{n-i}\binom{n}{i} = 1$, and any j-ary BSP has at least one banned sequence. When $k = 0$, for all $X \in j^n$, we have $X_\emptyset = \emptyset \notin f(\emptyset, X_{[n]}) = \emptyset$. So all $X \in j^n$ are banned.

Otherwise, we proceed by induction. We show

$$B(f) \geq B(\hat{f}) + B(f') \cdot (j-1).$$

For each sequence $\hat{X}$ that is banned by $\hat{f}$, there is at least one extension X which is banned by f, and we pick one such extension. For each sequence X' banned by f', there are at least $j-1$ extensions X of X' which are banned by f but were not obtained by extending banned sequences for $\hat{f}$. Therefore these banned sequences constructed from f' and $\hat{f}$ have no common members, and so we have

$$B(f) \geq B(\hat{f}) + B(f') \cdot (j-1),$$

as desired. By induction, we have that

$$\begin{aligned} B(f) &\geq j^{n-1} - \sum_{i=0}^{k-2}(j-1)^{n-1-i}\binom{n-1}{i} \\ &\quad + (j-1)\left(j^{n-1} - \sum_{i=0}^{k-1}(j-1)^{n-1-i}\binom{n-1}{i}\right) \\ &= j^n - \sum_{i=0}^{k-1}(j-1)^{n-i}\binom{n}{i}. \end{aligned}$$

Thus, f has at most $\sum_{i=0}^{k-1}(j-1)^{n-i}\binom{n}{i}$ solutions.

□

5.4.2 On the op-rank shatter function

The context of banned j-ary sequences allows us to work in the op-rank context of [20], which we reframe in terms of set systems. Whereas VC dimension and Littlestone dimension make use of binary trees, op$_s$-rank makes use of 2^s-ary trees.

Definition 5.4.6. A 2^s*-ary element tree* T of height n with labels from X is a labeling of each node $\nu \in (2^s)^{<n}$ by s-tuples $x_\nu = (x_{\nu,0}, \ldots, x_{\nu,s-1})$ from X. A *leaf* of T is an element of $(2^s)^n$. A leaf ξ is properly labeled by a set A if, for all $j < n$ and for all $i < s$, $x_{\xi|_{[j]},i} \in A$ iff $\xi(j)(i) = 1$.

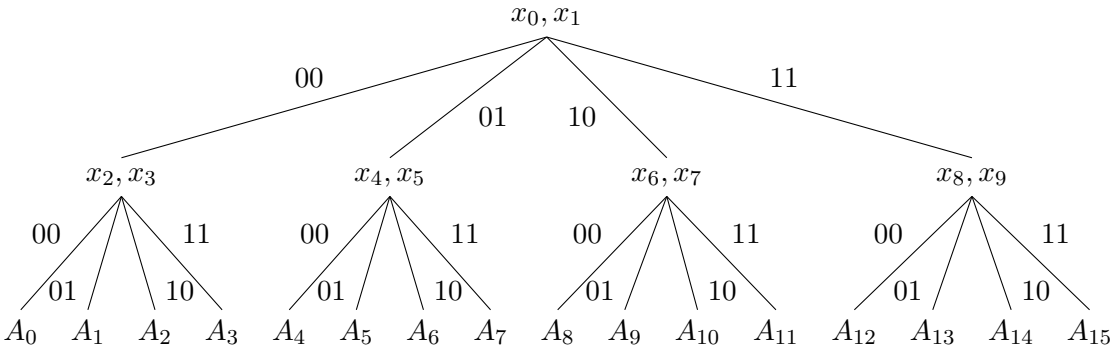

Figure 2: A 2^2-ary element tree of height 2. A_9 properly labels its leaf if it contains x_0 and x_7, but does not contain x_1 and x_6, with no requirements on membership of the other elements.

While this will be the definition that we use in practice, it is often useful think of such trees as binary trees with certain requirements on uniformity of labels within levels.

