On the Complexity of Approximating the VC Dimension

Elchanan Mossel and Christopher Umans
Microsoft Research, One Microsoft Way, Redmond, WA 98052

E-mail: [mossel,umans]@microsoft.com

We study the complexity of approximating the VC dimension of a collection of sets, when the sets are encoded succinctly by a small circuit. We show that this problem is:

- Σ_3^p-hard to approximate to within a factor $2-\epsilon$ for all $\epsilon > 0$,
- approximable in $\mathcal{AM}$ to within a factor 2, and
- $\mathcal{AM}$-hard to approximate to within a factor $N^{1-\epsilon}$ for all $\epsilon > 0$.

To obtain the Σ_3^p-hardness result we solve a randomness extraction problem using list-decodable binary codes; for the positive result we utilize the Sauer-Shelah(-Perles) Lemma. We prove analogous results for the q-ary VC dimension, where the approximation threshold is q.

1. INTRODUCTION

The VC dimension plays an important role in learning theory, finite automata, comparability theory and computational geometry. It was first defined in statistics by Vapnik and Červonenkis. Let $\mathcal{C}$ be a collection of subsets of a finite set U. The VC dimension of $\mathcal{C}$, denoted $VC(\mathcal{C})$, is the cardinality of the largest subset $F \subset U$ such that every subset of F can be written as the intersection of an element of $\mathcal{C}$ with F.

More generally, let $\mathcal{C}$ be a collection of vectors in $[q]^U$, where U is a finite set, and $q \geq 2$. The q-ary VC dimension of $\mathcal{C}$, denoted $VC_q(\mathcal{C})$, is then defined as the cardinality of the largest subset $F \subset U$, such that the projection of $\mathcal{C}$ to F, defined by $\{(v_x)_{x\in F} : v \in \mathcal{C}\}$, is maximal:

$$\{(v_x)_{x\in F} : v \in \mathcal{C}\} = [q]^F.$$

When $q = 2$, it is easily seen that by identifying sets with their indicator vectors, the VC dimension of a collection of sets is just the 2-ary VC dimension of the associated collection of indicator vectors.

In learning theory and other areas, it is common to prove bounds on the VC dimension of certain set systems or classes of set systems. It is then natural to ask

how hard the function $VC(\mathcal{C})$ is to compute, given a representation of the collection $\mathcal{C}$. Linial, Mansour and Rivest first asked this question in [11]. There the collection $\mathcal{C}$ of sets is given explicitly by a $|\mathcal{C}| \times |U|$ incidence matrix M (that is, $M_{S,x} = 1$ iff $x \in S$). It is noted in [11] that when the input is represented in this way, $VC(\mathcal{C})$ can be computed in time $O(n^{\log n})$. Later, Papadimitriou and Yannakakis [13] gave a more precise characterization of the complexity of this problem by defining a new complexity class LOGNP, and showing that the decision version is LOGNP-complete.

Schaefer [15] observed that in many natural examples, the set system may be exponentially large but have a small implicit representation. That is, there is a polynomial size circuit $C(i, x)$ which outputs 1 iff x is in the i-th set. A good example of such a set system is the collection of monomials over n variables, which is also a standard example from learning theory. In the more general q-ary setting, we consider collections of vectors represented implicitly by a polynomial size circuit $C(i, x)$ which outputs the x-th coordinate of the i-th vector.

Abusing notation slightly, we denote by $VC_q(C)$ the q-ary VC dimension of the collection of vectors encoded by circuit C. The decision version of computing $VC_q(C)$ has been shown to be Σ_3^p-complete ([15], for $q = 2$). An important and natural remaining question is to determine how hard it is to *approximate* $VC_q(C)$. A first step in this direction was taken in [15], in which it was shown that approximating $VC_2(C)$ to within $N^{1-\epsilon}$ is NP-hard for all $\varepsilon > 0$. Still, a large gap remained between this hardness of approximation result, at the first level of the Polynomial Hierarchy, and the complexity of exactly computing the VC dimension, at the third level.

In this paper, we settle the complexity of approximating the q-ary VC dimension, for all q. Specifically we show that computing the VC dimension of a polynomial size circuit C with N inputs is:

- Σ_3^p-hard to approximate to within a factor $q - \Omega(N^{-\varepsilon})$ for all $\varepsilon < 1/4$ if q is a prime power, and $q - \varepsilon$ for all $\varepsilon > 0$ for arbitrary q,
- approximable[1] in $\mathcal{AM}$ to within a factor $q - O(N^{-1/2} \log N)$, and
- $\mathcal{AM}$-hard to approximate to within a factor $N^{1-\epsilon}$ for all $\epsilon > 0$.

In particular, this implies that the problem is Σ_3^p-hard to approximate to within a factor of $q - \epsilon$ and "easy" to approximate to within a factor of q. However, notice that for prime powers q, we are able to locate the threshold of approximability for this problem with unusual accuracy. In statistical physics terminology, we derive non-trivial bounds on the "critical exponent" near the "critical point".

