can be broken down to four mappings of $n = 4$ to $M = 16$ which only requires 16 binary sequences in each smaller mapping to be mapped while preserving the distance.

ACKNOWLEDGMENT

The authors would like to thank Prof. I. Broere for his helpful comments and verifying of the proofs as well as the anonymous reviewers for comments and criticism that improved this correspondence, in particular Reviewer B for suggesting an alternative proof to Proposition 1.

REFERENCES

[1] A. J. H. Vinck, "Coded modulation for powerline communications," *Proc. Int. J. Elec. Commun.*, vol. 54, no. 1, pp. 45–49, 2000.
[2] I. F. Blake, "Permutation codes for discrete channels," *IEEE Trans. Inf. Theory*, vol. 20, no. 1, pp. 138–140, Jan. 1974.
[3] M. Deza and S. A. Vanstone, "Bounds on permutation arrays," *J. Stat. Planning Inference*, vol. 2, no. 2, pp. 197–209, 1978.
[4] I. F. Blake, G. Cohen, and M. Deza, "Coding with permutations," *Inf. Contr.*, vol. 43, no. 1, pp. 1–19, Oct. 1979.
[5] H. C. Ferreira and A. J. H. Vinck, "Interference cancellation with permutation trellis codes," in *Proc. IEEE Veh. Technol. Conf. Fall 2000*, Boston, MA, Sep. 2000, pp. 2401–2407.
[6] H. C. Ferreira, D. A. Wright, and A. L. Nel, "Hamming distance preserving mappings and trellis codes with constrained binary symbols," *IEEE Trans. Inf. Theory*, vol. 35, no. 5, pp. 1098–1103, Sept. 1989.
[7] C. A. French, "Distance preserving run-length limited codes," *IEEE Trans. Magn.*, vol. 25, no. 5, pp. 4093–4095, Sep. 1989.
[8] T. G. Swart, I. de Beer, and H. C. Ferreira, "Simulation results for permutation trellis codes using M-ary FSK," in *Proc. Int. Symp. on Power Line Commun. and its Applications*, Vancouver, BC, Canada, Apr. 2005, pp. 317–321.
[9] H. C. Ferreira, A. J. H. Vinck, T. G. Swart, and I. de Beer, "Permutation trellis codes," *IEEE Trans. Commun.*, vol. 53, no. 11, pp. 1782–1789, Nov. 2005.
[10] J.-C. Chang, R.-J. Chen, T. Kløve, and S.-C. Tsai, "Distance-preserving mappings from binary vectors to permutations," *IEEE Trans. Inf. Theory*, vol. 49, no. 4, pp. 1054–1059, Apr. 2003.
[11] K. Lee, "New distance-preserving mappings of odd length," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2539–2543, Oct. 2004.
[12] J.-C. Chang, "Distance-increasing mappings from binary vectors to permutations," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 359–363, Jan. 2005.
[13] T. G. Swart, I. de Beer, and H. C. Ferreira, "On the optimality of permutation mappings," in *Proc. Int. Symp. Inf. Theory*, Adelaide, Australia, Sep. 4–9, 2005, pp. 1068–1072.
[14] T. Wadayama and A. J. H. Vinck, "A multilevel construction of permutation codes," *IEICE Trans. Fund.*, vol. E84-A, no. 10, pp. 2518–2522, Oct. 2001.

Improving the Alphabet-Size in Expander-Based Code Constructions

Eran Rom and Amnon Ta-Shma

***Abstract*—Various code constructions use expander graphs to improve the error resilience. Often the use of expanding graphs comes at the expense of the alphabet size. In this correspondence, we show that by replacing the balanced expanding graphs used in the above constructions with unbalanced dispersers the alphabet size can be dramatically improved.**

***Index Terms*—Disperser graphs, expander graphs, extractor codes, list decoding, randomness extractors.**

I. INTRODUCTION

Error-correcting codes were built to deal with the task of correcting errors in transmission over noisy channels. Formally, an $(N, n, d)_q$[1] error correcting code over alphabet Σ, where $|\Sigma| = q$, is a subset $C \subseteq \Sigma^N$ of cardinality q^n in which every two elements are distinct in at least d coordinates. n is called the dimension of the code, N the block length of the code, and d the distance of the code. We call $\frac{d}{N}$ the relative distance of the code. If C is a linear subspace of $[\mathbb{F}_q]^N$, where Σ is associated with some finite field $\mathbb{F}_q$ we say that C is a linear code, and denote it $[N, n, d]_q$ code. From the definition we see that one can uniquely identify a codeword in which at most $\frac{d-1}{2}$ errors occurred during transmission. Moreover, since two codewords from Σ^N can differ in at most N coordinates, the largest number of errors from which unique decoding is possible is $N/2$.