Definition 5.4.7. An *alternative* 2^s*-ary element tree* T of height n with labels from X is a labeling of $2^{<ns}$ by elements of X such that given any two nodes σ and σ' with labels x_σ and $x_{\sigma'}$, if $|\sigma| = |\sigma'| = l$ and $\sigma|_{s[\lfloor \frac{l}{s} \rfloor]} = \sigma'|_{s[\lfloor \frac{l}{s} \rfloor]}$, then $x_\sigma = x_{\sigma'}$. A *leaf* of T is an element of 2^{ns}, i.e. a binary sequence of length ns. A leaf τ is properly labeled by a set A if, for all $j < ns$, $x_{\tau|_{[j]}} \in A$ iff $\tau(j) = 1$.

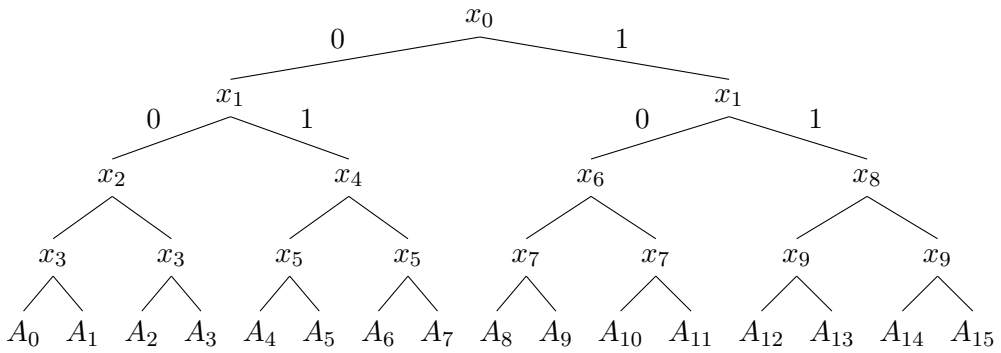

Figure 3: An alternative 2^2-ary element tree of height 2. Observe that the labels on the first two levels are uniform. Then, on the fourth level (and, trivially, the third level), labels are uniform across all nodes with the same initial segment of length 2. We identify 1 with the right branch. As before, A_9 properly labels its leaf if it contains x_0 and x_7, but does not contain x_1 and x_6, with no requirements on membership of the other elements.

Definition 5.4.8. The op_s*-rank* of a set system $(X, \mathcal{F})$, written $\mathrm{opR}_s(X, \mathcal{F})$ or $\mathrm{opR}_s(\mathcal{F})$, is the largest $k < \omega$ such that there is a 2^s-ary element tree of height k with labels from X such that every leaf can be properly labeled by elements of $\mathcal{F}$, or is infinite if there are such trees of arbitrary height. As a convention, we set $\mathrm{opR}_s(\mathcal{F}) = -\infty$ if $\mathcal{F} = \emptyset$. The op_s *shatter function*

$\psi^s_{\mathcal{F}}(n)$ is the maximum number of leaves properly labeled by elements of $\mathcal{F}$ in a 2^s-ary element tree of height n.

It is easy to verify that the op_s-rank and op_s shatter function do not depend on which definition of 2^s-ary element tree we take.

The op_s context is therefore intermediate between the stable context and VC context. Instead of picking labels node by node (as in the stable context) or uniformly for a single level (as in the VC context), we pick labels s at a time. We observe that Littlestone dimension is just the op_1-rank, and VC dimension is the greatest integer s such that the op_s-rank is at least 1.

Likewise, the op_s shatter function is a natural generalization of both the VC and stable shatter functions—observe that the VC shatter function $\pi_{\mathcal{F}}(n)$ is exactly $\psi^n_{\mathcal{F}}(1)$, and the stable shatter function $\rho_{\mathcal{F}}(n)$ is exactly $\psi^1_{\mathcal{F}}(n)$. Although the op-ranks as developed in [20] were indeed intended as a generalization of Littlestone dimension (there referred to as Shelah's 2-rank) and have natural connections with VC dimension, they more strongly considered the geometric properties as they pertained to model theory, and did not study the combinatorics surrounding the shatter function. We study the shatter function here, in particular examining connections between finite op-ranks and growth rates of op-shatter functions.

As before, the dual op_s-rank and dual op_s shatter function of a set system are the op_s-rank and op_s shatter function of the dual set system.