Our result is, to our knowledge, the first to establish a *constant* approximability threshold for an optimization problem above NP in the Polynomial Hierarchy – several Σ_2^p minimization problems are shown to be hard to approximate within N^ϵ factors in [21], and Ko and Lin [10, 9] show that several Π_2^p function approximation problems are hard to approximate to within constant factors, but matching upper bounds are not known. We note that although the latter results ([10, 9]) rely on PCP techniques, [21] proves strong inapproximability results using "only" explicit

[1] In the next section we cast the approximation problem as a promise problem and make precise what we mean by "approximable in AM."

dispersers. In the present paper we are also able to avoid complicated PCP constructions using similar tools; we prove inapproximability by building "zero-error dispersers" for a special class of distributions.

Our $\mathcal{AM}$-hardness result, coupled with the approximability of the q-ary VC dimension within a factor q in $\mathcal{AM}$, shows that the promise problem with gap g, for $N^{1-\epsilon} \geq g \geq q$, is $\mathcal{AM}$-complete. We note that the $\mathcal{AM}$-hardness result for $q = 2$ can be seen as a derandomization of Schaefer's NP-hardness result [15], as $\mathcal{AM}$ is just the class of languages randomly reducible to NP.

The remainder of the paper is organized as follows. We begin with some preliminaries in Section 2. In Section 3, we present a randomness extraction problem and show how to solve it using good list-decodable codes. This construction is the main technical component of our Σ_3^p-hardness result, and may be of independent interest. We proceed in Section 4 with the Σ_3^p hardness result, which builds on Schaefer's reduction. In Section 5 we present the $\mathcal{AM}$ approximation algorithm, whose correctness follows quite easily from the Sauer-Shelah(-Perles) Lemma [14, 17] and its generalizations [8, 1]. Finally, we prove the $\mathcal{AM}$-hardness result in Section 6; here we use deterministic amplification in a critical way to obtain the necessary gap.

2. PRELIMINARIES

We begin with some definitions. We denote by $[q]$ the set $\{0, 1, 2, \ldots, q-1\}$, and for a vector v, we denote by v_x the x-th coordinate of v.

DEFINITION 2.1. Let $\mathcal{C}$ be a collection of vectors in $[q]^U$. The *projection* of $\mathcal{C}$ to $F \subset U$ is defined by

$$\mathcal{C}^F = \{(v_x)_{x \in F} : v \in \mathcal{C}\}.$$

A set $F \subset U$ is *shattered* by $\mathcal{C}$ if $\mathcal{C}^F$ is maximal:

$$\mathcal{C}^F = [q]^{|F|}.$$

The *q-ary VC dimension* of $\mathcal{C}$, denoted $VC_q(\mathcal{C})$, is the size of the largest set $F \subseteq U$ that is shattered by $\mathcal{C}$.

In the following definition, we adopt the natural succinct encoding of vectors by circuits mentioned in the introduction, although our results hold for reasonable variations of this encoding as well.

DEFINITION 2.2. A circuit C computing a function $f : [2^n] \times U \to [q]$ implicitly defines the vectors $v^i = (C(i, x))_{x \in U}$. The *$q$-ary VC dimension of C*, denoted by $VC_q(C)$, is the q-ary VC dimension of the collection $\mathcal{C} = \{v^i\}_{i \in [2^n]}$.

The decision problem we are interested in is the following: Given a circuit C as above and an integer k, is $VC_q(C) \geq k$? It is easy to see that this problem is in Σ_3^p from the following equivalence:

$$\begin{aligned} VC_q(C) \geq k \iff\ & (\exists x_0, \ldots, x_{k-1} \in U) \\ & (\forall s \in [q]^k)(\exists i \in [2^n]) \\ & (\forall j \in \{0, \ldots, k-1\})\ C(i, x_j) = s_j. \end{aligned}$$

An important and easily seen fact is that the q-ary VC dimension of a collection $\mathcal{C}$ is at most $\log_q(|\mathcal{C}|)$. Therefore the final $\forall$ quantifier ranges over a domain of size at most n, so the final line is computable in polynomial time in the input size.

In order to make statements about the complexity of approximating the q-ary VC dimension, we need to define the "gap version" of the decision problem:

DEFINITION 2.3. **q-ary VC dimension with gap g**: Given a circuit C and an integer k, for which either (1) $VC_q(C) \geq k$ or (2) $VC_q(C) < k/g$, which case (1 or 2) holds?

In stating our results, we measure g in terms of the "size" of the instance. For our purposes, the most meaningful size measure is the input length of the circuit, N. In all our results we assume that the circuit C has size bounded by a polynomial in N.

Two of our results relate the complexity of approximating the q-ary VC dimension to the complexity class $\mathcal{AM}$. Recall that a language L is in $\mathcal{AM}$ if and only if there exists a polynomially-balanced, polynomial-time decidable predicate $R_L(x, y, z)$ such that:

$$\begin{aligned} x \in L &\Rightarrow \Pr_y[\exists z\ R_L(x, y, z) = 1] = 1 \\ x \notin L &\Rightarrow \Pr_y[\exists z\ R_L(x, y, z) = 1] \leq 1/2. \end{aligned}$$

Recall that $R_L(x, y, z)$ is polynomially-balanced if $|y| = \text{poly}(|x|)$ and $|z| = \text{poly}(|x|)$. It is straightforward to extend this definition to promise problems $L = (L_{\text{yes}}, L_{\text{no}})$ in the usual way; when we say that the q-ary VC dimension is approximable to within a factor g in $\mathcal{AM}$, we mean formally that the promise problem **q-ary VC dimension with gap g** is in promise-$\mathcal{AM}$. Also, it is sufficient to require that

$$x \notin L \Rightarrow \Pr_y[\exists z\ R_L(x, y, z) = 1] \leq 1 - 1/\text{poly}(|x|)$$

as simple repetition reduces the error to $1/2$.

