



Applications of Littlestone Dimension to Query Learning and to Compression

Hunter Chase  

Department of Mathematics, Statistics, and Computer Science,
University of Illinois at Chicago, IL, USA

James Freitag  

Department of Mathematics, Statistics, and Computer Science,
University of Illinois at Chicago, IL, USA

Lev Reyzin  

Department of Mathematics, Statistics, and Computer Science,
University of Illinois at Chicago, IL, USA

Abstract

In this paper we give several applications of Littlestone dimension. The first is to the model of Angluin and Dohrn [1], where we extend their results for learning by equivalence queries with random counterexamples. Second, we extend that model to infinite concept classes with an additional source of randomness. Third, we give improved results on the relationship of Littlestone dimension to classes with extended d -compression schemes, proving the analog of a conjecture of Floyd and Warmuth [4] for Littlestone dimension.

2012 ACM Subject Classification Theory of computation → Query learning; Theory of computation → Randomness, geometry and discrete structures; Mathematics of computing → Combinatoric problems

Keywords and phrases compression scheme, query learning, random queries, Littlestone dimension

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.42

Related Version *Full Version*: <https://arxiv.org/abs/2310.04812>

Funding This research was supported in part by award ECCS-2217023 from the National Science Foundation and National Science Foundation CAREER award 1945251.

Acknowledgements The authors wish to thank Shai Ben-David for comments on an early draft.

1 Introduction

In query learning, a learner attempts to identify an unknown concept from a collection via a series of data requests called queries. Typically, algorithms designed for learning in this setting attempt to bound the number of required queries to identify the target concept in the worst-case scenario. If one imagines the queries of the learner being answered by a teacher, the usual setup imagines the teacher answering queries in an adversarial manner, with minimally informative answers. Alternatively, for a given algorithm, the bounds for the traditional model are on the *worst-case answers over all potential targets*. In variations of the model, one of these two factors is usually modified.

For instance, Kumar, Chen, and Singla [7] study the case in which the answers are assumed to be maximally informative in a certain sense. In this manuscript, we first work in the setup originating with Angluin and Dohrn [1], where we assume that the answers to the queries are randomly selected with respect to some fixed probability distribution.



© Hunter Chase, James Freitag, and Lev Reyzin;
licensed under Creative Commons License CC-BY 4.0

49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024).

Editors: Rastislav Kráľovič and Antonín Kučera; Article No. 42; pp. 42:1–42:10

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Consider a concept class $\mathcal{C} = \{C_1, \dots, C_n\}$, subsets of a fixed set X . Fix a target concept $A \in \mathcal{C}$. An *equivalence query* consists of the learner submitting a hypothesis $B \in \mathcal{C}$ to a teacher, who either returns *yes* if $A = B$, or a counterexample $x \in A \Delta B$. 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.

Angluin and Dohrn [1] fix a probability distribution μ on X and assume that the teacher selects the counterexamples randomly with respect to μ restricted to $A \Delta B$. They show that for a concept class \mathcal{C} 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 combinatorial notion of dimension which can be used to bound the expected number of queries independent of the size of the class - perhaps even in infinite classes. In fact, Angluin and Dohrn [1] (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. On the other hand, Angluin and Dohrn [1] (Theorem 26), using an example of [9], show that the VC-dimension *cannot* provide an upper bound for the number of queries.

The motivation for bounds depending on some notion of dimension rather than the number of concepts is two-fold:

- Many combinatorial notions of dimension (e.g. Littlestone or VC) of a class \mathcal{C} can be small while $|\mathcal{C}|$ is large.
- Investigating this model of learning in settings where \mathcal{C} is an infinite class will require methods and bounds that do not use $|\mathcal{C}|$.

Roughly speaking, the *Littlestone dimension* (or Ldim) [9] of a concept class \mathcal{C} over domain X is the maximal depth of a complete binary decision tree T such that T 's nodes are associated with elements of X and T 's edges are associated with binary labels such that each root-to-leaf path in T agrees in the labeling of its respective elements with some concept in \mathcal{C} . If we require all nodes of T at the same depth to be associated with the same element of X , this yields the definition of VC dimension.