This motivates the list decoding problem, first defined in [4]. In list decoding we give up unique decoding, allowing potentially more than $N/2$ errors, and require that there are only few possible codewords having some modest agreement with any received word. Formally, we say that an $(N, n, d)_q$ code C is (p, K)-list decodable, if for every $r \in \Sigma^N, |\{c \in C \mid \Delta(r, c) \leq pN\}| \leq K$, where $\Delta(x, y)$ is the number of coordinates in which x and y differ. That is, the number of codewords which agree with r on at least $(1 - p)N$ coordinates is at most K. We call the ratio n/N the rate of the code, and p the error rate.

We can demonstrate the difference between unique decoding and list decoding with Reed-Solomon codes. Reed-Solomon codes are linear $[N, n, N - n + 1]_q$ codes, defined for every q such that $\mathbb{F}_q$ is a finite field, and $n \leq N \leq q$. Every $(N, n, d)_q$ code is $(1 - \sqrt{1 - d/N}, qN)$-list decodable [5, Lecture 8]. For Reed–Solomon codes there exists an efficient list decoding algorithm [6]. Thus, unique decoding is possible with at most $N/2$ errors, while by [6] list decoding is possible with up to $N - \sqrt{Nn}$ errors, and the number of all possible decodings is small.

Manuscript received December 17, 2004; revised December 25, 2005. This work was supported by the Israel Science Foundation, by the Binational Science Foundation, and by the EU Integrated Project QAP. The material in this correspondence was presented in part at STACS 2005, Stuttgart, Germany, February 2005.

E. Rom is at 38 Anatot St., Tel-Aviv 69080, Israel (e-mail: eranrom@post.tau.ac.il).

A. Ta-Shma is with the Computer Science Department, Tel-Aviv University, Ramat-Aviv, Tel-Aviv 69978, Israel (e-mail: amnon@post.tau.ac.il).

Communicated by A. Ashikhmin, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2006.878166

[1]We will use n to denote the dimension of a code to avoid confusion with with the min-entropy parameter of extractors and dispersers, for which k is usually reserved.

Often, good code constructions exist for constant relative distance, but are hard to get for large relative distance. For example, Justesen codes are asymptotically good (i.e., have both constant rate and constant relative distance), but the rate dependence on the distance as the relative distance tends to half (and to one over larger alphabet size) is not good. A similar phenomenon exists with list-decoding [3].

In [1] the authors get a better dependence of the rate on the distance, by starting with a constant error Justesen code amplified using an expander-based construction. This approach has been influential in coding theory, and several papers employ this approach for dealing with high noise in both the unique-decoding and the list-decoding setting (e.g., [1]–[3]). A drawback of this approach, is that often the resulting code has a larger than needed alphabet size (for the precise parameters see later).

The amplification above can be done using any *disperser* and, in fact, a good expander is just a special case of a balanced disperser. In this correspondence, we show that by carefully choosing the dispersers used and, in particular, by taking *unbalanced* dispersers, we can dramatically improve the alphabet size without harming the other parameters. We illustrate this on two case studies: the Alon *et al.* construction of an explicit asymptotically good code [1] and the construction of good, high-noise list-decodable codes [3].

A. Two Case Studies

1) High-Noise Unique-Decodable Codes: As we said before, [1] build a code $G \circ C_{\mathrm{Jus}}$ by starting with a constant-error Justesen code C_{Jus} and composing it with an expander G. They show that the composition gives asymptotically good codes with relative distance arbitrarily close to 1, but with large alphabet size. We show that the alphabet size can be much smaller when using an unbalanced disperser G.

Theorem 1: For every $\epsilon > 0$, there exists an explicitly constructible family of codes of rate $\Omega(\epsilon)$, relative distance $(1-\epsilon)$ over alphabet of size:

- $2^{O(\frac{1}{\epsilon})}$, when G is a good expander as in [1];
- $2^{2^{\mathrm{polyloglog}(\frac{1}{\epsilon})}}$, when G is the currently best explicit unbalanced disperser;
- $poly(\frac{1}{\epsilon})$ when using the best nonexplicit disperser.

Similar improvements apply to other results in [1].

2) High-Noise List-Decodable Codes: A simple probabilistic argument shows that $(1-\epsilon, O(\frac{1}{\epsilon}))$-list decodable codes with $rate = \Omega(\epsilon)$, and $|\Sigma| = O(\frac{1}{\epsilon^2})$ exist. Also the rate must be $O(\epsilon)$, and $|\Sigma| = \Omega(\frac{1}{\epsilon})$. Until recently, the best known explicit constructions only achieved rate of ϵ^2. Recently, Guruswami in [3], used an expander based construction to give the first explicit construction of rate $\Omega(\frac{\epsilon}{\log^{O(1)}\frac{1}{\epsilon}})$. However, the alphabet size (and the decoding list size in the fully explicit construction) is huge. The relationship that [3] has found between strong extractors[2] and high-noise list decodable codes, and our improvement are given in the following theorem.