3. A RANDOMNESS EXTRACTION PROBLEM

The main technical hurdle in the reduction in the next section can be viewed as a randomness extraction problem for a particular type of imperfect random source. Here, we isolate this extraction problem and show that it can be solved in a straightforward way using good efficiently list-decodable codes.

The general extraction problem requires an efficiently computable function $f : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with the property that any input distribution on $\{0,1\}^n$ with "h bits of randomness" (min-entropy at least h) together with the uniform distribution on $\{0,1\}^d$ induces an output distribution that is statistically close to uniform; a function f with this property is called an *extractor*. There is a large body of recent work on extractors (see the survey [12] and the references in [19]).

Earlier work considered the extraction problem for classes of distributions properly contained in the class of distributions with high min-entropy. One example is the class of "bit-fixing sources" introduced by Vazirani [22]. A distribution in this class has $n - h$ (unknown) bit positions fixed to (unknown) values, and the

remaining h bits are chosen uniformly. In this case, many positive results are known [4, 3] and it is even unnecessary to inject truly random bits, as is required in the more general setting. The seemingly minor variation which allows $n-h$ positions to be set to values *that depend on the value of the h random positions* has also been studied. There, it is a consequence of [7] that it is impossible to extract even one almost-random bit deterministically when $n - h > \Omega(n/\log n)$. In this section we consider a generalization of these distributions to q-ary strings, in which h positions vary independently and uniformly over $[q]$, and the remaining $n-h$ positions are set to values that may depend on the h random positions. Notice that the support of such a distribution has q-ary VC dimension h.

The main result in this section is an explicit *zero-error disperser* for sets with large q-ary VC dimension. For comparison, we first define the general disperser which has been considered extensively in the extractor literature as the one-sided variant of an extractor.

DEFINITION 3.1. An (h, ϵ) *disperser* is a function $f : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with the property that for all sets $X \subset \{0,1\}^n$ with $|X| \geq 2^h$,

$$|\{f(x,y) : x \in X, y \in \{0,1\}^d\}| \geq (1-\epsilon)2^m.$$

The parameter ϵ is called the *error*, and in the general case, zero error is not possible. However, we can achieve zero error by imposing additional structure on the source; the *VC-disperser* defined next works on sources with high q-ary VC dimension.

DEFINITION 3.2. An $(h, 0)$ *q-ary VC-disperser* is a function $f : [q]^n \times \{0,1\}^d \to \{0,1\}^m$ with the property that for all sets $X \subset [q]^n$ with $VC_q(X) \geq h$,

$$|\{f(x,y) : x \in X, y \in \{0,1\}^d\}| = 2^m.$$

A disperser f is called *explicit* if f can be computed in polynomial time. We build explicit $(h, 0)$ q-ary VC-dispersers with $h = (1/q + \delta)n$ for all $\delta > 0$, with output length $m = n^{\Omega(1)}$, and with $d = O(\log 1/\delta)$. Our main theorem in this section gives a simple construction of such dispersers from efficiently list-decodable codes (e.g., from [6]).

THEOREM 3.1. *Let ECC be a q-ary error-correcting code with*

- *encoding function $E : [q]^k \to [q]^n$ and*
- *list-decoding function $L : [q]^n \times D \to [q]^k$; i.e., for all $v \in [q]^n$ and $u \in [q]^k$ for which $E(u)$ differs from v in at most e locations, there exists $j \in D$ for which $L(v, j) = u$.*

Let $s : [q]^k \to \{0,1\}^m$ be a surjection for $m \leq \lfloor k \log q \rfloor$. Then the function $f : [q]^n \times D \to \{0,1\}^m$ defined by

$$f(v, j) = s(L(v, j))$$

is a $(n - e, 0)$ q-ary VC-disperser. If s and L are computable in polynomial time, then f is explicit.

Proof. Fix a subset $X \subset [q]^n$ with $VC_q(X) \geq (n-e)$. We need to show that for all $w \in \{0,1\}^m$, there exists a $v \in X$ and a $j \in D$ for which $f(v,j) = w$.

Since s is a surjection, there exists some $u \in [q]^k$ for which $s(u) = w$. Consider the codeword $E(u)$. Since $VC_q(X) \geq n - e$, there exists $F \subset [n]$ of size $n - e$ that is shattered by X. This implies that some $v \in X$ agrees with $E(u)$ in the $n - e$ positions indexed by F; in other words $E(u)$ and v differ in at most e locations. We are therefore guaranteed that for some j, $L(v,j) = u$. We conclude that $f(v,j) = s(L(v,j)) = s(u) = w$, as desired. ■

In the next two lemmas, we describe the current best explicit list-decodable code for our purposes; the first is directly from Guruswami and Sudan [6], and the second follows from [6] with some minor additional work. The main difference between the two is that the first code has a superior dependence on γ, but requires that q be a prime power.

