On the Complexity of Approximating the VC dimension

Elchanan Mossel
Microsoft Research
One Microsoft Way
Redmond, WA 98052
mossel@microsoft.com

Christopher Umans
Microsoft Research
One Microsoft Way
Redmond, WA 98052
umans@microsoft.com

Abstract

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 any* $\epsilon > 0$*,*
- *approximable in* $\mathcal{AM}$ *to within a factor* 2*, and*
- $\mathcal{AM}$*-hard to approximate to within a factor* N^ϵ *for some constant* $\epsilon > 0$*.*

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. The exact value of* ϵ *in the* $\mathcal{AM}$*-hardness result depends on the degree achievable by explicit disperser constructions.*

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 any subset of F can be written as the intersection of an element of $\mathcal{C}$ with F. We refer the reader to [13] for references and more background.

It is fairly common to compute bounds on the VC dimension of a certain set systems or class of set systems in the context of, say, a learning theory result. It is then natural to ask how hard the function $VC(\mathcal{C})$ is to compute from a representation of the collection $\mathcal{C}$. Linial, Mansour and Rivest first asked this question in [9]. There, $\mathcal{C}$ is given explicitly by an incidence matrix M of size $n = |\mathcal{C}| \times |U|$ such that $M_{S,x} = 1_{\{x \in S\}}$. It is shown in [9] that the $VC(\mathcal{C})$ dimension can be computed in time $O(n^{\log n})$. Later, Papadimitriou and Yannakakis [11] gave a more precise characterization of the complexity of the decision version of this problem by defining a new complexity class LOGNP, and showing the problem to be LOGNP-complete.

Schaefer [13] 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 element x belongs to the set labeled by i. Following Schaefer, we denote by $VC(C)$ the VC dimension of the set system represented by circuit C. The decision version of this variant of the problem is Σ_3^p-complete [13]. An important and natural remaining question is to determine how hard it is to *approximate* $VC(C)$. A first step in this direction was taken in [13], in which it was shown that approximating $VC(C)$ to within $N^{1-\epsilon}$ is NP-hard for all $\varepsilon > 0$.

In this paper, we settle the complexity of approximating the VC dimension. 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 $g \leq 2 - \Omega(N^{-\varepsilon})$ for all $\varepsilon < 1/4$,
- approximable[1] in $\mathcal{AM}$ to within a factor $g \geq 2 - O(N^{-1/2} \log^p N)$ for any constant p, and
- $\mathcal{AM}$-hard to approximate to within a factor $g \leq N^\epsilon$ for some constant $\epsilon > 0$.

In particular, this implies that the problem is Σ_3^p-hard to approximate to within a factor of $2-\epsilon$ and "easy" to approximate to within a factor of 2. However, notice that 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 [18], and Ko and Lin [8, 7] show that several Π_2^p function approximation problems are hard to approximate to within constant factors, but matching upper bounds are not known.)

---

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

Our $\mathcal{AM}$-hardness result, coupled with the approximability of the VC dimension within a factor 2 in $\mathcal{AM}$, shows that the promise problem with gap g, for $N^\epsilon \geq g \geq 2$, is $\mathcal{AM}$-complete. We note that the $\mathcal{AM}$-hardness result can be seen as a derandomization of Schaefer's result [13] that approximation to within a factor $N^{1-\epsilon}$ is NP-hard (as $\mathcal{AM}$ is just the class of languages randomly reducible to NP). If we had an explicit construction of optimal dispersers, we would achieve a factor of $N^{1-\epsilon}$ for all $\epsilon > 0$.

The Σ_3^p hardness result in section 4 builds on Schaefer's reduction. In order to obtain the necessary gap for the inapproximability result, we solve a randomness extraction problem using good list-decodable codes. This construction (in section 3) is the main technical component of our reduction and may be of independent interest. For the $\mathcal{AM}$ hardness result in section 6, we use deterministic amplification in a critical way to obtain the necessary gap. Finally, the proof of the upper bound in section 5 follows quite easily from Sauer-Shelah(-Perles) Lemma [12, 14]. It is amusing to note that a slightly weaker version of the upper bound actually follows from the original Vapnik-Červonenkis paper [19].