Corollary 5.4.9. *Let $\mathcal{F}$ be a set system with $\mathrm{opR}_s(\mathcal{F}) = k$. Then*

$$\psi^s_{\mathcal{F}}(n) \leq \sum_{i=0}^{k} (2^s - 1)^{n-i} \binom{n}{i}.$$

The proof follows our proof of Theorem 5.2.5, using j-ary banned sequence problems.

Proof. Let T be an 2^s-ary element tree of height n. Identifying the 2^s binary sequences of length s with $[2^s]$, we obtain a hereditary $(k+1)$-fold banned 2^s-ary sequence problem f of length n as follows. Given $S = \{s_0, \ldots, s_k\} \in \binom{[n]}{k+1}$, where $s_0 < s_1 < \cdots < s_k$ and $Y : \bar{S} \to 2^s$, we obtain a 2^s-ary element tree of height $k+1$ by taking all paths $\xi \in (2^s)^n$ through T such that $Y \subset \tau$. Decisions will only be made at nodes ν, where $|\nu| \in S$, so removing all other nodes gives us a 2^s-ary element tree $T_{S,Y}$ of height $k+1$. Since $\mathrm{opR}_s(\mathcal{F}) = k$, not all leaves of $T_{S,Y}$ can be properly labeled, so let $f(S,Y)$ be the set of all sequences whose corresponding leaves in $T_{S,Y}$ can be properly labeled. Then a leaf in T can only be properly labeled if the corresponding sequence is a solution to f.

It remains to show that f is hereditary. Fix $S = \{s_0, \ldots, s_k\}$, and suppose for contradiction that this choice of S witnesses that f is *not* hereditary. Then, for any $Z_\alpha : S \to [2^s]$, there is $Y_\alpha : \bar{S} \to [2^s]$ such that $Z_\alpha \in f(S, Y_\alpha)$. We obtain a complete 2^s-ary tree of height $k+1$ specified by each path $Y_\alpha \sqcup Z_\alpha$ constructed in this manner, restricted to S. Since each Z_α is not banned, we have a 2^s-ary tree of height $k+1$ in which every leaf can be properly labeled, a contradiction.

The result then follows from Theorem 5.4.3. □

The bound of Corollary 5.4.9 can be improved by using more information—in particular, when bounding the op_s shatter function, we can consider op_r-ranks for $r \leq s$. We can already give a better bound for the case where a set system has op_r-rank 0 for some r.

Proposition 5.4.10. *Let $\mathcal{F}$ be a set system with* $\mathrm{opR}_r(\mathcal{F}) = 0$. *Then*

$$\psi^s_{\mathcal{F}}(n) \leq \left(\sum_{i=0}^{r-1} \binom{s}{i} \right)^n$$

Proof. Call a node *live* if it is the initial segment of a leaf that can be properly labeled. At each node of the tree, we consider s elements. Observing that $\mathrm{opR}_r(\mathcal{F}) = 0$ says precisely that the VC dimension of $\mathcal{F}$ is strictly less than r, Theorem 5.2.2 tells us that we can find sets which properly label at most $\sum_{i=0}^{r-1} \binom{s}{i}$ of the possible boolean combinations of the s elements. That is, each live node has at most $\sum_{i=0}^{r-1} \binom{s}{i}$ live successors in the next level. Therefore, there are at most $\left(\sum_{i=0}^{r-1} \binom{s}{i} \right)^m$ live nodes at the level of height m (counting from 0). Since leaves in a tree of height n appear at the nth level, the result follows. □

The set system of half-spaces in $\mathbb{R}^r$ achieves the bound of Proposition 5.4.10 for the *dual* op_s shatter function. (This is the famous cake-cutting problem.)

Proposition 5.4.11. *Let $\mathcal{F}$ be the dual set system to the set system of $\mathbb{R}^r$ consisting of half-spaces. Then*

$$\psi^s_{\mathcal{F}}(n) = \left(\sum_{i=0}^{r} \binom{s}{i} \right)^n.$$

In particular, $\mathrm{opR}_{r+1}(\mathcal{F}) = 0$.