LEMMA 3.1 ([6]). *For every prime power q, integer k, and $\gamma > 0$, there exists an explicitly specified q-ary linear code $E : [q]^k \to [q]^n$ with block length $n = O(\frac{k^2}{\gamma^4})$ for which the following holds for $e \leq n(1 - \frac{1}{q})(1-\gamma)$:*

- *For any received word $v \in [q]^n$ a list of all messages $u \in [q]^k$ for which $E(u)$ differs from v in at most e places can be found in polynomial time.*
- *The list has size at most $O(\gamma^{-2})$.*

LEMMA 3.2. *For all integers q, k, and all* constant *$\gamma > 0$, there exists an explicitly specified q-ary code $E : [q]^k \to [q]^n$ with block length $n = O(\frac{k}{\gamma^8})$ for which the following holds for $e \leq n(1 - \frac{1}{q})(1-\gamma)$:*

- *For any received word $v \in [q]^n$ a list of all messages $u \in [q]^k$ for which $E(u)$ differs from v in at most e places can be found in polynomial time.*
- *The list has size at most $O(\gamma^{-2})$.*

Proof. We modify the proof of Corollary 3 from [6]. Their code is a p^m-ary Reed-Solomon outer code concatenated with any $[O(m/\gamma^2), m, d]_p$ inner code with good distance properties; but it requires p to be a prime power.

We instead pick p to be the smallest prime power greater than q, and $m' = O(m)$ so that $q^{m'} \geq p^m$. We start with an injective map from $[q]^k$ to p^m-ary strings. We then apply the same p^m-ary Reed-Solomon outer code. We also have an injective map from the symbols of the Reed Solomon codewords to q-ary strings of length m'. We then use as our inner code a $(O(m/\gamma^2), m', d)_q$ code with good distance properties. We can construct this code by greedily picking codewords with the required minimum distance, and decode it by brute force. As long as γ is a constant, the construction and decoding take polynomial time in n. ■

Plugging these two codes into Theorem 3.1, we obtain the zero-error VC-dispersers in the following two corollaries.

COROLLARY 3.1. *For every prime power q, integer k, and $1 > \delta > 3/4$, there exists an explicit $(h, 0)$ q-ary VC-disperser*

$$f : [q]^n \times \{0,1\}^{2(1-\delta)\log n + O(1)} \to \{0,1\}^m$$

where $m = \lfloor k \log q \rfloor$, $h = n/q + n^{\delta}$, *and* $n = k^{O(1)}$.

Proof. Using Lemma 3.1 with $\gamma = k^{-\alpha}$ where α is specified later, we obtain a code with block length $n = O(k^{2+4\alpha})$, efficient list-decoding from up to $e = n(1 - \frac{1}{q})(1 - k^{-\alpha})$ errors, and list size $O(k^{2\alpha})$.

Plugging this into Theorem 3.1, we obtain the desired bound on h provided that

$$\delta \leq 1 - \frac{\alpha \log k}{\log n} = 1 - \frac{\alpha \log k}{(2 + 4\alpha) \log k + O(1)}.$$

This expression can be made arbitrarily close to 3/4 by taking α to be a sufficiently large constant. Finally, note that the list size is bounded by $O(n^{2(1-\delta)})$. ■

COROLLARY 3.2. *For all integers* q, k *and all constant* $\delta > 0$, *there exists an explicit* $(h, 0)$ q*-ary VC-disperser*

$$f : [q]^n \times \{0,1\}^{2\log(1/\delta)+O(1)} \to \{0,1\}^m$$

where $m = \lfloor k \log q \rfloor$, $h = n(1/q + \delta)$, *and* $n = k^{O(1)}$.

Proof. We use Lemma 3.2 with $\gamma = \delta$. This yields a code that can tolerate $e = n(1-1/q)(1-\delta)$ errors, and with list size $O(1/\delta^2)$. Plugging this into Theorem 3.1, we get an $(n - e, 0)$ q-ary VC-disperser, which is also the desired $(h, 0)$ q-ary VC-disperser since $h = n(1/q + \delta) > n - e$. ■

Using recent constructions of list-decodable error-correcting codes [5] in Corollary 3.1 we can tolerate any $\delta > 1/2$; however, in this case q must be exponential in n. In closing we remark that the idea of using error-correcting codes "the wrong way" for extracting randomness (from the smaller class of bit-fixing sources) is used in a different way in [4].

4. Σ_3^P-HARDNESS

In this section we use zero-error VC-dispersers to prove Σ_3^p-hardness of approximating the q-ary VC dimension to within a factor $q - \epsilon$. As our reduction builds on Schaefer's reduction, it is instructive to briefly review that reduction (for 2-ary VC dimension) in our terminology.

We begin with an instance of $\mathbf{QSAT}_3$ given by the CNF formula $\phi(a, b, c)$, with $|a| = |b| = |c| = k$. The problem is to determine if $\exists a \, \forall b \, \exists c \, \phi(a, b, c)$. Our instance of **2-ary VC dimension** is a circuit C which encodes a collection $\mathcal{C}$ of binary vectors of length $2^k \cdot k$. We view these vectors as being composed of 2^k blocks of length k each. For every σ, u, and $w \in \{0,1\}^k$, circuit C encodes the vector with u in the σ-th block and zeros elsewhere if $\phi(\sigma, u, w) = 1$, and the all-zero vector otherwise. Therefore, for every $\sigma, u \in \{0,1\}^k$, if $\exists w \ \phi(\sigma, u, w)$, the collection $\mathcal{C}$ includes the vector $0^{\sigma k} \, u \, 0^{(2^k - \sigma - 1)k}$.

