Electronic Colloquium on Computational Complexity, Report No. 55 (1998)

Constructions of Near-Optimal Extractors Using Pseudo-Random Generators

[Preliminary Version]

Luca Trevisan*

September 4, 1998

Abstract

We introduce a new approach to construct extractors — combinatorial objects akin to expander graphs that have several applications. Our approach is based on error correcting codes and on the Nisan-Wigderson pseudorandom generator. An application of our approach yields a construction that is simple to describe and analyze, does not utilize any of the standard techniques used in related results, and improves or subsumes almost all the previous constructions.

1 Introduction

Informally defined, an extractor is a function that extracts randomness from a weakly random distribution. Explicit constructions of extractors have several applications and are typically very hard to achieve. In this paper we introduce a new approach to the explicit construction of extractors. Our approach yields a construction that improves most of the known results, and that is optimal for certain parameters. Furthermore, our construction is simple and uses techniques that were never used in this field before — indeed, we do not utilize any of the usual techniques of this field either. The main conceptual contribution of this paper is the use of the Nisan-Wigderson pseudorandom generator in a framework where information-theoretic randomness is being considered.

Extractors and Dispersers. The formal definition of an extractor involves many parameters: an $(n, m, k, d, \varepsilon)$-extractor is a function $Ext : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ that on input a string $u \in \{0,1\}^n$ sampled from a source having *min-entropy* k[1] and a string y uniformly sampled from $\{0,1\}^d$, returns a string $Ext(u, y) \in \{0,1\}^m$ whose distribution (over the choices of u and y) is ε-*close* (see Section 2 for a definition) to the uniform distribution over $\{0,1\}^m$. In section 2 we also define $(n, m, k, d, \varepsilon)$-dispersers ; for the same choice of parameters, an extractor is also a disperser, but not vice-versa. Indeed, constructions of dispersers are somewhat easier than constructions of extractors with the same parameters.

Previous Results and Applications. Dispersers were first defined by Sipser [Sip88], while extractors were first defined by Nisan and Zuckerman [NZ93]. Extractors and dispersers are useful

*`lucat@dimacs.rutgers.edu`. DIMACS Center, Rutgers University, Piscataway, NJ & Columbia University, New York, NY. Work done while at MIT.

[1]A random variable X over $\{0,1\}^n$ has min-entropy k if for every $u \in \{0,1\}^n$ it holds $\mathbf{Pr}[X = u] \leq 2^{-k}$. An illustrative example of a random variable having min-entropy k is the uniform distribution over a set $S \subseteq \{0,1\}^n$ such that $|S| = 2^k$.

ISSN 1433-8092

in several applications where one wants to weaken the randomness requirements of a randomized procedure. One of the main applications of explicit constructions of such objects is the simulation of randomized algorithms using weak random sources. This research area originates from early work by Vazirani and Vazirani [VV85], Santha and Vazirani [SV86], Vazirani [Vaz86, Vaz87], Chor and Goldreich [CG88], and Cohen and Wigderson [CW89] who defined increasingly general models of weak random sources. The recognition of min-entropy as the "right" parameter to measure the amount of algorithmically usable randomness in a source is due to Zuckerman [Zuc90]. Extractors allow to use weak random sources in order to simulate every BPP algorithms; dispersers allow for simulation of RP algorithms. Several constructions of extractors and dispersers [NZ93, SZ94, SSZ98, TS96] were motivated by this application. An optimal result has been achieved in this respect by Saks et al. [SSZ98] for RP algorithms, by finding an explicit construction of $(n, n^{\gamma}, n^{\gamma'}, O(\log n), 1/2)$-dispersers for every $0 < \gamma < \gamma' < 1$. Andreev et al. [ACRT97] showed how to use dispersers in order to simulate BPP algorithms (their result is based on techniques from [ACR98]). The result of [ACRT97], together with the dispersers of [SSZ98] implies an optimal simulation of BPP algorithms. The existence of extractors strong enough to give directly an optimal simulation of any BPP algorithm was still an open question (which we solve in this paper).

Construction of extractors also yield *oblivious samplers*, with applications to randomness-efficient reduction of error in randomized algorithms and in interactive proof-systems, and to leader election in anonymous networks (see [Zuc96b] for a construction whose parameters are optimal for these applications). Construction of extractors (but dispersers would suffice) also yield construction of expander graphs, superconcentrators, and sorting networks. See [WZ93] for results establishing this connection. Constructions of extractors and dispersers yielding tight constructions of expanders, superconcentrators and sorting networks are still not known, though progress was made in [NZ93, SZ94, SSZ98, TS96, TS98]. Other applications have been found more recently in complexity theory: Andreev, Clementi and Rolim [ACR97] use dispersers to prove that certain circuit-complexity assumptions imply P=BPP (without dispersers they would need a stronger assumption). Goldreich and Zuckerman [GZ97] show how to use constructions of extractors to give a simple proof that MA is in $\mathrm{ZPP}^{\mathrm{NP}}$. An open question that may be solved by better construction of dispersers is to prove that Max Clique is not approximable within $n^{1-\varepsilon}$ unless P=NP (the current randomized reduction [FGL+91, Zuc96a, FK94, BS94] from PCP to Max Clique and the PCP construction of Håstad [Hås97] only imply the somewhat weaker consequence that ZPP=NP). It is likely that more applications of extractors will be found in the future. Nisan remarks that extractors "exhibit some of the most 'random-like' properties of explicitly constructed combinatorial structures" [Nis96].