We show that the Littlestone dimension provides such an upper bound; we give an algorithm that yields a bound that is linear in the Littlestone dimension for the expected number of queries needed to learn any concept. In Section 2 we establish the bounds for finite concept classes \mathcal{C} .

In Section 3 we give a specific example that shows finite Littlestone dimension of an infinite class \mathcal{C} is not sufficient to guarantee the learnability of the class in the model of Angluin and Dohrn [1]. That is, we show the expected number of queries is impossible to bound over all target concepts, even in very simple infinite classes. Suppose that the target concept is itself selected randomly with respect to some (perhaps unrelated to the feedback mechanism) probability distribution. In this case, we give an algorithm so that the expected number of queries (over both sources of randomness) is at most $\tilde{O}(d)$ where d is the Littlestone dimension of the class \mathcal{C} . This result uses the bounds developed in Section 2 in an essential way, in particular by using the finite class's Littlestone dimension instead of its size.

In Section 4, we give another application of Littlestone dimension - to compression schemes, which answers a question of Johnson and Laskowski [6] on d -compression with b extra bits, a notion originating with Floyd and Warmuth [4]. The existence of a d -compression is closely related to various notions of learning; d -compressibility of a class \mathcal{C} implies the class has VC-dimension at most d . A famous conjecture of Floyd and Warmuth [4] asks if

every VC-class has a d -compression where d is the VC-dimension.¹ Our result in Section 4 proves a strong version of the conjecture for Littlestone dimension. In Section 4 we also explain some of the many variants of this problem which have been previously solved.

2 Random counterexamples and EQ-learning

In this section, we essentially work in the setting of Angluin and Dohrn [1] with 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 2.1.** 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 $\{A \in \mathcal{C} \mid A(x_j) = i_j, j = 1, \dots, n\}$. For $A \in \mathcal{C}$ and $a \in X$, we let

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

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 2.2.** 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 that is similar to the *elimination graph* of Angluin and Dohrn [1].

► **Definition 2.3.** 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 $\text{Ldim}(\mathcal{C}) - \text{Ldim}(\mathcal{C}_{x=B(x)})$ over $x \in \Delta(A, B)$ with respect to the distribution $\mu|_{\Delta(A, B)}$.²

► **Definition 2.4.** The query rank of $A \in \mathcal{C}$ is defined as: $\inf_{B \in \mathcal{C}} (d(A, B))$.

► **Lemma 2.5.** 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 2.2:

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

► **Definition 2.6** (Angluin and Dohrn [1], 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, \dots, 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 Resolving whether there is an $O(d)$ compression has a reward of 600 dollars [12].

² 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 .

42:4 Applications of Littlestone Dimension to Query Learning and to Compression

The next result is similar to Theorems 16 (the case $l = 3$) and Theorem 17 (the case $l > 3$) of Angluin and Dohrn [1], but our proof is rather different (note that the case $l = 2$ follows easily from Lemma 2.5).

► **Theorem 2.7.** *The thicket query graph $G_{TQ}(\mathcal{C}, \mu)$ has no degenerate l -cycles for $l \geq 2$.*

The analogue of Theorem 16 of Angluin and Dohrn [1] can be adapted in a very similar manner to the technique employed by them. However, the analogue of the proof of Theorem 17 of Angluin and Dohrn [1] falls apart in our context; the reason is that Lemma 2.2 is analogous to their Lemma 6 (and Lemma 2.5 is analogous to their Lemma 13), but our lemmas involve inequalities instead of equations. The inductive technique of Angluin and Dohrn [1, 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 Angluin and Dohrn [1].

Proof. Suppose the vertices in the degenerate l -cycle are A_0, \dots, A_{l-1} . By the definition of degenerate cycles and $d(-, -)$, we have, for each $i \in \mathbb{Z}/l\mathbb{Z}$, that

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

Clearing the denominator we have

$$\sum_{a \in \Delta(A_i, A_{i+1})} \mu(a) u(A_i, a) \leq \frac{1}{2} \mu(\Delta(A_i, A_{i+1})). \quad (2.1)$$

Note that throughout this argument, the coefficients are being calculated modulo l . Notice that for at least one value of i , the inequality in 2.1 must be strict.