Now, if $\forall b \, \exists c \, \phi(a, b, c)$, then it is clear that the set of k indices corresponding to the a-th block is shattered by $\mathcal{C}$.

Conversely, if $VC_2(C) \geq k$, then some set of at least k indices is shattered by $\mathcal{C}$, and we observe that no set with indices in multiple blocks can be shattered by $\mathcal{C}$; therefore, the set that is shattered must be exactly the k indices corresponding to

the a-th block, for some a. This implies that $\mathcal{C}$ includes vectors with all 2^k binary strings in the a-th block, and hence $\forall b\ \exists c\ \phi(a,b,c)$.

To give a similar reduction for **2-ary VC dimension with gap** g, we would like to be able to conclude that $VC_2(C) \geq k/g$ implies $\forall b\ \exists c\ \phi(a,b,c)$ for some a. However, with the current construction, we can only conclude that for *many* $b\ \exists c\ \phi(a,b,c)$. We transform this "many" into "all" by augmenting the construction with the zero-error VC-dispersers from the previous section.

THEOREM 4.1. *For all prime powers q, q-ary VC* **dimension with gap** $q - \Omega(N^{-\varepsilon})$ *is Σ_3^p-hard for all $\varepsilon < 1/4$, and for arbitrary q, q-ary VC* **dimension with gap** $q - \epsilon$ *is Σ_3^p-hard for all constant $\varepsilon > 0$.*

Proof. We give the proof for prime powers q; to get the weaker result for arbitrary q, one should use the zero-error VC-disperser from Corollary 3.2 in place of the one from Corollary 3.1.

Let $\phi(a,b,c)$ be an instance of $\mathbf{QSAT}_3$, with $|a| = |b| = |c| = k$, and take $f : [q]^n \times \{0,1\}^d \to \{0,1\}^k$ to be the $(h,0)$ q-ary VC-disperser from Corollary 3.1, where $\delta > 3/4$, $h \geq n/q + n^\delta$, and $d = 2(1-\delta)\log n + O(1)$. Let $D = 2^d$.

Like above, our instance of **q-ary VC dimension** is a circuit C encoding a collection $\mathcal{C}$ of q-ary vectors of length $2^k \cdot n$. We view each vector as being composed of 2^k blocks of length n; formally, we index our vectors by the set $\{0,1\}^k \times [n]$. For every $\sigma \in \{0,1\}^k$ and every non-zero $u \in [q]^n$, collection $\mathcal{C}$ will include the vector with u in the σ-th block and zeros elsewhere, iff

$$\bigwedge_{j=0}^{D-1} \exists w\ \phi(\sigma, f(u,j), w), \tag{1}$$

plus the all-zeros vector.

It is slightly cumbersome to encode the collection $\mathcal{C}$ in a small circuit C. Recall that circuit C takes two arguments: the "name" of a vector, and an index x, and it outputs the x-th symbol of the named vector. Let $L = \{0,1\}^k$. We name our vectors by tuples from $L \times [q]^n \times L^D$, and recall that each vector is indexed by the set $L \times [n]$. Circuit C is described as follows:

$$C((\sigma, u, w_0, w_1, \ldots, w_{D-1}), (\tau, i)) = \begin{cases} u_i & \text{if } \tau = \sigma \text{ and } \bigwedge_{j=0}^{D-1} \phi(\sigma, f(u,j), w_j) \\ 0 & \text{otherwise} \end{cases}$$

We write v^ℓ for the vector with name $\ell = (\sigma, u, w_0, w_1, \ldots w_{D-1})$ and v^ℓ_τ for the τ-th block of v^ℓ; i.e., $v^\ell_\tau = (v^\ell_{(\tau,i)})_{i \in [n]}$. If v^ℓ is non-zero, then it has the form $v^\ell_\sigma = u$ for some σ, and zeros elsewhere, which implies:

$$\bigwedge_{j=0}^{D-1} \phi(\sigma, f(u,j), w_j) \Rightarrow \bigwedge_{j=0}^{D-1} \exists w\ \phi(\sigma, f(u,j), w)$$

Conversely, if $\forall j\ \exists w_j\ \phi(\sigma, f(u,j), w_j)$, then for $\ell = (\sigma, u, w_1, w_2, \ldots, w_{D-1})$, vector v^ℓ has $v^\ell_\sigma = u$ and zeros elsewhere. Thus C encodes exactly the collection of vectors described above (1).

Without loss of generality we may assume that $k = O(n^{1/2})$ and notice that $D = o(n^{1/2})$ (since $\delta > 3/4$). Circuit C has $N = k + n\lceil \log_2(q) \rceil + Dk + k + \lceil \log n \rceil$ inputs, and we see that $N = O(n)$.

CLAIM 1. ϕ is a positive instance $\Rightarrow VC_q(C) \geq n$.

We know that $(\exists a)(\forall b)(\exists c)\phi(a,b,c)$; fix an a for which $(\forall b)(\exists c)\phi(a,b,c)$. Because ϕ is a positive instance, expression (1) holds for $\sigma = a$ and all $u \in [q]^n$, which implies that $\mathcal{C}$ includes a vector with u in the a-th block, for all u. Therefore, set $F = \{a\} \times [n]$ is shattered, which implies $VC_q(C) \geq n$. ■

CLAIM 2. $VC_q(C) \geq h + 1 = n/q + n^\delta + 1 \Rightarrow \phi$ is a positive instance.