2. Preliminaries

We begin with some definitions. For a bit-string s, we use s_i to denote the i-th bit of the string.

Definition 2.1. *Let $\mathcal{C} = \{S_i\}$ be a collection of subsets of a finite set U. A set F is* shattered *by $\mathcal{C}$ if every subset $F' \subseteq F$ can be expressed as $F' = F \cap S_i$ for some i. The* VC dimension *of $\mathcal{C}$ is the size of the largest set $F \subseteq U$ that is shattered by $\mathcal{C}$.*

Definition 2.2. *Given a circuit $C(i,x)$, define the set $S_i = \{x : C(i,x) = 1\}$. The* VC dimension of *C, denoted by $VC(C)$, is the VC dimension of the collection $\mathcal{C} = \{S_i\}_i$.*

The decision problem we are interested in is the following: Given a circuit $C(i,x)$ and an integer k, is $VC(C) \geq k$? It is easy to see that this problem is in Σ_3^p from the following equivalence (here the inputs to $C(i,x)$ are n-bit and m-bit strings, respectively):

$$\begin{aligned} VC(C) \geq k \quad \iff \quad & (\exists x_0, \ldots, x_{k-1} \in \{0,1\}^m) \\ & (\forall s \in \{0,1\}^k)(\exists i \in \{0,1\}^n) \\ & (\forall j \in \{0, \ldots, k-1\}) C(i, x_j) = s_j. \end{aligned}$$

An important fact is that the VC dimension of a class $\mathcal{C}$ is at most $\log_2(|\mathcal{C}|)$. Therefore the final $\forall$ quantifier is over a domain of size at most n, so the final line is computable in polynomial time.

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

VC dimension with gap g

Instance:	Circuit $C(i,x)$ and an integer k
Question:	Determine which of the following cases holds:

YES: $VC(C) \geq k$

NO: $VC(C) < k/g$

In stating the results, we measure g in terms of the "size" of instance. For our purposes, the most meaningful size measure is the number of inputs to the circuit, N; however the circuit always has size polynomial in N.

Two of our results relate the complexity of approximating the 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:

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

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 VC dimension is approximable to within a factor 2 in $\mathcal{AM}$, we mean that the promise problem **VC dimension with gap 2** is in $\mathcal{AM}$. 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 of the protocol 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.

In the most general setting, the 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 "k bits of randomness" (min-entropy at least k), 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*. In the one-sided variant we require only that the output distribution "hits" a $1-\epsilon$ fraction of the range (its support has size at least $(1-\epsilon)2^m$); in this case f is called a *disperser*. The parameter ϵ is referred to as the *error*. There is a large body of recent work on extractors and dispersers (see the survey [10] and the references in [16]).

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 [20]. 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.

A seemingly minor variation allows the $n-h$ "fixed" bit positions to be set to values *that depend on the value of the h random bits*. The source is therefore a uniform distribution on a size 2^h subset for which there exists a h-dimensional subcube such that the projection of the distribution onto this subcube is the full subcube. For lack of a better term, we call such distributions *generalized bit-fixing sources of dimension* h, since they properly include the class of bit-fixing sources. It is a consequence of [6] that it is impossible to extract even one almost-random bit deterministically when $n-h > \Omega(n/\log n)$.

Our application requires a disperser for these distributions with *zero error* ($\epsilon = 0$) that uses at most $O(\log n)$ truly random bits. Using good efficiently list-decodable codes (e.g., from [5]), we can build the desired zero error dispersers for generalized bit-fixing source of dimension $h = n/2+n^\delta$ for $\delta > 3/4$ whose output length is $h^{\Omega(1)}$.

Theorem 3.1. *Let encoding function* $E : \{0,1\}^k \to \{0,1\}^n$ *define a binary error-correcting code, with a polynomial-time list-decoding algorithm* L *that on input* $x \in \{0,1\}^n$ *returns a size* $D = poly(n)$ *subset of* $\{0,1\}^k$ *containing all* y *for which* $E(y)$ *differs from* x *in at most* e *locations. Then* $f : \{0,1\}^n \times \{0,1\}^{\log D} \to \{0,1\}^k$ *defined by* $f(x,z) = z^{th}$ *element of* $L(x)$ *is a zero-error disperser for generalized bit-fixing sources of dimension* $h \geq n-e$.