Let G, H be a partition of

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

Now define

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

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

► **Fact 2.8.** *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 2.1 as i ranges from 1 to l . On the LHS 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 RHS 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 $\{A_1, \dots, A_l\}$ 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}, \dots, 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 2.2, 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 RHS, we have

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

For each G, H a partition of X , the terms appearing in the above sum occur in pairs as above by Fact 2.8, and so, we have the LHS is at least as large as the RHS of the sum of inequalities 2.1, which is impossible since one of the inequalities must have been strict by our degenerate cycle. ◀

► **Theorem 2.9.** *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 $B, f(B), \dots, f^n(B) = B$. By construction, this yields a bad cycle, contradicting Theorem 2.7. ◀

2.1 The thicket max-min algorithm

In this subsection we show how to use the lower bound on query rank proved in Theorem 2.9 to give an algorithm that 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 Angluin and Dohrn [1, 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

$$A = \operatorname{argmax}_{A \in \mathcal{C}_i} (\min_{B \in \mathcal{C}_i} d_{\mathcal{C}_i}(A, B)).$$

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

Proof. The expected drop in the Littlestone dimension of the concept class induced by any query before the algorithm terminates is at least $1/2$ by Theorem 2.9; so the probability that the drop in the Littlestone dimension is positive is at least $1/2$ for any given query. So, from $2n$ queries, one expects at least n drops in Littlestone dimension, at which point the class is learned. ◀

3 Equivalence queries with random counterexamples and random targets

Let \mathcal{C} consist the collection of intervals $\left\{ \left(\frac{1}{n+1}, \frac{1}{n} \right) \mid n \in \mathbb{N} \right\}$ with μ the Lebesgue measure on the unit interval. This concept class has Littlestone dimension one since any two concepts are disjoint. There is no upper bound on the number of expected queries (using the model with random counterexamples of the previous section) that is uniform over all targets.

To see why, suppose the learner guesses interval $\left(\frac{1}{n+1}, \frac{1}{n} \right)$ for some n . For any $\epsilon > 0$ there is $N \in \mathbb{N}$ such that with probability greater than $1 - \epsilon$, the learner gets a counterexample from the interval they guessed, $\left(\frac{1}{n+1}, \frac{1}{n} \right)$. Of course, even with this additional information, no matter the learner's guess at any stage at which they have received only negative counterexamples, this is clearly still the case. Thus, there can be no bound on expected queries which is uniform over all target concepts.

In this section we introduce an additional source of randomness, which allows for learning over infinite classes \mathcal{C} .³ So, suppose \mathcal{C} is a (possibly infinite) set of concepts on a set X . Suppose that we have probability measures μ on X and τ on \mathcal{C} . Suppose a target $A \in \mathcal{C}$ is selected randomly according to the distribution τ and the counterexamples to equivalence queries are selected randomly according to the distribution μ .

► **Theorem 3.1.** *Suppose that \mathcal{C} is countable with finite Littlestone dimension d . There is an algorithm such that the expected number of queries over distributions μ on X and τ on \mathcal{C} is at most $\tilde{O}(d)$.*

Proof. Let $\epsilon_k = \frac{1}{2^{k+1}}$ for $k \in \mathbb{N}$. The idea of the algorithm is to run our earlier algorithm on a $1 - \epsilon_k$ fraction of the concepts with respect to the measure τ .

At stage k of the algorithm, we observe the following. Since \mathcal{C} is countable, enumerate the collection $\mathcal{C} = \{C_i\}_{i \in \mathbb{N}}$. Then since $\sum_{i=1}^{\infty} P(C_i) = 1$, for any $\epsilon_k > 0$, there is $N_k = N(\epsilon_k) \in \mathbb{N}$ such that $\sum_{i=1}^{\infty} P(C_i) \geq 1 - \epsilon_k$.