The literature on explicit construction of extractors and dispersers is vast and technically challenging. An excellent and accessible introduction is given by a recent survey by Nisan [Nis96] (see also [NTS98]). In Table 1 we summarize the best known constructions, for different combination of the parameters, and we state the parameters of (a special case of) our construction.

OUR MAIN RESULT. In this paper we introduce a new approach to constructing extractors. An application of this approach yields a construction that works for any min-entropy $k = n^{\Omega(1)}$, extracts a slightly sub-linear fraction of the original randomness (i.e. the length of the output is $m = k^{1-\gamma}$ for an arbitrarily small γ) and uses $O(\log n)$ bits of true randomness. Formally,

Theorem 1 (Main) *For every $m, n, \varepsilon, \gamma$ we can construct a $(n, m, m^{1+\gamma}, d, \varepsilon)$-extractor where $d = O((\log n/\varepsilon)^2 \frac{1}{\gamma} e^{\frac{1}{\gamma}} \frac{1}{\log m})$.*

In particular, for fixed constants $\varepsilon > 0$ and $0 < \gamma < \gamma' < 1$ we have for every n an explicit $(n, n^{\gamma}, n^{\gamma'}, O(\log n), \varepsilon)$-extractor.

Reference	Min entropy k	Output length m	Additional randomness	Type
[Zuc96b]	$k = \Omega(n)$	$m = \Omega(k)$	$O(\log n)$	Extractor
[TS96]	any k	k	poly $\log n$	Extractor
[TS96]	$k = n^{\Omega(1)}$	$m = k^{\Omega(1)}$	$O(\log n \log \cdots \log n)$	Extractor
[SSZ98]	$k = n^{\Omega(1)}$	$m = k^{\Omega(1)}$	$O(\log n)$	Disperser
[TS98]	any k	$m = k^{1-o(1)}$	$O(\log n)$	Disperser
This paper	$k = n^{\Omega(1)}$	$m = k^{\Omega(1)}$	$O(\log n)$	Extractor

Table 1: A summary of previous results and our result.

Our construction improves on the construction of Saks, Srinivasan and Zhou [SSZ98] since we construct an extractor rather than a disperser, and improves over the constructions of Ta-Shma [TS96] since the additional randomness is logarithmic instead of slightly super-logarithmic. The best previous construction of extractors using $O(\log n)$ additional randomness was the one of Zuckerman [Zuc96b], that only works when the min-entropy is a constant fraction of the input length, while in our construction every min-entropy of the form n^γ is admissible. Our construction shows an optimal way of using weak random sources to simulate every randomized procedure. In contrast to the result of [ACRT97] we can use a weak random source to generate almost uniformly distributed random bits independently of the purpose for which the random bits are to be used. This is desirable if, for example, one wants to do probabilistic encryption (or whatever cryptographic application that requires randomness) using a weak random source.

Our construction is not yet the best possible, since we lose part of the randomness of the source and because the additional randomness is logarithmic only as long $k = n^{\Omega(1)}$. We believe that some mix of our approach and of previous techniques will eventually give tight constructions. Our approach is different from previous ones in this field: we do not use any of the standard techniques (hash functions in combination with the leftover hash lemma, composition, etc.), whereas our main tool is the Nisan-Wigderson pseudorandom generator [NW94], which we use for the first time in a framework where information-theoretic randomness is being studied. It is known that a Nisan-Wigderson generator constructed from a *fixed hard function* transforms a small seed of truly random bits into a distribution of longer strings that is *computationally indistinguishable* from the uniform distribution. In this paper we show that applying the Nisan-Wigderson construction to a *random* function sampled from a distribution with certain properties, the outcome of the generator will be *statistically close* to the uniform distribution. The random function can be obtained by encoding with an error correcting code the outcome of a distribution having sufficiently large min-entropy. Note that, in comparison with the standard analysis of the Nisan-Wigderson pseudorandom generator [NW94], we show how to use a stronger assumption (that the function used to construct the generator is random rather than fixed and hard) in order to obtain a stronger consequence (that the output is statistically close, rather than computationally indistinguishable, from the uniform distribution).