Notice that for every vector in the collection $\mathcal{C}$, the indices (in $L \times [n]$) of the non-zero elements all have the same first coordinate. This implies that the elements of any set shattered by $\mathcal{C}$ also have the same first coordinate.

We know that some set F of size $h+1$ is shattered and that F has the form $\{a\} \times T$, for some $T \subset [n]$ of size exactly $h+1$. Focusing now on only the a-th block of the vectors in $\mathcal{C}$, we define the collection $\mathcal{C}_a = \{v_a | v \in \mathcal{C}\}$. Finally, pick any $t \in T$ and define $\mathcal{C}'_a = \{u \in \mathcal{C}_a | u_t \neq 0\}$. For all $u \in \mathcal{C}'_a$ we must have vectors in $\mathcal{C}$ with u in the a-th block, and all of these vectors are non zero, which implies, by (1):

$$\forall u \in \mathcal{C}'_a \; \forall j \in \{0,1\}^d \; \exists w \; \phi(a, f(u,j), w). \tag{2}$$

Now, the set $T' = T \setminus \{t\}$ is shattered by $\mathcal{C}'_a$, so $VC_q(\mathcal{C}'_a) \geq h$. Since f is a $(h,0)$ q-ary VC-disperser, we know that for every $b \in \{0,1\}^k$ there exists $u \in \mathcal{C}'_a$ and $j \in \{0,1\}^d$ such that $f(u,j) = b$. Therefore (2) implies:

$$\forall b \in \{0,1\}^k \; \exists w \; \phi(a,b,w),$$

which implies that ϕ is a positive instance. ■

We just need to determine the gap g in terms of N. Recalling that $N = O(n)$, we have:

$$g = \frac{n}{n/q + n^\delta + 1} = q(1 - \Omega(n^{\delta-1})) = q(1 - \Omega(N^{\delta-1})),$$

for all $\delta > 3/4$ as desired. ■

We note that improving the bound of Lemma 3.1 on n in terms of γ will result in improving the exponent $1/4$ in Theorem 4.1. We also remark that from a certain perspective, our use of list-decodable binary codes in this reduction is quite similar to an application of such codes to checking NP membership from "noisy" witnesses (see [6]).

5. APPROXIMATION IN AM

In this section we give an Arthur-Merlin protocol for determining the approximate q-ary VC dimension. The main idea is simple: Merlin specifies a set $F \subset U$ that is supposed to be shattered by collection $\mathcal{C}$, and Arthur uses randomness to check that *many* vectors in $[q]^F$ are indeed in the projection $\mathcal{C}^F$ (whereas for F to be shattered, *all* vectors in $[q]^F$ need to be in the projection). We again make the

transition from "many" to "all" by arguing that a slightly smaller set $F' \subseteq F$ is indeed shattered. This step relies on the Sauer-Shelah(-Perles) Lemma [14, 17] and its generalizations to the q-ary case [8] (or see [1] for a simpler proof) which we reformulate below:

LEMMA 5.1. *[8] Let U be an n-element set and $\mathcal{C}$ a collection of vectors in $[q]^U$ such that*

$$|\mathcal{C}| > \sum_{j=0}^{k} \binom{n}{j} (q-1)^{n-j}.$$

Then $\mathcal{C}$ shatters a set $F \subset U$ of size $k+1$.

We can now prove the following theorem:

THEOREM 5.1. *For all q, q-**ary** VC **dimension with gap** $q - O(N^{-1/2} \log N)$ is in $\mathcal{AM}$.*

Proof. We give a constant round Arthur-Merlin protocol for deciding the gap problem with gap $g = q - O(N^{-1/2} \log N)$. It is well-known that any problem decidable by such a constant-round protocol is in $\mathcal{AM}$ (see Babai and Moran [2]). The mutual input is a circuit C encoding a collection $\mathcal{C}$ of 2^n vectors in $[q]^U$, and an integer k, and it is promised that either $VC_q(C) \geq k$ or $VC_q(C) < k/g$. Notice that $VC_q(C) \leq \log_q(2^n) \leq n$, so we can assume $k \leq n$.

Protocol for approximate q-ary VC dimension

- Merlin sends Arthur a set of size k: $F = \{x_0, x_1, \ldots x_{k-1}\} \subset U$.
- Arthur sends Merlin a random vector $v \in [q]^k$.
- Merlin sends Arthur an index $i \in [2^n]$.
- The input is accepted if $C(i, x_j) = v_j$ for $j = 0, 1, \ldots k-1$ and rejected otherwise.

CLAIM 1. If $VC_q(C) \geq k$, then Merlin has a strategy that causes the input to be accepted with probability one.

Proof. Let $F = \{x_0, \ldots x_{k-1}\}$ be a set of size k that is shattered by $\mathcal{C}$. Merlin initially sends F. Since F is shattered, for any vector v that Arthur chooses, there is a response i such that $C(i, x_j) = v_j$ for all j. ■

CLAIM 2. If $VC_q(C) < k/g$, then the input is rejected with probability $n^{-O(1)}$.