Conditional on the target being among the first N_k concepts, the next idea is to run the algorithm from the previous section on this finite set for n steps where n is such that the probability that we have not identified the target after n steps is less than ϵ , for some $0 < \epsilon < 1$. This number $n = n_{d,\epsilon}$ depends only on the Littlestone dimension and ϵ , but not on N as we will explain.

We now bound the probability that the algorithm has not terminated after n steps, conditional on the target being in the first N_k many concepts. Since at any step, the probability that the Littlestone dimension drops is at least $\frac{1}{2}$ by Theorem 2.9, the probability that the algorithm has not terminated after n steps is at most the probability of a binomial random variable with probability $\frac{1}{2}$ achieving at most $d - 1$ successes in n attempts, which is

³ One might also think of the random EQ learning of Angluin and Dohrn as analyzing the maximum number of expected number of queries over all possible targets, while our model will analyze the *expected* number of queries where the expectation is taken over the concepts (with a fixed but arbitrary distribution) and over the counterexamples.

$$\sum_{k=0}^{d-1} \binom{n}{k} \left(\frac{1}{2}\right)^n \leq n^d/2^n.$$

Note that $n^d/2^n < \epsilon$ whenever $n - d \log n > \log(\frac{1}{\epsilon})$. Hence,

$$n \geq \tilde{O}(d + \log(1/\epsilon))$$

is sufficient.

So at stage k , we run the algorithm for n steps as specified above. Either the target concept is found or we continue to stage $k + 1$ on the larger concept class N_k . Since

$$(1 - \epsilon_1) \left(\sum_{k=1}^{\infty} \epsilon_k \right) = 1/2 \sum_{k=1}^{\infty} 1/2^{k+1} < 1,$$

the expected total number of queries is still bounded by $\tilde{O}(d + \log(1/\epsilon))$.⁴ ◀

4 Compression schemes and stability

In this section, we follow the notation and definitions given in Johnson and Laskowski [6] on *compression schemes*, a notion due to Littlestone and Warmuth [10]. 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, including as size d -array compression [2], extended compression schemes with b extra bits [4], and as unlabeled compression schemes [8]. In the definitions below, we let $\text{dom}(f)$ denote the domain of f and \mathcal{C}_{fin} the restriction of \mathcal{C} to finite subsets.

The following definition gives the notion of compression we consider within this section; the notion is equivalent to the notion of a d -compression with b extra bits [4]. The equivalence of these two notions is proved by Johnson and Laskowski [6, Proposition 2.1]. In our compression schemes, the role of the b extra bits is played by the reconstruction functions, and of course, the number of extra bits can be bounded in terms of the number of reconstruction functions (and vice versa). Of course, one is interested in optimizing both the size of the compression and the number of reconstruction functions (extra bits) in general.

► **Definition 4.1.** *We say that a concept class \mathcal{C} has a d -compression if there is a compression function $\kappa : \mathcal{C}_{\text{fin}} \rightarrow X^d$ and a finite set \mathcal{R} of reconstruction functions $\rho : X^d \rightarrow 2^X$ such that for any $f \in \mathcal{C}_{\text{fin}}$*

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

We work with the above notion mainly because it is the notion used in Johnson and Laskowski [6], and our goal is to improve a result therein. That result was later improved by Laskowski and appears in the unpublished notes of Guingona [5] (Theorem 4.1.3). When the original work on this result was completed, we were not aware of the work of Guingona [5], but as it turns out, our result improves both of these (the latter uses exponentially many reconstruction functions, while we use linearly many).

⁴ There isn't anything particularly special about the sequence ϵ_k that we chose. Any sequence (ϵ_k) going to zero whose sum converges can be seen to work in the algorithm and affects only the constants in the expected number of steps, which we are not optimizing.

Johnson and Laskowski [6] prove that a concept class with finite Littlestone dimension has an extended d -compression for some d .⁵ The precise value of d is not determined there, but was conjectured to be the Littlestone dimension. In Theorem 4.4, we will show that d can be taken to be the Littlestone dimension and $d + 1$ many reconstruction functions suffice.⁶

The question in Johnson and Laskowski [6] is the analogue (for Littlestone dimension) of a well-known open question from VC-theory [4]: 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 a bound that is at most exponential in d [11].