Later results. Shortly after the development of the results of this paper, Vadhan [Vad98] showed how to reduce the number of additional random bits that are used in our construction when the length of the output is required to be very close to the min-entropy of the source. In our construction, if the input has min-entropy k and the output is required to be of length m, then the additional randomness is $O(m^{1/\log(k/m)}(\log n)^2/\log(k/m))$. In Vadhan's construction the dependency is $O((\log n)^2/\log(k/m))$. Vadhan then shows how to recursively compose his construction with itself (along the lines of [WZ93]) and he obtains in this way another construction where $k = m$ and the additional randomness is $O(\log^3 n)$. Constructions of extractors with parameters $k \equiv m$ have applications to the explicit construction of expander graphs [WZ93]. In particular, Vadhan [Vad98] presents constructions of expander graphs and of superconcentrators that improve previous ones by Ta-Shma [TS96]. Vadhan's improvements are obtained by reducing the seed length in the Nisan-Wigderson generator that we use. In particular, while the Nisan-Wigderson generator is built upon "combinatorial designs", Vadhan shows that a combinatorial structure of weaker properties (which he calls "weak designs") suffices to make the generator work. This yields an improvement since efficient constructions of weak designs exist whith parameters that are provably better than what would be obtainable using combinatorial designs. Indeed, Vadhan shows a lower bound for design constructions that implies the optimality of the designs used in this paper (and therefore the necessity of using weak designs in order to improve our results). We reference the reader to [Vad98] for further details.

This paper. This paper is in a very preliminary version. Send email to `lucat@dimacs.rutgers.edu` if you are interested in receiving a more polished version as soon as it will be available.

2 Background

In this section we formally define some standard notions that will be used later.

For two random variables X_1 and X_2 over $\{0,1\}^m$, their *statistical distance* is defined as

$$\max_{T:\{0,1\}^m \to \{0,1\}} |\,\mathbf{Pr}[T(X_1) = 1] - \mathbf{Pr}[T(X_2) = 1]| \tag{1}$$

and we say that two distributions are ε-close if their statistical distance is at most ε. The predicate T occuring in Expression (1) will also be called a *statistical test* later. Rephrasing the definition of extractor given in the introduction, we have that an $(n, m, k, d, \varepsilon)$-extractor is a function $Ext : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ such that for every random variable X on $\{0,1\}^n$ with min-entropy k and for every predicate $T : \{0,1\}^m \to \{0,1\}$ it holds

$$\left|\mathbf{Pr}_y[T(Ext(X,y)) = 1] - \mathbf{Pr}_v[T(v) = 1]\right| \leq \varepsilon$$

where y and v are uniformly distributed in $\{0,1\}^d$ and $\{0,1\}^m$, respectively.

An $(n, m, k, d, \varepsilon)$-extractor can also be seen as a bipartite multigraph $G = (U, V, E)$ where $U = \{0,1\}^n$, $V = \{0,1\}^m$, and there exists an edge (u, v) in E for every y such that $Ext(u, y) = v$ (so that the left-degree is 2^d). G has the property (which is implied but not equivalent to the property of Ext being an extractor) that for every subset $S \subseteq U$ with $|S| \geq 2^k$ and every $T \subseteq V$ it holds

$$\left|\frac{cut(S,T)}{2^d|S|} - \frac{|T|}{2^n}\right| \leq \varepsilon$$

where we denote by $cut(S, T)$ the number of edges having one endpoint in S and the other in T. A $(n, m, k, d, \varepsilon)$-disperser is a bipartite multigraph $G = (U, V, E)$ with the same size and degree as before having the property that for every subset $S \subseteq U$ with $|S| \geq 2^k$ and every $T \subseteq V$ with $|T| > \varepsilon 2^m$ there is at least one edge connecting S and T (i.e. $cut(S, T) \geq 1$).