Proof. It suffices to verify that taking s hyperplanes in general position (i.e. so that any m hyperplanes intersect in a $(r-m)$-dimensional subspace) partitions $\mathbb{R}^r$ into $\sum_{i=0}^{r} \binom{s}{i}$ pieces, each of which contains an open set (in the Euclidean topology). Such a partition corresponds

to one level in the 2^s-ary tree. Each piece may then be partitioned further in the same manner for each successive level of the tree.

We proceed by induction. The $s = 1$ case is obvious, for all r. The $r = 1$ case is obvious, for all s.

Consider the $s+1$ and $r+1$ case. Removing one of the $s+1$ hyperplanes, we have $\sum_{i=0}^{r+1} \binom{s}{i}$ pieces by induction. Restore the hyperplane that we removed. Viewing that hyperplane as a copy of $\mathbb{R}^r$, it is partitioned into $\sum_{i=0}^{r} \binom{s}{i}$ pieces by the other hyperplanes, by induction. Each such piece corresponds to a piece in $\mathbb{R}^{r+1}$ which is cut into two pieces by the restored hyperplane. We therefore find that the total number of pieces is

$$\sum_{i=0}^{r+1} \binom{s}{i} + \sum_{i=0}^{r} \binom{s}{i} = \sum_{i=0}^{r+1} \binom{s+1}{i}.$$

as desired. □

We can further refine our methods. Fix a base set X. We identify any set system $(X, \mathcal{F})$ on X with $\mathcal{F}$.

Proposition 5.4.12. *1. Let $\mathcal{F}_1 \subseteq \mathcal{F}_2$. Then, for any s, $\mathrm{opR}_s(\mathcal{F}_1) \leq \mathrm{opR}_s(\mathcal{F}_2)$.*

2. Let $s_1 < s_2$. Then $\mathrm{opR}_{s_1}(\mathcal{F}) \geq \lfloor \frac{s_2}{s_1} \rfloor \mathrm{opR}_{s_2}(\mathcal{F})$.

Proof. (1) is trivial. For (2), suppose that we have a 2^{s_2}-ary element tree T of height $n_2 := \mathrm{opR}_{s_2}(\mathcal{F})$, with labels $x_\nu = (x_{\nu,0}, \ldots, x_{\nu,s_2-1})$ for each $\nu \in (2^{s_2})^{<n_2}$, in which every leaf can be properly labeled. Then we can obtain a 2^{s_1}-ary element tree T' of height $n_1 := \lfloor \frac{s_2}{s_1} \rfloor n_2$ in which

every leaf can be properly labeled. Let $t = \lfloor \frac{s_2}{s_1} \rfloor$. Intuitively, we split each level of the 2^{s_2}-ary tree into t levels of the 2^{s_1}-ary tree, with any label $x_\nu = (x_{\nu,0}, \ldots, x_{\nu,s_2-1})$ splitting into t labels

$$(x_{\nu,0}, \ldots, x_{\nu,s_1-1}), (x_{\nu,s_1}, x_{\nu,2s_1-1}), \ldots, (x_{\nu,(t-1)s_1}, \ldots, x_{\nu,ts_1-1}).$$

More formally, suppose $\xi \in (2^{s_1})^i$, for $i < n_1$. Suppose $i = jt + k$, for $0 \leq k < t$. Then label ξ with

$$x_\xi = (x_{\nu_\xi,ks_1}, \ldots x_{\nu_\xi,(k+1)s_1-1}),$$

where $\nu_\xi \in (2^{s_2})^j$ is as follows. Let $\sigma_l = \xi(l) \in 2^{s_1}$. Then let $\tau_m \in 2^{s_2}$ be the concatenation of $\sigma_{mt}, \ldots, \sigma_{(m+1)t-1}$, appending as many 0s as needed to obtain a sequence of length s_2. Then let

$$\nu_\xi := (\tau_0, \ldots, \tau_{j-1}).$$

Then the labeling of T' by the x_ξ gives a 2^{s_1}-ary tree of height n_1 in which every leaf can be properly labeled (in particular, by one of the labels of the leaves of the 2^{s_2}-ary tree). □

(2) shows how different finite op_s ranks can interact; in particular, a finite op_s rank establishes upper bounds on $\mathrm{op}_{s'}$ ranks, for $s < s'$. (2) above is somewhat easier to see using the alternative definition—we simply view the tree as a 2^{s_1}-ary tree instead of a 2^{s_2}-ary tree, possibly after removing some levels. Figure 3 is the 2^1-ary tree obtained from Figure 2 by this process.