Proof. The proof is simple. Fix a generalized bit-fixing source $X \subset \{0,1\}^n$. We need to show that for all $y \in \{0,1\}^k$, there exists an $x \in X$ and a $z \in \{0,1\}^{\log D}$ for which $f(x,z) = y$. Consider the codeword $E(y)$. Some $x \in X$ agrees with $E(y)$ in those h bit positions that are not fixed. Since $E(y)$ and x differ in at most e locations, we are guaranteed that x appears in the list output by $L(x)$. Therefore there exists some z for which $f(x,z) = y$. ■

The parameters of the current best explicit list-decodable code for our purposes are given in the following lemma, due to Guruswami and Sudan [5]:

Lemma 3.2 ([5]). *For all* k *and* $\gamma > 0$*, there exists an explicitly specified binary linear code* $E : \{0,1\}^k \to \{0,1\}^n$ *of rate* k*, block length* $n = O(\frac{k^2}{\gamma^4})$ *and such that for all* $e \leq (1-\gamma)n/2$ *the following holds:*

- *For any received word* $x \in \{0,1\}^n$ *a list of all messages* $y \in \{0,1\}^k$ *for which* $E(y)$ *differs from* x *in at most* e *places can be found in polynomial time.*
- *The list has size at most* $O(\gamma^{-2})$.

The list decoding algorithm of Guruswami and Sudan does not return a list of only those codewords within the specified bound, but a potentially larger list. However, since the minimal distance of the code is $(1-\gamma^2)n/2$, if follows by a well known bound (see e.g. [2], Lemma A.1), that the number of codewords which differ from the received word in at most $(1-\gamma)n/2$ places is $O(\gamma^{-2})$. Since the encoding function in Lemma 3.2 is computable in polynomial time, it follows that by pruning we may assume that the length of the list of codewords D is always bounded as $O(\gamma^{-2})$.

We obtain the following corollary:

Corollary 3.3. *For all* k *and* $1 > \delta > 3/4$*, there exists an explicit zero-error disperser*

$$f : \{0,1\}^n \times \{0,1\}^{2(1-\delta)\log n + \Theta(1)} \to \{0,1\}^k$$

for generalized bit-fixing sources of dimension $h \geq n/2 + n^\delta$*, and* $n = poly(k)$.

Proof. Using Lemma 3.2 with $\gamma = k^{-\alpha}$ where α is specified later, we obtain a code with rate k, block length $O(k^{2+4\alpha})$, such it is possible to find in polynomial time the list of all codewords differing from the word in at most $(1-k^{-\alpha})n/2$ places. Moreover the size of this list is $O(k^{2\alpha})$.

Plugging this into Theorem 3.1, we obtain the desired construction for

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

This expression can be made arbitrarily close to $3/4$ by taking α to be a sufficiently large constant. ■

We remark that the idea of using error-correcting codes “the wrong way” for bit extraction (from the smaller class of bit-fixing sources) originated in [4].

4. Σ_3^p-hardness

Theorem 4.1. VC dimension with gap *g is Σ_3^p-hard when $g \leq 2 - \Omega(N^{-\varepsilon})$, for all $\varepsilon < 1/4$.*

Proof. The reduction from $\mathbf{QSAT}_3$ is similar to Schaefer's reduction. Let $\phi(a, b, c)$ be an instance of $\mathbf{QSAT}_3$, with $|a| = |b| = |c| = k$. We let $f : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^k$ be the disperser which was constructed in Corollary 3.3 for generalized bit-fixing sources of dimension $h \geq n/2 + n^\delta$ where $d = 2(1-\delta)\log n + \Theta(1)$ and $D = 2^d$.