3 Overview of our construction

Nisan and Wigderson introduce in [NW94] a generic construction that given a hard Boolean functions $f : \{0,1\}^l \to \{0,1\}$ yields a pseudorandom generator $NW_f : \{0,1\}^d \to \{0,1\}^m$ that stretches a random seed y of length $d = O(l)$ into a longer string $NW_f(y)$ that is computationally indistinguishable from the uniform distribution. Our first idea would be to define an extractor $Ext(u,y) = NW_u(y)$, where we are identifying the string $u \in \{0,1\}^n$ that the extractor receives in input with a Boolean function $u : \{0,1\}^{\log n} \to \{0,1\}$ in the natural way. This idea is motivated by the fact that the Nisan-Wigderson construction, though formulated in a computational setting, also works in an information-theoretic setting: if $T : \{0,1\}^m \to \{0,1\}$ is a statistical test[2] that distinguishes the output of $NW_f(\cdot)$ from the uniform distribution, then Nisan and Wigderson show how to define a function g that has a short description given T (indeed, they have to show that g is *easily computable* given T, but we need not care about that in our setting) and that *approximates* f, i.e. agrees with f in noticeably more than half of the domain. Our plan is to prove that if u is sampled from a distribution with enough min-entropy, then, for every fixed statistical test T, there will only be a very small probability that a u is sampled such that $NW_u(\cdot)$ fails test T. In order to prove such a claim, we would argue that every string u for which the generator $NW_u(\cdot)$ fails test T has small description, and so there are a few such strings u, and the probability that one of them is sampled is small. This construction, however, is not strong enough to prove our Main Theorem. Informally speaking, the difficulty is that, even though there are not too many boolean functions having small descriptions, there are a lot of functions that are *approximated by* functions having small descriptions, therefore we need to assume a very large min-entropy in order to prove that the construction works for almost all the sampled strings u. One way to improve the construction, and to prove a slightly worse version of the Main Theorem, would be to use a pseudorandom generator construction by Impagliazzo and Wigderson [IW97]. The generator $IW_{\cdot}(\cdot)$ of Impagliazzo and Wigderson has the property that if f is a boolean function and T is a test that distinguishes $IW_f(\cdot)$ from the uniform distribution, then f has a small description given T. Given (an appropriate quantitative version of) the previous property of the Impagliazzo-Wigderson generator, it would be easy to see that an extractor defined as $Ext(u,y) = IW_u(y)$ proves the Main Theorem . However, the previously stated property of the Impagliazzo-Wigderson generator was never observed, let alone proved, before, and even though such a property is "implicitly proved" in [IW97], an explicit proof (whether done by this author, or left to the reader) would be long and complicated. In this paper we rather follow a simpler route and we use *error correcting codes*. Our final way of define $Ext(u,y)$ will be to encode u into a string $\bar{u}$ using an error correcting code, to view $\bar{u}$ as a boolean function $\bar{u} : \{0,1\}^{O(\log n)} \to \{0,1\}$, and then let $Ext(u,y) = NW_{\bar{u}}(y)$. Suppose now that T is a statistical test that is failed by $NW_{\bar{u}}(\cdot)$; then there exists some string u' that is close to $\bar{u}$ and that has a short description, and now, since $\bar{u}$ comes from an error correcting code, the string u' (almost) completely determines $\bar{u}$. In turn, $\bar{u}$ uniquely determines u, and so we can conclude that u itself has a short description. If u is sampled from a distribution with sufficiently large min-entropy, it is now easy to prove that there is a very low probability that a string with a short description be sampled, and so T is almost never failed. We stress that our construction and its analysis are presented in this paper in a completely self-contained way, and that they can be understood without previous knowledge of works on pseudorandom generators.

[2]For a generator $G : \{0,1\}^l \to \{0,1\}^m$ and a statistical test $T : \{0,1\}^m \to \{0,1\}$ we will say that T *distinguishes* $G(\cdot)$ from the uniform distribution (or that G *fails* test T) if $|\mathbf{Pr}_y[T(G(y)) = 1] - \mathbf{Pr}_r[T(r) = 1]| \geq \varepsilon$, where y is uniform in $\{0,1\}^l$, r is uniform in $\{0,1\}^m$, and ε is some fixed constant that depends on context.

4 Main Result

4.1 Preliminaries

In this section we state some known technical results that will be used in the analysis of our extractor. For an integer n we denote by $[n]$ the set $\{1,\ldots,n\}$. We denote by $u_1 \cdot u_2$ the string obtaining by concatenating the strings u_1 and u_2.

Lemma 2 (Error Correcting Codes) *For every n and δ there is an efficient encoding $EC : \{0,1\}^n \to \{0,1\}^{\bar{n}}$ where $\bar{n} = \mathrm{poly}(n, 1/\delta)$ such that every ball of Hamming radius $(1/2-\delta)\bar{n}$ in $\{0,1\}^{\bar{n}}$ contains at most $\mathrm{poly}(1/\delta)$ codewords. Furthermore $\bar{n}$ can be assumed to be a power of 2.*

Stronger parameters are achievable. In particular the length of the encoding can be $\bar{n} = n\mathrm{poly}(1/\delta)$. However, the stronger bounds would not improve our constructions.

Lemma 3 (Design [NW94]) *For every m, $\gamma > 0$ and l there exists an efficiently constructible family of sets $\mathcal{S} = S_1, \ldots, S_m$ such that*

- $S_i \subseteq [d]$, *where* $d = O(l^2 \frac{1}{\gamma} e^{\frac{1}{\gamma}} / \log m)$
- $|S_i| = l$
- $|S_i \cap S_j| \leq \gamma \log m$.

The family $\mathcal{S}$ will be called an (m, l, γ)-design.

Lemma 3 was proved in [NW94] for the special case of $\gamma = 1$. The general case follows using the same proof, but a little care is required while doing a certain probabilistic argument (one has to choose the right Chernoff bound).

The following notation will be useful in the next definition: if $S \subseteq [d]$, with $S = \{s_1, \ldots, s_l\}$ (where $s_1 < s_2 < \cdots < s_l$) and $y \in \{0,1\}^d$, then we denote by $y_{|S} \in \{0,1\}^l$ the string $y_{s_1} \cdot y_{s_2} \cdots y_{s_l}$.