Given $\mathcal{F}$, $x_0, \ldots, x_{s-1} \in X$, and $\sigma : [s] \to 2$, let

$$\mathcal{F}_\sigma := \{Y \in \mathcal{F} \,|\, \text{for all } i < n,\, x_i \in Y \text{ iff } \sigma(i) = 1\}.$$

Call each $\mathcal{F}_\sigma$ a child of $\mathcal{F}$. Then, in an op_s-tree with root $(x_0, \ldots, x_{s-1})$, $\mathcal{F}_\sigma$ consists of all sets in $\mathcal{F}$ which properly label a leaf whose path begins with σ. Observe that if for all $\sigma : [s] \to 2$, $\mathrm{opR}_s(\mathcal{F}_\sigma) \geq a$, then $\mathrm{opR}_s(\mathcal{F}) \geq a + 1$; we can obtain a 2^s-ary tree of height $a + 1$ by labeling the root with $(x_0, \ldots, x_{s-1})$, and appending 2^s-ary trees of height a witnessing $\mathrm{opR}_s(\mathcal{F}_\sigma) \geq a$ at the appropriate successor nodes.

The following lemma generalizes the observation that, given $\mathcal{F}$ with Littlestone dimension $a < \infty$ and any $x \in X$, at most one of $\{F \in \mathcal{F} \,|\, x \in F\}$ and $\{F \in \mathcal{F} \,|\, x \notin F\}$ has Littlestone dimension a; if both had Littlestone dimension a, joining the two binary element trees witnessing this with root x would witness that $\mathcal{F}$ has Littlestone dimension $a + 1$.

Lemma 5.4.13. *Suppose* $\mathrm{opR}_r(\mathcal{F}) = a < \infty$. *Then, given any* $x_0, \ldots, x_{s-1} \in X$, *we have* $\mathrm{opR}_r(X_\sigma) \leq a - 1$ *for at least* $2^s - \sum_{i=0}^{r-1} \binom{s}{i}$ *children* $\mathcal{F}_\sigma$. *More generally, we have* $\mathrm{opR}_r(X_\sigma) \leq a - l$ *for at least* $2^s - \sum_{i=0}^{lr-1} \binom{s}{i}$ *children* $\mathcal{F}_\sigma$.

Proof. We obtain an independent r-fold banned binary sequence problem f of length s as follows. For each $S \in \binom{[s]}{r}$, let $f(S)$ be those functions $\eta : S \to 2$ such that $\mathrm{opR}_r(\mathcal{F}_\eta) \leq a - 1$, where

$$\mathcal{F}_\eta := \{Y \in \mathcal{F} \,|\, \text{for all } i \in S,\, x_i \in Y \text{ iff } \eta(i) = 1\}.$$

Each $f(S)$ is nonempty, or else those $\mathcal{F}_\eta$ witness that that $\mathrm{opR}_r(\mathcal{F}) \geq a+1$, a contradiction. Then $\sigma : [s] \to 2$ is banned by f if there is some $S \in \binom{[s]}{r}$ such that $\mathrm{opR}_r(\mathcal{F}_{\sigma_S}) \leq a-1$, whence $\mathrm{opR}_r(\mathcal{F}_\sigma) \leq a-1$. So sequences banned by f have the corresponding child drop in op_r-rank, of which there are at least $2^s - \sum_{i=0}^{r-1} \binom{s}{i}$ many.

For the more general case, we instead obtain an independent lr-fold banned binary sequence problem. For each $S \in \binom{[s]}{lr}$, let $f(S)$ be those $\eta : S \to 2$ such that $\mathrm{opR}_r(\mathcal{F}_\eta) \leq a-l$. Each $f(S)$ is nonempty, or else those $\mathcal{F}_\eta$ witness that $\mathrm{opR}_r(\mathcal{F}) \geq a+1$. Then sequences banned by f have the corresponding child drop in op_r-rank by at least l, of which there are at least $2^s - \sum_{i=0}^{lr-1} \binom{s}{i}$ many. □

The boundary between finite and infinite op-ranks serves as an important parameter in obtaining better bounds. It is also of model-theoretic interest, coinciding with other known properties.