We now describe the collection $\mathcal{C}$ of sets comprising our instance of **VC dimension**. Let $L = \{0,1\}^k$ be set of "witnesses." The elements of our sets will be elements of $L \times [n]$, and the sets will be indexed by tuples from $L \times \{0,1\}^n \times L^D$. We stress that the same set may have multiple indices; in particular if ϕ is not satisfiable, then there is only one set – the empty set – and it is indexed by all tuples. We define the set $S_{(\sigma, v, w_0, w_1, \ldots, w_{D-1})}$ to be:

$$\begin{cases} \{(\sigma, i) \in L \times [n] | v_i = 1\} & \text{if } \bigwedge_{j=0}^{D-1} \phi(\sigma, f(v, j), w_j) \\ \emptyset & \text{otherwise} \end{cases} \tag{1}$$

It is easy to see that there is a polynomial-size circuit C that decides if an element $x \in L \times [n]$ belongs to a set specified by an element of $L \times \{0,1\}^n \times L^D$. 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 $N = k + n + Dk + k + \lceil \log n \rceil$ inputs, and we see that $N = \Theta(n)$.

Claim 4.2. *ϕ is a positive instance $\implies VC(C) \geq n$.*

We know that $(\exists a)(\forall b)(\exists c)\phi(a, b, c)$; fix an a for which $(\forall b)(\exists c)\phi(a, b, c)$. We claim that the set $T = \{a\} \times [n]$ is shattered. Pick an arbitrary vector $v \in \{0,1\}^n$ and let $T' = \{(a, i) | v_i = 1\}$. Because ϕ is a positive instance, we know that for each j, there exists some $w_j \in L$ such that $\phi(a, f(v, j), w_j)$ is true. Notice that the set $S_{(a, v, w_0, w_1, \ldots, w_{D-1})}$ is just T'. Therefore T is shattered, which implies that $VC(C) \geq n$.

Claim 4.3. *$VC(C) \geq h + 1 = n/2 + n^\delta + 1 \implies \phi$ is a positive instance.*

Notice that for every set in the collection $\mathcal{C}$ (defined above by (1)), all of the elements have the same first coordinate. This is also true of any set shattered by $\mathcal{C}$.

We know that some set T of size $h + 1$ is shattered. Set T has the form $\{(a, i) | t_i = 1\}$ for some $t \in \{0,1\}^n$ with exactly $h + 1$ ones. Notice that every subset $Q \subseteq T$ can be written as $\{(a, i) | q_i = 1\}$ for some $q \preceq t$ (i.e., for all i, $q_i \leq t_i$). Let i^* be an index such that $t_{i^*} = 1$.

We now argue that $(\forall b)(\exists c)\phi(a, b, c)$. Since T is shattered, each $Q \subseteq T$ as above can be expressed as $Q = R(Q) \cap T$ for some $R(Q) \in \mathcal{C}$. If Q is not empty, then $R(Q)$ must be of the form $R(Q) = S_{(a, r(Q), w_0, w_1, \ldots, w_{D-1})}$ where $r(Q) \in \{0,1\}^n$ and

$$q = r(Q) \wedge t \tag{2}$$

$$\bigwedge_{j=0}^{D-1} \phi(a, f(r(Q), j), w_j) = 1 \tag{3}$$

By (2), the collection $\{r(Q) : Q \subseteq T, r(Q)_{i^*} = 1\}$ is a generalized bit-fixing source of dimension h, where the h positions that are *not* fixed are $\{i : t_i = 1, i \neq i^*\}$. Now, since f is a zero error disperser, if follows that for all b, there exists a Q and $0 \leq j < D$ such that $f(r(Q), j) = b$. Therefore (3) implies $(\forall b)(\exists c)\phi(a, b, c)$.

We therefore achieve a gap of

$$g = \frac{n}{n/2 + n^\delta + 1} = 2(1 - \Omega(n^{\delta-1})) = 2(1 - \Omega(N^{\delta-1})),$$

for all $\delta > 3/4$ as needed. ■

We note that improving the bound of Lemma 3.2 on n in terms of γ will result in improving the exponent $1/4$ in Theorem 4.1. 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 [5]).

5. Approximation in AM

Theorem 5.1. VC dimension with gap *g is in $\mathcal{AM}$ for $g \geq 2 - O(N^{-1/2}\log^p N)$ for all constant p.*

The proof of the theorem relies on the Sauer-Shelah(-Perles) Lemma which we reformulate below:

Lemma 5.2. *[12, 14] Let $\mathcal{C}$ be a collection of subsets of an n element set U such that $|\mathcal{C}| > \sum_{j=0}^{k} \binom{n}{j}$. Then $\mathcal{C}$ shatters a set of size $k + 1$.*

Proof of Theorem 5.1: We give a constant round Arthur-Merlin protocol for deciding the gap problem. It is well-known that any problem decidable by such a constant-round protocol is in AM (see Babai and Moran [1]). The mutual input is a circuit $C(i, x)$ and an integer k, and it is promised that either $VC(C) \geq k$ or $VC(C) < k/g$. Let $n = |i|$, so there are at most 2^n implicitly defined sets, and therefore the VC dimension is no larger than n.