► **Definition 4.2.** Suppose $\text{Ldim}(\mathcal{C}) = d$. Given a partial function f , say that f is exceptional for \mathcal{C} if for all $a \in \text{dom}(f)$,

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

has Littlestone dimension d .

► **Definition 4.3.** Suppose $\text{Ldim}(\mathcal{C}) = d$. Let $f_{\mathcal{C}}$ be the partial function given by

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

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

► **Theorem 4.4.** 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 $f \in \mathcal{C}_{\text{fin}}$ 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

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

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

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

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

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

⁵ Their result is formulated for the sets of realizations of first-order formulas that are *stable*, but their proofs work for general concept classes, and Chase and Freitag [3] explain that stability is equivalent to finite Littlestone dimension.

⁶ After proving this, we became aware of the unpublished result of Laskowski appearing as [5, Theorem 4.1.3] which shows one can take d to be the Littlestone dimension and uses 2^d many reconstruction functions.

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 $i = 1, \dots, d$. By choice of a_i and d_i , this tuple consists of d distinct elements. By construction the set

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

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

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

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 $\kappa(f) = (\bar{a}, a', \bar{d}, a', \dots, a') \in F^d$. From this tuple, one can recover (\bar{a}, \bar{d}) (assuming \bar{a} is nonempty), so we let $\rho_1(\bar{a}, a', \bar{d}, a', \dots, a')$ 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 $\kappa(f) = (\bar{d}, d', \dots, d') \in F^d$, 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 $\rho_0(\bar{d}, d', \dots, d')$ 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 $\kappa(f) = (c, \dots, c)$ for some $c \in F$. Then $\text{Ldim}(\mathcal{C}) = \text{Ldim}(\mathcal{C}_{(c, l)})$ for some $l \in \{0, 1\}$. In particular, if we have defined $\kappa(f') = (c, \dots, c)$ 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 $\rho(c, \dots, c)$ to be a total function extending f_c , which itself extends f . For any tuple output by our algorithm, one of the reconstruction functions produces an extension of the original concept. ◀

References

- 1 Dana Angluin and Tyler Dohrn. The power of random counterexamples. In *International Conference on Algorithmic Learning Theory*, pages 452–465, 2017.
- 2 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.
- 3 Hunter Chase and James Freitag. Model theory and machine learning. *Bulletin of Symbolic Logic*, 25(3):319–332, 2019.
- 4 Sally Floyd and Manfred Warmuth. Sample compression, learnability, and the Vapnik-Chervonenkis dimension. *Machine learning*, 21(3):269–304, 1995.

42:10 Applications of Littlestone Dimension to Query Learning and to Compression

- 5 Vincent Guingona. NIP theories and computational learning theory. URL: <https://tigerweb.towson.edu/vguingona/NIPTCLT.pdf>.
- 6 Hunter R Johnson and Michael C Laskowski. Compression schemes, stable definable families, and o-minimal structures. *Discrete & Computational Geometry*, 43(4):914–926, 2010.
- 7 Akash Kumar, Yuxin Chen, and Adish Singla. Teaching via best-case counterexamples in the learning-with-equivalence-queries paradigm. *Advances in Neural Information Processing Systems*, 34:26897–26910, 2021.
- 8 Dima Kuzmin and Manfred K Warmuth. Unlabeled compression schemes for maximum classes. *Journal of Machine Learning Research*, 8(9), 2007.
- 9 Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1988.
- 10 Nick Littlestone and Manfred Warmuth. Relating data compression and learnability. Technical report, University of California, Santa Cruz, 1986.
- 11 Shay Moran and Amir Yehudayoff. Sample compression schemes for VC classes. *Journal of the ACM (JACM)*, 63(3):21, 2016.
- 12 Manfred K. Warmuth. Compressing to vc dimension many points. In Bernhard Schölkopf and Manfred K. Warmuth, editors, *Learning Theory and Kernel Machines*, pages 743–744, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.