Definition 5.4.14. The op-*dimension* of a set system $\mathcal{F}$ is

$$\sup\{r \mid \mathrm{opR}_r(\mathcal{F}) = \infty\}.$$

Expressed in model-theoretic terms, the op-dimension of a (type-)definable set X in some model is the supremum of the op-dimension of set systems on X generated finite sets of formulas. In this context, op-dimension coincides with o-minimal dimension in o-minimal theories and dp-rank in distal theories [20].

We use Lemma 5.4.13 to obtain better bounds on the op_s shatter function by using op-dimension.

Definition 5.4.15. Let $\psi^s_{r,b}(n)$ be the greatest possible number of properly labeled leaves in a 2^s-ary tree of height n by any set system $\mathcal{F}$ with $\mathrm{opR}_r(\mathcal{F}) \leq b < \omega$.

Theorem 5.4.16. *Let* $a_0 := \sum_{i=0}^{r-1} \binom{s}{i}$ *and* $a_1 = 2^s - a_0$. *Then*

$$\psi^s_{r,b}(n) \leq \sum_{i=0}^{b} \binom{n}{i} a_0^{n-i} a_1^i.$$

Proof. The case $n = 0$ is trivial for all b. We proceed by induction on b. The case $b = 0$ is Proposition 5.4.10.

For the case $b + 1$, we observe that, by monotonicity of $\psi^s_{r,b}(n)$ in b, we maximize the possible number of properly labeled leaves by having as many children as possible not decrease in op_r-rank. We now proceed by induction on n. By Lemma 5.4.13, we can have at most a_0

such children, and the remaining a_1 children must drop in op_r-rank by at least 1. We therefore obtain the recurrence

$$\begin{aligned}
\psi^s_{r,b+1}(n) &\le a_0\psi^s_{r,b+1}(n-1) + a_1\psi^s_{r,b}(n-1) \\
&\le a_0 \sum_{i=0}^{b+1} \binom{n-1}{i} a_0^{n-i-1}a_1^i + a_1 \sum_{i=0}^{b} \binom{n-1}{i} a_0^{n-i-1}a_1^i \quad \text{by induction} \\
&\le \sum_{i=0}^{b+1} \binom{n-1}{i} a_0^{n-i}a_1^i + \sum_{i=0}^{b} \binom{n-1}{i} a_0^{n-i-1}a_1^{i+1} \\
&\le \binom{n-1}{0} a_0^n + \sum_{i=1}^{b+1} \binom{n-1}{i} a_0^{n-i}a_1^i + \sum_{i=1}^{b+1} \binom{n-1}{i-1} a_0^{n-i}a_1^i \\
&\le \binom{n}{0} a_0^n + \sum_{i=1}^{b+1} \binom{n}{i} a_0^{n-i}a_1^i \\
&\le \sum_{i=0}^{b+1} \binom{n}{i} a_0^{n-i}a_1^i
\end{aligned}$$

as desired.

□

In particular, for a set system with op-dimension d, we take $r = d+1$. Then the op shatter function is bounded by an exponential function with the base a_0 determined by d. Furthermore, coefficients for lower order terms can be improved when $r \le \frac{s}{2}$, as then the more general case of Lemma 5.4.13 dictates that some children must drop in op_r-rank by more than 1. This creates a more complicated recurrence, but the result remains exponential in a_0.