Protocol for approximate VC dimension

- Merlin sends Arthur a set of k elements $X = \{x_0, x_1, \ldots x_{k-1}\}$.
- Arthur sends Merlin a random string $s \in \{0,1\}^k$.
- Merlin sends Arthur an index $i \in \{0,1\}^n$.

- The input is accepted if $C(i, x_j) = s_j$ for $j = 0, 1, \dots k-1$ and is rejected otherwise.

Claim 5.3. *If $VC(C) \geq k$, then Merlin has a strategy that causes the input to be accepted with probability one.*

Proof. Let $X = \{x_0, \dots x_{k-1}\}$ be a set of size k that is shattered. Merlin initially sends X. Since X is shattered, for any string s Arthur chooses, there is a response i such that $C(i, x_j) = s_j$ for all j.

Claim 5.4. *If $VC(C) < k/g$ then the input is rejected with probability $\Omega(n^{-2p})$.*

Proof. Recall that $S_i = \{x : C(i,x) = 1\}$. Consider the collection of sets $\mathcal{C}' = \mathcal{C} \cap X = \{S_i \cap X\}$. Clearly, this collection satisfies $VC(\mathcal{C}') \leq VC(C) < k/g$. Therefore, by Lemma 5.3,

$$\begin{aligned} |\mathcal{C}'| &\leq \sum_{j=0}^{k/g} \binom{k}{j} \\ &= \sum_{j=0}^{k/2+O(n^{-1/2}\log^p n)} \binom{k}{j} \leq (1-\Omega(n^{-2p}))2^k. \end{aligned}$$

Therefore the probability that there exists an i such that $C(i, x_j) = s_j$ for all j is $1 - \Omega(n^{-2p})$.

Noting that $n \leq N$, the theorem follows. ■

6. AM-hardness

Theorem 6.1. VC dimension with gap *g is $\mathcal{AM}$-hard when $g \leq N^{\varepsilon}$ for some $\varepsilon > 0$. If optimal explicit dispersers exist, then the theorem holds for all $\varepsilon < 1$.*

We first need the following lemma, in which we use dispersers for efficient deterministic amplification (a technique first used by Sipser to amplify RP [15]). Recall that a function $f : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a $(k, 1/4)$ *disperser* if for all sets $S \subseteq \{0,1\}^n$ of size at least 2^k, $|f(S, \{0,1\}^d)| \geq (1-1/4)2^m$. Nonconstructively, dispersers exist with $d = \log n + O(1)$; explicit constructions do not yet match that bound.

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

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

Moreover, if explicit dispersers exist with $d = \log n + O(1)$, then $|a| + |b| = |a|^{1+\delta}$.

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. Pick $k = m^{3/2}$, $n = k^{\delta^{-1}}$, and let $f : \{0,1\}^n \times \{0,1\}^{d=O(\log n)} \to \{0,1\}^m$ be a $(k, 1/4)$ disperser (we can use, e.g., the construction in [17]). Define the new predicate R'_L as follows:

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

Note that $|a| + |b| = n + 2^d m$, which is $O(n^{1+\delta})$ if $d = \log n + O(1)$.

If $x \in L$, then it is clear that $\forall a \exists b$ for which $R'_L(x,a,b) = 1$. If $x \notin L$, then the set B of random strings y for which $\exists z R_L(x,y,z) = 1$ is small, i.e. $|B| \leq 1/2 \cdot 2^m$. We want to bound the number of "bad" random strings a for which $\exists b R'_L(x,a,b) = 1$. 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^k bad strings a. The error is then $2^k/2^n = 2^{|a|^\delta}/2^{|a|}$ as required. ■

We proceed with the proof of Theorem 6.1.