Definition 4 (Nisan-Wigderson Generator [NW94]) *For a function $f : \{0,1\}^l \to \{0,1\}$ and an (m, l, γ)-design $\mathcal{S} = (S_1, \ldots, S_m)$, the Nisan-Wigderson generator $NW_{f,\mathcal{S}} : \{0,1\}^d \to \{0,1\}^m$ is defined as*

$$NW_{f,\mathcal{S}}(y) = f(y_{|S_1}) \cdots f(y_{|S_m})$$

For two functions $f, g : \{0,1\}^l \to \{0,1\}$ and a number $0 \leq \rho \leq 1$ we say that g *approximates* f within a factor ρ if f and g agree on at least a fraction ρ of their domain, i.e. $\mathbf{Pr}_x[f(x) = g(x)] \geq \rho$.

Lemma 5 (Analysis of the NW Generator [NW94]) *Let $\mathcal{S}$ be an (m, l, γ)-design, $f : \{0,1\}^l \to \{0,1\}$ be a Boolean function and $T : \{0,1\}^m \to \{0,1\}$ be such that*

$$\left| \mathbf{Pr}_{y \in \{0,1\}^d}[T(NW_{f,\mathcal{S}}(y)) = 1] - \mathbf{Pr}_{r \in \{0,1\}^m}[T(r) = 1] \right| \geq \varepsilon \ .$$

Then there exists a function $g_{u,T} : \{0,1\}^l \to \{0,1\}^m$ computable by a circuit of size $m^{1+\gamma}$ such that either $T(g_{u,T}(\cdot))$ or its complement approximates $f(\cdot)$ within $1/2 - \varepsilon/m$.

The proof is identical to the proof of Lemma 2.4 in [NW94]. For the ease of the reader we sketch a proof in the appendix.

4.2 Construction

The construction has parameters n, $m \leq n$, γ and ε.

Let $EC : \{0,1\}^n \rightarrow \{0,1\}^{\bar{n}}$ be as in Lemma 2, with $\delta = \varepsilon/m$, so that $\bar{n} = \text{poly}(n, 1/\varepsilon)$, and define $l = \log \bar{n} = O(\log n/\varepsilon)$. For an element $u \in \{0,1\}^n$, view $EC(u)$ as a boolean function $\bar{u}\{0,1\}^l \rightarrow \{0,1\}$.

Let $\mathcal{S} = S_1, \ldots, S_m$ be as in Lemma 3, such that $S_i \subseteq [d]$, $|S_i| = l$, $|S_i \cap S_j| \leq \gamma \log m$, and $d = O(l^2 \frac{1}{\gamma} e^{\frac{1}{\gamma}} \frac{1}{\log m})$.

Then we define $Ext : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^m$ as

$$Ext(u, y) = NW_{\bar{u},\mathcal{S}}(y) = \bar{u}(y_{|S_1}) \cdots \bar{u}(y_{|S_m}) \ .$$

4.3 Analysis

Lemma 6 *For every fixed predicate $T : \{0,1\}^m \rightarrow \{0,1\}$, there are at most $2^{m^{1+\gamma+o(1)}}$ strings $u \in \{0,1\}^n$ such that*

$$\left| \mathop{\mathbf{Pr}}_{y \in \{0,1\}^d}[T(Ext(u,y)) = 1] - \mathop{\mathbf{Pr}}_{r \in \{0,1\}^m}[T(r) = 1] \right| \geq \varepsilon \tag{2}$$

Proof: It follows from the definition of Ext and from Lemma 5 that if u is such that (2) holds, then there exists a function $g_{u,T} : \{0,1\}^l \rightarrow \{0,1\}^m$ computable by a circuit of size $m^{1+\gamma}$ and a bit $b \in \{0,1\}$ such that the function $b \oplus T(g_{u,T}(\cdot))$ approximates $\bar{u}(\cdot)$ within $1/2 - \varepsilon/m = 1/2 - \delta$.

Now, we claim that there are at most $\text{poly}(m/\varepsilon) 2^{m^{1+\gamma+o(1)}}$ strings u for which there exists a function g of circuit complexity $m^{1+\gamma}$ and a bit b such that $\bar{u}(\cdot)$ and $b \oplus T(g(\cdot))$ are within relative distance $1/2 - \delta$. Indeed, there are at most $2^{m^{1+\gamma+o(1)}}$ circuits of size $m^{1+\gamma}$, and therefore there are at most $2^{m^{1+\gamma+o(1)}}$ functions of the form $b \oplus T(g(\cdot))$, where g has circuit complexity $m^{1+\gamma}$. Furthermore, each such function can be within relative distance $1/2 - \varepsilon/m$ from at most $\text{poly}(m/\varepsilon)$ functions $\bar{u}(\cdot)$ coming from the error correcting code of Lemma 2.