Finally, we observe that we can recover both the VC and stable Sauer-Shelah bounds from Theorem 5.4.16. If $\mathcal{F}$ has VC dimension r, then $\mathrm{opR}_{r+1}(\mathcal{F}) = 0$. Then

$$\pi_{\mathcal{F}}(s) = \psi^s_{\mathcal{F}}(1) \leq \psi^s_{r+1,0}(1) \leq \sum_{i=0}^{r} \binom{s}{i}.$$

Similarly, if $\mathcal{F}$ has Littlestone dimension b, this says that $\mathrm{opR}_1(\mathcal{F}) = b$. Then

$$\rho_{\mathcal{F}}(n) = \psi^1_{\mathcal{F}}(n) \leq \psi^1_{1,b}(n) \leq \sum_{i=0}^{b} \binom{n}{i}.$$

CITED LITERATURE

[1] Dana Angluin. Learning regular sets from queries and counterexamples. *Information and computation*, 75(2):87–106, 1987.

[2] Dana Angluin. Queries and concept learning. *Machine learning*, 2(4):319–342, 1988.

[3] Dana Angluin. Negative results for equivalence queries. *Machine Learning*, 5(2):121–150, 1990.

[4] Dana Angluin and Tyler Dohrn. The power of random counterexamples. In *International Conference on Algorithmic Learning Theory*, pages 452–465, 2017.

[5] Dana Angluin and Dana Fisman. Learning regular omega languages. *Theoretical Computer Science*, 650:57–72, 2016.

[6] Dana Angluin and Dana Fisman. Regular omega-languages with an informative right congruence. *arXiv preprint arXiv:1809.03108*, 2018.

[7] Peter Auer and Philip M Long. Simulating access to hidden information while learning. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 263–272. ACM, 1994.

[8] José L. Balcázar, Jorge Castro, David Guijarro, and Hans Ulrich Simon. The consistency dimension and distribution-dependent learning from queries. *Theoretical Computer Science*, 288(2):197–215, 2002.

[9] Shai Ben-David and Ami Litman. Combinatorial variability of Vapnik-Chervonenkis classes with applications to sample compression schemes. *Discrete Applied Mathematics*, 86(1):3–25, 1998.

[10] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *Proceedings of the 22nd Annual Conference on Learning Theory (COLT)*, 2009.

[11] Siddharth Bhaskar. Thicket density. *arXiv preprint arXiv:1702.03956*, 2017.

[12] Hunter Chase and James Freitag. Model theory and combinatorics of banned sequences. *arXiv preprint arXiv:1801.07640, to appear, Journal of Symbolic Logic*, 2018.

[13] Hunter Chase and James Freitag. Bounds in query learning. *arXiv preprint arXiv:1904.10122*, 2019.

[14] Hunter Chase and James Freitag. Model theory and machine learning. *Bulletin of Symbolic Logic*, 25(3):319–332, 2019.

[15] Artem Chernikov, Daniel Palacin, and Kota Takeuchi. On n-dependence. *arXiv preprint arXiv:1411.0120*, 2014.

[16] Dana Fisman. Inferring regular languages and ω-languages. *Journal of Logical and Algebraic Methods in Programming*, 98:27–49, 2018.

[17] Dana Fisman, Udi Boker, and Dana Angluin. Families of DFAs as acceptors of ω-regular languages. *Logical Methods in Computer Science*, 14, 2018.

[18] Sally Floyd and Manfred Warmuth. Sample compression, learnability, and the Vapnik-Chervonenkis dimension. *Machine learning*, 21(3):269–304, 1995.

[19] Vincent Guingona. NIP theories and computational learning theory. *https://tigerweb.towson.edu/vguingona/NIPTCLT.pdf*.

[20] Vincent Guingona and Cameron Donnay Hill. On a common generalization of Shelah's 2-rank, dp-rank, and o-minimal dimension. *Annals of Pure and Applied Logic*, 166(4):502–525, 2015.

[21] Lisa Hellerstein, Krishnan Pillaipakkamnatt, Vijay Raghavan, and Dawn Wilkins. How many queries are needed to learn? *Journal of the ACM*, 43(5):840–862, 1996.

[22] Yoshiyasu Ishigami and Sei'ichi Tani. VC-dimensions of finite automata and commutative finite automata with k letters and n states. *Discrete Applied Mathematics*, 74(2):123–134, 1997.

[23] Hunter R Johnson and Michael C Laskowski. Compression schemes, stable definable families, and o-minimal structures. *Discrete & Computational Geometry*, 43(4):914–926, 2010.