Proof. Let F be the set sent by Merlin. We observe that the projection $\mathcal{C}^F$ satisfies $VC_q(\mathcal{C}^F) \leq VC_q(C) < k/g$. Therefore, by Lemma 5.1,

$$\begin{aligned}
|\mathcal{C}^F| &\leq \sum_{j=0}^{k/g} \binom{k}{j} (q-1)^{k-j} = \sum_{j=0}^{\frac{k}{q} + O(kn^{-1/2} \log n)} \binom{k}{j} (q-1)^{k-j} \\
&= q^k \Pr[\mathrm{Bin}(k, 1/q) \leq \frac{k}{q} + O(kn^{-1/2} \log n)] \\
&\leq q^k \Pr[\mathrm{Bin}(k, 1/q) \leq \frac{k}{q} + O(k^{1/2} \log k)] \\
&\leq q^k (1 - k^{-O(1)}) \leq q^k (1 - n^{-O(1)}).
\end{aligned}$$

Therefore the probability (over v) that there exists an i such that $C(i, x_j) = v_j$ for all j, is at most $1 - n^{-O(1)}$. ■

Finally, we note that the input length, N, of circuit C is at least n, which concludes the proof. ■

6. AM-HARDNESS

In this section we prove that approximating the q-ary VC dimension to within $N^{1-\epsilon}$ is $\mathcal{AM}$-hard. The proof uses non-trivial deterministic amplification of $\mathcal{AM}$ in a critical way, after which the reduction and proof follow easily. We illustrate the main idea for the 2-ary case. Let $R_L(x, y, z)$ be the predicate associated with language $L \in \mathcal{AM}$. We have vector y in our collection $\mathcal{C}$ iff $\exists z\ R_L(x, y, z)$. Then, if x is a positive instance, $VC_q(\mathcal{C}) \geq |y|$. However, if x is a negative instance, then $|\mathcal{C}| \leq \delta 2^{|y|}$, where δ is the probability of accepting a negative instance, which implies $VC_q(\mathcal{C}) \leq |y| + \log(\delta)$. By making δ extremely small, we achieve the required gap.

We use dispersers (recall Definition 3.1) for efficient deterministic amplification, a technique first used by Sipser to amplify RP [18]. The parameter d in the definition is called the *seed length*, and to get the strongest results, we need constant-error dispersers with seed length close to the optimal $\log n + O(1)$. Such dispersers were constructed only very recently in [20], with a simpler construction appearing in [16]. We use the following lemma:

LEMMA 6.1 ([16]). *For all h and every $\delta > 0$, there exists an explicit $(h, 1/4)$ disperser*

$$f : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$$

with $d = (1+\delta)\log n + O(\log h)$ and $m = h^{\Omega(\delta)}/\log^{O(1)} n$.

This allows us to prove:

LEMMA 6.2. *For every language L in $\mathcal{AM}$ and every constant $\delta > 0$, there exists a polynomially balanced, polynomial-time decidable predicate $R'_L(x, a, b)$ such that*

$$x \in L \ \Rightarrow\ \Pr_a[\exists b\ R'_L(x, a, b) = 1] = 1$$
$$x \notin L \ \Rightarrow\ \Pr_a[\exists b\ R'_L(x, a, b) = 1] \leq 2^{|a|^\delta}/2^{|a|}.$$

Moreover $|b| = |a|^{1+O(\delta)}$.

Proof. Let $R_L(x, y, z)$ be the predicate from the definition of $\mathcal{AM}$, and let $m = |y|$. Without loss of generality we may assume that $|z| = m$ as well. Let $f : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be an $(h, 1/4)$ disperser from Lemma 6.1, with $n = m^{\Theta(1/\delta^2)}$, $h = n^\delta$ and $d = (1 + O(\delta))\log n$.

Define the new predicate R'_L as follows:

$$R'_L(x; a; b = (z_0, z_1, z_2, \ldots, z_{2^d-1})) = \bigwedge_{j=0}^{2^d-1} R_L(x, f(a, j), z_j),$$

and note that $|b| = 2^d m = n^{1+O(\delta)}$.

If $x \in L$, then it is clear that $\forall a\ \exists b\ R'_L(x,a,b)$. If $x \notin L$, then the set B of random strings y for which $\exists z\ R_L(x,y,z)$ is small, i.e. $|B| \leq 1/2 \cdot 2^m$. We want to bound the number of "bad" random strings a for which $\exists b\ R'_L(x,a,b)$. We notice that string a is bad exactly when $f(a,j) \in B$ for all j. Therefore the set of bad strings a fails to disperse, which implies that there are at most 2^h bad strings a. The error then is $2^h/2^n = 2^{|a|^\delta}/2^{|a|}$ as required. ■

We can now give the reduction:

THEOREM 6.1. *For all q, q-**ary** VC **dimension with gap** $N^{1-\varepsilon}$ is $\mathcal{AM}$-hard for all constant $\varepsilon > 0$.*

Proof. The reduction is a generic reduction. Let L be a language in $\mathcal{AM}$, and let $R'_L(x,a,b)$ be the predicate guaranteed by Lemma 6.2, for some $\delta > 0$ that we specify later. Given an instance x, our instance of **q-ary VC dimension** is a circuit C encoding a collection $\mathcal{C}$ of q-ary vectors of length $n = |a|$. Let $\ell = \lfloor n \log_q 2 \rfloor$ and let $g : [q]^\ell \to \{0,1\}^n$ be any efficiently computable injection. Our collection $\mathcal{C}$ will contain exactly those vectors $v \in [q]^n$ for which $\exists b\ R'_L(x, g(v), b)$, plus the zero vector.