We conclude that $\text{poly}(m/\varepsilon) 2^{m^{1+\gamma+o(1)}}$ is an upper bound on the number of strings u for which Expression (2) can occur. □

Theorem 7 *Ext as described above is a $(n, m, m^{1+\gamma+o(1)}, d, \varepsilon + o(1))$-extractor.*

Proof: Fix a predicate $T : \{0,1\}^m \rightarrow \{0,1\}$. From Lemma 6 we have that the probability that sampling a u from a source of min-entropy k we can have

$$|\mathop{\mathbf{Pr}}_{y}[T(Ext(u,y)) = 1] - \mathop{\mathbf{Pr}}_{r}[T(r) = 1]| \geq \varepsilon$$

is at most $2^{m^{1+\gamma+o(1)}} \cdot 2^{-k}$ which can be made $o(1)$ by choosing $k = m^{1+\gamma+o(1)}$. A Markov argument shows that

$$|\mathop{\mathbf{Pr}}_{u,y}[T(Ext(u,y)) = 1] - \mathop{\mathbf{Pr}}_{r}[T(r) = 1]| \leq \varepsilon + o(1)$$

□

5 Final Remarks

The idea of applying results on pseudorandomness to the context of information-theoretic randomness was inspired by previous work of Andreev et al. [ACRT97]. The use of error-correcting codes was inspired by an unpublished new proof of the results of [IW97] due to Madhu Sudan.

Both the error correcting codes of Lemma 2 and the design of Lemma 3 can be constructed in logarithmic space. The construction of designs in logarithmic space requires a logarithmic amount of randomness, and only succeeds with high probability (see [IW97] and also [AR98, Section 5] for details), but both these limitations are not a problem in our construction, since the randomness can be taken from the seed, and a small error probability only contributes to a slight increase of the final statistical difference from the uniform distribution. Therefore, our extractors can be constructed in logarithmic space, unlike the dispersers of [SSZ98, TS98] and the extractors of [TS96].

The analysis of Section 4.3 is not as tight as it could be. Specifically, in order to prove an upper bound on the number of strings u for which $Ext(u, \cdot)$ fails a test T, we first compute an upper bound on the size of circuits "encoding" u, and then we upper bound the number of functions that are computed by circuits of a given size. It would be tighter to state Lemma 5 in a different way, by saying that $g_{u,T}$ belongs to a small family of functions, and giving a bound on the size of the family. Such a bound can be stronger than the bound implied by the circuit size. This improvement is not particularly important in our construction, but becomes essential in the construction of Vadhan [Vad98]. See [Vad98] for this alternative counting argument.

Acknowledgments

I acknowledge the contribution of Oded Goldreich. I thank Danny Lewin, Salil Vadhan, Adam Klivans, Yevgeny Dodis, Venkatesan Guruswami and Amit Sahai for several conversations on [NW94, IW97]. I thank Oded Goldreich, Madhu Sudan and Salil Vadhan for clarifying discussions.

A Appendix

A.1 A Sketch of the Proof of Lemma 3

The following version of the Chernoff bound will be used.

Lemma 8 *Let $X_1, \ldots, X_n$ be 0/1 mutually independent random variables such that $\mathbf{E}[\sum_i X_i] = \mu$. Then, for every $\alpha > 1$ it holds*

$$\mathbf{Pr}[\sum_i X_i \geq \alpha\mu] \leq e^{-((\ln \alpha) + \frac{1}{\alpha} - 1)\alpha\mu}$$

This bound is proved in the standard way, and a proof can be found for example in [LV97]. We can now sketch the proof of Lemma 3 as it was carried on in [NW94].

Proof:[Of Lemma 3] Sequentially choose m subsets of $[d]$ such that any of the chosen subsets intersects the previously chosen ones in less than $\gamma \log m$ points. A probabilistic argument using tha above Chernoff bound shows that the algorithm is always able to choose a new subset as long as the total number of sets in no more than m (in the probabilistic argument we will choose a multi-set of elements, so as to be able to use the Chernoff bound, and then we will discard duplicates.) □

A.2 A Sketch of the Proof of Lemma 5

The following result will be used.

Lemma 9 (Distinguishability versus Predictability [Yao82]) *Let* $T : \{0,1\}^m \rightarrow \{0,1\}$, $g : \{0,1\}^{m-1} \rightarrow \{0,1\}$, $f : \{0,1\}^l \rightarrow \{0,1\}$ *and* $\varepsilon > 0$*; if*

$$|\underset{x\in\{0,1\}^l}{\mathbf{Pr}}[T(g(x), f(x)) = 1] - \underset{x\in\{0,1\}^l, r\in\{0,1\}}{\mathbf{Pr}}[T(g(x), r) = 1]| \geq \varepsilon$$

then there exists two bits $b_0, b_1 \in \{0,1\}$ *such that the function* $b_0 \oplus T(g(x), b_1)$ *agrees with* $f(x)$ *on at least a fraction* $1/2 + \varepsilon$ *of the inputs.*

We now prove Lemma 5.