[24] Michael C Laskowski. Vapnik-Chervonenkis classes of definable sets. *Journal of the London Mathematical Society*, 2(2):377–384, 1992.

[25] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1988.

[26] Nick Littlestone and Manfred Warmuth. Relating data compression and learnability. Technical report, University of California, Santa Cruz, 1986.

[27] Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

[28] Wolfgang Maass and György Turán. On the complexity of learning from counterexamples and membership queries. In *Foundations of Computer Science, 1990. Proceedings., 31st Annual Symposium on*, pages 203–210. IEEE, 1990.

[29] Maryanthe Malliaris and Saharon Shelah. Regularity lemmas for stable graphs. *Transactions of the American Mathematical Society*, 366:1551–1585, 2014.

[30] Maryanthe Malliaris and Caroline Terry. On unavoidable induced subgraphs in large prime graphs. *Journal of Graph Theory, accepted*, 2017.

[31] David Marker. *Model Theory: an Introduction*. Springer, Graduate Texts in Mathematics, 217, Second Edition, 2002.

[32] Robert McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9(5):521–530, 1966.

[33] Shay Moran and Amir Yehudayoff. Sample compression schemes for VC classes. *Journal of the ACM (JACM)*, 63(3):21, 2016.

[34] Anil Nerode. Linear automaton transformations. *Proceedings of the American Mathematical Society*, 9(4):541–544, 1958.

[35] Hung Q. Ngo. Three proofs of Sauer-Shelah lemma. *Course notes, https://www.cse.buffalo.edu/~hungngo/classes/2010/711/lectures/sauer.pdf*, 2010.

[36] Nir Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. In *Logic in Computer Science, 2006 21st Annual IEEE Symposium on*, pages 255–264. IEEE, 2006.

[37] Robert E Schapire. The design and analysis of efficient learning algorithms. Technical report, Massachusetts Institute of Technology Lab for Computer Science, 1991.

[38] Saharon Shelah. *Classification theory and the number of non-isomorphic models. Studies in Logic and the Foundations of Mathematics*. Volume 92, North-Holland Publishing Company, New York, 1978.

[39] Saharon Shelah. *Classification theory: and the number of non-isomorphic models*, volume 92. Elsevier, 1990.

[40] Pierre Simon. *A guide to NIP theories*. Cambridge University Press, 2015.

[41] K. Tent and M. Ziegler. *A Course in Model Theory*. Lecture Notes in Logic. Cambridge University Press, 2012.

APPENDIX

The following statement appears on `https://www.cambridge.org/about-us/rights-permissions/faqs` and governs re-use of material appearing in Chapter 2:

> In certain circumstances, permissions requests are not required from authors who wish to re-use original material they have written for a Cambridge publication, provided that the subsequent use includes a full acknowledgement of the original publication, together with the copyright notice and the phrase 'Reprinted with permission'.
>
> Permissions requests are waived if:
>
> - The author of the work wishes to reproduce a single chapter (not exceeding 20 per cent of their work), journal article or shorter extract in a subsequent work (i.e. with a later publication date) of which he or she is to be the author, co-author or editor.

VITA

NAME	Hunter Sato Chase
EDUCATION	Ph.D, Mathematics, University of Illinois at Chicago, Chicago, IL, 2020.
	M.S., Mathematics, University of Illinois at Chicago, Chicago, IL, 2016.
	B.S., Mathematics, The University of Chicago, Chicago, IL, 2014.
TEACHING	Teaching Assistant, University of Illinois at Chicago. 2015–2019.
AWARDS	Research and Training Grant Pre-doctoral Fellow, University of Illinois at Chicago. 2014–2015, 2018–2020.
PUBLICATIONS	Hunter Chase and James Freitag. Model theory and machine learning. *Bulletin of Symbolic Logic*, 25(3): 319–332, 2019. DOI: 10.1017/bsl.2018.71
PREPRINTS	Hunter Chase and James Freitag. Bounds in query learning. *arXiv preprint arXiv:1904.10122*, 2019.
	Hunter Chase and James Freitag. Model theory and combinatorics of banned sequences. *arXiv preprint arXiv:1801.07640*, 2018. Accepted, *Journal of Symbolic Logic*.