We encode $\mathcal{C}$ by a small circuit C that takes two arguments: the "name" of a vector and an index i, and outputs the i-th coordinate of the named vector. Our vectors will be named by a pair from $[q]^n \times \{0,1\}^{|b|}$, and indexed by the set $[\ell]$. Specifically,

$$C((v,b),i) = \begin{cases} v_i & \text{if } R'_L(x,g(v),b) \\ 0 & \text{otherwise} \end{cases}$$

It is clear that C encodes exactly the vectors in $\mathcal{C}$, as described above.

Now, if $x \in L$, then the set $[\ell]$ is shattered. That is, $\forall a\ \exists b\ R'_L(x,a,b)$, which implies that $\mathcal{C} = [q]^\ell$; i.e., every vector is present.

If $x \notin L$, then the number of distinct vectors in $\mathcal{C}$ is at most the number of a for which $\exists b\ R'_L(x,a,b)$, plus one for the zero vector, which is at most $2^{n^\delta} + 1$. Since $VC_q(\mathcal{C}) \leq \log_q(|\mathcal{C}|)$, we see that in this case the q-ary VC dimension of $\mathcal{C}$ is at most $n^\delta(\log_q 2) + 1$.

We thus have proved a gap of $\Omega(n^{1-\delta})$. Now, the input length of the circuit C is $N = n + |b| + \log \ell \leq n + |b| + \log n$. Recall that Lemma 6.2 guaranteed that $|b| \leq n^{1+O(\delta)}$, so the gap is at least $N^{(1-\delta)/(1+O(\delta))}$. By taking δ sufficiently small, we obtain a gap of $N^{1-\varepsilon}$ for any $\varepsilon > 0$, as desired. ■

ACKNOWLEDGMENT

We thank Venkat Guruswami and Gil Kalai and for helpful discussions and Adam Smith for a useful reference. We thank the anonymous referees for many helpful suggestions.

REFERENCES

1. N. Alon. On the density of sets of vectors. *Discrete Mathematics*, 46:199–202, 1983.
2. L. Babai and S. Moran. Arthur-merlin games: A randomized proof system, and a hierarachy of complexity classes. *Journal of Computer and System Sciences*, 36:254–276, 1988.
3. C. Bennett, G. Brassard, and J. Robert. Privacy amplicifcation by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.

4. B. Chor, J. Friedman, O. Goldreich, J. Hastad, S. Rudich, and R. Smolensky. The bit extraction problem or t-resilient functions. In *Proceedings of the 26th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 396–407, 1985.
5. V. Guruswami and P. Indyk. Expander-based constructions of efficiently decodable codes. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 658–667, 2001.
6. V. Guruswami and M. Sudan. List decoding algorithms for certain concatenated codes. In *Proceedings of the 32nd ACM Symposium on Theory of Computing (STOC)*, 2000.
7. J. Kahn, G. Kalai, and N. Linial. The influence of variables on Boolean functions. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 68–80, 1988.
8. M. G. Karpovsky and V. Milman. Coordinate density of sets of vectors. *Discrete Mathematics*, 24:177–184, 1978.
9. K.-I. Ko and C.-L. Lin. Non-approximability in the polynomial-time hierarchy. Technical Report TR-94-2, Department of Computer Science, State University of New York at Stony Brook, 1994.
10. K.-I. Ko and C.-L. Lin. On the complexity of min-max optimization problems and their approximation. In D.-Z. Du and P. M. Pardalos, editors, *Minimax and Applications*, pages 219–239. Kluwer Academic Publishers, 1995.
11. N. Linial, Y. Mansour, and R. L. Rivest. Results on learnability and the Vapnik-Červonenkis dimension. In *Proceedings of FOCS*, pages 120–129, 1988.
12. N. Nisan and A. Ta-Shma. Extracting randomness: A survey and new constructions. *J. Comput. Syst. Sci.*, 58(1):148–173, Feb. 1999.
13. C. H. Papadimitriou and M. Yannakakis. On limited nondeterminism and the complexity of the V–C dimension. *Journal of Computer and System Sciences*, 53(2):161–70, 1996.
14. N. Sauer. On the density of families of sets. *Journal of combinatorial Theory, A*, 13:145–147, 1972.
15. M. Schaefer. Deciding the Vapnik-Červonenkis dimension is Σ_3^p-complete. *J. Comput. Syst. Sci.*, 58(1):177–182, Feb. 1999.
16. R. Shaltiel and C. Umans. Simple extractors for all min-entropies and a new pseudo-random generator. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 648–657, 2001.
17. S. Shelah. A combinatorial problem: Stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41:247–261, 1972.
18. M. Sipser. Expanders, randomness, or time versus space. *J. Comput. Syst. Sci.*, 36(3):379–383, June 1988.
19. A. Ta-Shma, C. Umans, and D. Zuckerman. Loss-less condensers, unbalanced expanders, and extractors. In *Proceedings of the 33rd ACM Symposium on Theory of Computing (STOC)*, pages 143–152, 2001.
20. A. Ta-Shma, D. Zuckerman, and S. Safra. Extractors from Reed-Muller codes. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 638–647, 2001.
21. C. Umans. Hardness of approximating Σ_2^p minimization problems. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 465–474, 1999.
22. U. Vazirani. Strong communication complexity or generating quasi-random sequences from two communicating semi-random sources. *Combinatorica*, 7(4):375–392, 1987.