PROOF: [Of Lemma 5] The main idea is that if T distinguishes $NW_{f,\mathcal{S}}(\cdot)$ from the uniform distribution, then we can find a bit of the output where this distinction is noticeable, and then we will apply Lemma 9. In order to find the "right bit", we will use the so-called *hybrid argument.* We define $m + 1$ distributions $D_0, \ldots, D_m$; D_i is defined as follows as follows: sample a string $v = NW_{f.\mathcal{S}}(y)$ for a random y, and then sample a string $r \in \{0,1\}^m$ according to the uniform distribution, then concatenate the first i bits of v with the last $m - i$ bits of r. By definition, D_0 is distributed as $NW_{f.\mathcal{S}}(y)$ and D_m is the uniform distribution over $\{0,1\}^m$. Using the hypothesis of the Lemma and the triangle inequality we have

$$\begin{aligned}
\varepsilon &\leq |\,\mathbf{Pr}_y[T(NW_{f.\mathcal{S}}(y)) = 1] - \mathbf{Pr}_r[T(r)]| \\
&= |\,\mathbf{Pr}[T(D_0) = 1] - \mathbf{Pr}[T(D_m) = 1]| \\
&= |\sum_{i=0}^{m-1} (\mathbf{Pr}[T(D_i) = 1] - \mathbf{Pr}[T(D_{i+1}) = 1])| \\
&\leq \sum_{i=0}^{m-1} |\,\mathbf{Pr}[T(D_i) = 1] - \mathbf{Pr}[T(D_{i+1}) = 1]|
\end{aligned}$$

In particular, there exists an index i such that

$$|\,\mathbf{Pr}[T(D_i) = 1] - \mathbf{Pr}[T(D_{i+1}) = 1]| \geq \varepsilon/m \tag{3}$$

and there exists a bit $b \in \{0,1\}$ such that

$$\mathbf{Pr}[b \oplus T(D_i) = 1] - \mathbf{Pr}[b \oplus T(D_{i+1}) = 1] \geq \varepsilon/m \tag{4}$$

Now, recall that

$$D_i = f(y_{|S_1}) \cdots f(y_{|S_{i-1}}) r_i r_{i+1} \cdot r_m$$

and

$$D_{i+1} = f(y_{|S_1}) \cdots f(y_{|S_{i-1}}) f(y_{|S_i}) r_{i+1} \cdot r_m$$

and we can use an averaging argument to claim that we can fix $r_1, \ldots, r_m$ to some values $c_i \cdots c_m$, as well as all the all the bits of y except those in S_i, and still have an expression like (4). In particular, we have the relation

$$\mathbf{Pr}[b \oplus T(g_1(x) \cdots g_{i-1}(x) c_i c_{i+1} \cdots c_m) = 1] - \mathbf{Pr}[b \oplus T(g_1(x) \cdots g_{i-1}(x) f(x) c_{i+1} \cdots c_m) = 1] \geq \varepsilon/m$$

where $g_j(x)$ is $f(y_{S_j})$ where y is the string whose bits in S_i are fixed according to x, and whose other bits had been set non-uniformly. Since, by the property of the sets $S_1, \ldots, S_m$, every set S_j contains at most $\gamma \log m$ elements of S_i, it follows that $g_j(x)$ depends on at most $\gamma \log m$ bits of its input and therefore is computable by a circuit of size n^γ. We can now apply Lemma 9 and we have that $b_0 \oplus b \oplus T(g_1(x) \cdots g_{i-1}(x) b_1 c_{i+1} \cdots c_m)$ agrees with f on a fraction $1/2 + \varepsilon/m$ of the inputs. The m-tuple $(g_1(x) \cdots g_{i-1}(x) b_1 c_{i+1} \cdots c_m)$ is computable by a circuit of size at most $m^{1+\gamma}$. □

References

[ACR97] A.E. Andreev, A.E.F. Clementi, and J.D.P. Rolim. Worst-case hardness suffices for derandomization: A new method for hardness vs randomness trade-offs. In *Proceedings of the 24th International Colloquium on Automata, Languages and Programming*, pages 177–187, 1997.

[ACR98] A.E. Andreev, A.E.F. Clementi, and J.D.P. Rolim. A new general derandomization method. *Journal of the ACM*, 45(1):179–213, 1998.

[ACRT97] A.E. Andreev, A.E.F. Clementi, J.D.P. Rolim, and L. Trevisan. Weak random sources, hitting sets, and BPP simulations. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pages 264–272, 1997.

[AR98] E. Allender and K. Reinhardt. Isolation, matching, and counting. Technical Report TR98-019, Electronic Colloquium on Computational Complexity, 1998.