Proof of Theorem 6.1: 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, with some $\delta > 0$. Given an instance x, the collection of sets comprising our instance of **VC dimension** is as follows. Each set is labeled by a tuple in $\{0,1\}^{|a|} \times \{0,1\}^{|b|}$.

$$S_{(a,b)} = \begin{cases} \{i | a_i = 1\} & \text{if } R'(x,a,b) \\ \emptyset & \text{otherwise} \end{cases} \tag{4}$$

If $x \in L$, then the set $[|a|]$ is shattered. For every $a \in \{0,1\}^{|a|}$, there exists some b for which $R'_L(x,a,b) = 1$, which implies that $S_{(a,b)} = \{i | a_i = 1\}$. Therefore every subset of $[|a|]$ is present in the collection of sets.

If $x \notin L$, then the number of distinct sets specified by (4) is at most the number of a for which $\exists b R'_L(x,a,b) = 1$, plus one (for the empty set), which is at most $2^{|a|^\delta} + 1$. Since the VC dimension can be no larger than the log of the number of sets, we see that in this case it is at most $|a|^\delta + 1$. We thus have proved a gap of $|a|^{1-\delta}$. Since ${R_L}'$ is polynomially balanced, the number of inputs to the circuit N satisfies $N = |a|^{O(1)}$. We therefore obtain a gap of $N^{(1-\delta)/O(1)}$ as needed. If explicit optimal dispersers exist, we have that for all δ, $N = |a| + |b| + \log|a| = O(|a|^{1+\delta})$, and therefore the gap is $N^{(1-\delta)/(1+\delta)}$, i.e., if δ is sufficiently small, we obtain a gap of $N^{1-\varepsilon}$ for any $\varepsilon > 0$. ■

Acknowledgment. We thank Gil Kalai for helpful discussions and Adam Smith for a useful reference.

References

[1] 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.

[2] M. Bellare, O. Goldreich, and M. Sudan. Free bits, PCP's and non-approximability – towards tight results. *SIAM Journal on Computing*, 27(3):804–915, 1998.

[3] C. Bennett, G. Brassard, and J. Robert. Privacy amplification 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 85)*, pages 396–407, 1985.

[5] V. Guruswami and M. Sudan. List decoding algorithms for certain concatenated codes. In *Proceedings of the 32nd ACM Symposium on Theory of Computing (STOC 00)*, 2000.

[6] 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 88)*, pages 68–80, 1988.

[7] 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.

[8] 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.

[9] N. Linial, Y. Mansour, and R. L. Rivest. Results on learnability and the Vapnik-Červonenkis dimension. In *Proceedings of FOCS*, pages 120–129, 1988.

[10] N. Nisan and A. Ta-Shma. Extracting randomness: A survey and new constructions. *J. Comput. Syst. Sci.*, 58(1):148–173, Feb. 1999.

[11] 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.

[12] N. Sauer. On the density of families of sets. *Journal of combinatorial Theory, A*, 13:145–147, 1972.

[13] M. Schaefer. Deciding the Vapnik-Červonenkis dimension is Σ_3^p-complete. *J. Comput. Syst. Sci.*, 58(1):177–182, Feb. 1999.

[14] S. Shelah. A combinatorial problem: Stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41:247–261, 1972.

[15] M. Sipser. Expanders, randomness, or time versus space. *J. Comput. Syst. Sci.*, 36(3):379–383, June 1988.

[16] A. Ta-Shma, C. Umans, and D. Zuckerman. Unbalanced expanders and improved extractors and dispersers. To appear in Proceedings of the 33rd ACM Symposium on Theory of Computing (STOC 2001), 2001.

[17] L. Trevisan. Construction of extractors using pseudorandom generators. In *Proceedings of the 31st ACM Symposium on Theory of Computing (STOC 99)*, 1999.

[18] C. Umans. Hardness of approximating Σ_2^p minimization problems. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS 99)*, 1999.

[19] V. N. Vapnik and A. Y. Červonenkis. On the uniform convergance of relative frequencies of events to their probability. *Theory of Probability and its Applications*, 16(4):264–280, 1971.

[20] U. Vazirani. Strong communication complexity or generating quasi-random sequences from two communicating semi-random sources. *Combinatorica*, 7(4):375–392, 1987.