[BS94] M. Bellare and M. Sudan. Improved non-approximability results. In *Proceedings of the 26th ACM Symposium on Theory of Computing*, pages 184–193, 1994.

[CG88] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, April 1988.

[CW89] A. Cohen and A. Wigderson. Dispersers, deterministic amplification, and weak random sources. In *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science*, pages 14–19, 1989.

[FGL+91] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost NP-complete. In *Proceedings of the 32nd IEEE Symposium on Foundations of Computer Science*, pages 2–12, 1991.

[FK94] U. Feige and J. Kilian. Two prover protocols - low error at affordable rates. In *Proceedings of the 26th ACM Symposium on Theory of Computing*, pages 172–183, 1994.

[GZ97] O. Goldreich and D. Zuckerman. Another proof that $BPP \subseteq PH$ (and more). Technical Report TR97-045, Electronic Colloquium on Computational Complexity, 1997.

[Hås97] J. Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. Technical Report TR97-38, Electronic Colloquium on Computational Complexity, 1997. Preliminary version in *Proc. of FOCS'96.*

[IW97] R. Impagliazzo and A. Wigderson. $P = BPP$ unless E has sub-exponential circuits. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, 1997.

[LV97] F.T. Leighton and S. Vempala. Lecture notes for 6.042: Mathematics for computer science, mit, fall'97. Lecture 25. Available at `http://theory.lcs.mit.edu/classes/6.042/Fall97-pub`, 1997.

[Nis96] N. Nisan. Extracting randomness: How and why. In *Proceedings of the 11th IEEE Conference on Computational Complexity*, pages 44–58, 1996.

[NTS98] N. Nisan and A. Ta-Shma. Extrating randomness : A survey and new constructions. *Journal of Computer and System Sciences*, 1998. To appear. Preliminary versions in [Nis96, TS96].

[NW94] N. Nisan and A. Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49:149–167, 1994. Preliminary version in *Proc. of FOCS'88*.

[NZ93] N. Nisan and D. Zuckerman. More deterministic simulation in Logspace. In *Proceedings of the 25th ACM Symposium on Theory of Computing*, pages 235–244, 1993.

[Sip88] M. Sipser. Expanders, randomness or time vs. space. *Journal of Computer and System Sciences*, 36, 1988.

[SSZ98] M. Saks, A. Srinivasan, and S. Zhou. Explicit OR-dispersers with polylogarithmic degree. *Journal of the ACM*, 45(1):123–154, 1998. Preliminary version in *Proc. of STOC'95*.

[SV86] M. Santha and U. Vazirani. Generating quasi-random sequences from slightly random sources. *Journal of Computer and System Sciences*, 33:75–87, 1986.

[SZ94] A. Srinivasan and D. Zuckerman. Computing with very weak random sources. In *Proceedings of the 35th IEEE Symposium on Foundations of Computer Science*, pages 264–275, 1994.

[TS96] A. Ta-Shma. On extracting randomness from weak random sources. In *Proceedings of the 28th ACM Symposium on Theory of Computing*, pages 276–285, 1996.

[TS98] A. Ta-Shma. Almost optimal dispersers. In *Proceedings of the 30th ACM Symposium on Theory of Computing*, 1998.

[Vad98] S. Vadhan. Extracting all the randomness from a weakly random source. Technical Report TR98-048, Electronic Colloquium on Computational Complexity, 1998.

[Vaz86] U. Vazirani. *Randomness, Adversaries and Computation.* PhD thesis, University of California, Berkeley, 1986.

[Vaz87] U. Vazirani. Efficiency considerations in using semi-random sources. In *Proceedings of the 19th ACM Symposium on Theory of Computing*, pages 160–168, 1987.

[VV85] U. Vazirani and V. Vazirani. Random polynomial time is equal to slightly random polynomial time. In *Proceedings of the 26th IEEE Symposium on Foundations of Computer Science*, pages 417–428, 1985.

[WZ93] A. Wigderson and D. Zuckerman. Expanders that beat the eigenvalue bound: Explicit construction and applications. In *Proceedings of the 25th ACM Symposium on Theory of Computing*, pages 245–251, 1993.

[Yao82] A.C. Yao. Theory and applications of trapdoor functions. In *Proceedings of the 23th IEEE Symposium on Foundations of Computer Science*, pages 80–91, 1982.

[Zuc90] D. Zuckerman. General weak random sources. In *Proceedings of the 31st IEEE Symposium on Foundations of Computer Science*, pages 534–543, 1990.

[Zuc96a] D. Zuckerman. On unapproximable versions of *NP*-complete problems. *SIAM Journal on Computing*, 25(6):1293–1304, 1996. Preliminary Version in *Proc. of Structures'93*.

[Zuc96b] D. Zuckerman. Randomness-optimal sampling, extractors and constructive leader election. In *Proceedings of the 28th ACM Symposium on Theory of Computing*, pages 286–295, 1996.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'