Theorem 2 (Connection Between Strong Extractors and L.D.C.): Let $K = K(N)$ be arbitrary. If a family of $(K, 1/4)$-strong extractors $f : [N] \times [D] \to [M]$ with degree $D = O(\log N)$ and entropy loss $O(1)$ can be explicitly constructed, then one can explicitly construct $(1-\epsilon, O(1/\epsilon))$-list decodable codes of rate $\Omega(\frac{\epsilon}{log(1/\epsilon)})$ over an alphabet size:

- $2^{O(\epsilon^{-1}\log(1/\epsilon))}$ when using balanced expanders [3];
- $2^{2^{\mathrm{polyloglog}(\frac{1}{\epsilon})}}$ when using the currently best explicit disperser;
- $\mathrm{poly}(\frac{1}{\epsilon})$ when using the best nonexplicit disperser.

The above construction assumes the existence of a family of extractors that nonexplicitly exists, but currently we do not know how to construct. Using the currently best explicit extractors, one gets a polynomial time constructible family of $(1-\epsilon, 2^{O(\sqrt{n \log n})})$-list decodable codes, of similar rate and alphabet size. That is, all the parameters (and the improvements) stay the same, and the price of using explicit (nonoptimal) extractor constructions is in the huge number of codewords in the output list.

Another point to make is that Theorem 2 gives (under the assumption made) an explicit code with good list decoding properties, i.e., for any given word w there are only few codewords that are too close to it. Theorem 2, however, does not guarantee *efficient* list decoding. For efficient list decoding one needs to require further properties from the extractor family assumed. Such properties are known for some explicit constructions (e.g., the one in [7]) but not for families with the strong parameters required for the theorem.

B. The Technique

To understand our technical contribution we need to understand the previous work. We first introduce the basic objects (extractors, dispersers, extractor codes), then the error-reduction technique of [1], and the decoding algorithm of [3], and finally our improvement.

1) Introducing the Basic Objects:

- Strong Extractors. An extractor is a function which extracts randomness from a weak random source. A weak random source is a distribution which might be far form uniform but still has some randomness in it. A standard measure for the amount of randomness contained in a source is its min-entropy. A distribution X over $\{0,1\}^n$ has k min-entropy, denoted $H_\infty(X) = k$, if $\forall x, X(x) \leq 2^{-k}$. If $H_\infty(X) = k$ we say that X has k bits of min-entropy. An example of a weak random source, having k min-entropy is a uniform distribution over some subset of 2^k elements from $\{0,1\}^n$.
  A simple fact is that randomness extraction from a weak source cannot be done without additional randomness independent of the source. This leads to the following definition.

Definition 1: $F : [N] \times [D] \to [M]$ is a $(K, \zeta_{\mathrm{ext}})$-strong extractor if for every X distributed over $[N]$ with $H_\infty(X) \geq \log K$, the distribution $y \circ F(x,y)$ is ζ_{ext}-close to $U_{[D]\times[M]}$,[3] where x is drawn from X and y is taken uniformly at random from $[D]$. The entropy loss of the strong extractor is $\frac{K}{M}$. The extractor error is ζ_{ext}. The strong extractor is explicit if $F(x,y)$ can be computed in time polynomial in the input length, i.e., polynomial in $\log N + \log D$.

In other words, the extractor gets an input from some unknown distribution X that is guaranteed to have at least $\log K$ min-entropy and uses some additional $\log D$ truly random bits, called the seed of the extractor, to extract $\log M$ random bits that together with the seed are close to uniform. An extractor (not necessarily strong) is one where we only require that the $\log M$ output bits are close to uniform.

- Dispersers. A disperser is the one-sided variant of an extractor. Instead of requiring that the output is ϵ-close to the uniform distribution, we require that the disperser's output covers at least a $1-\epsilon$ fraction of the target set.

Definition 2: $G : [L] \times [T] \to [D]$ is a $(H, \zeta_{\mathrm{disp}})$-disperser if for every $X \subseteq [L]$ with $|X| \geq H$ we have $|\Gamma_G(X)| \geq (1-\zeta_{\mathrm{disp}})D$, where $\Gamma_G(X) = \{G(\ell, j) \mid \ell \in X, j \in [T]\}$ is the neighbors set of X in G. The entropy loss of the disperser is $\frac{HT}{D}$. The disperser is explicit if $G(x,y)$ can be computed in time polynomial in the input length, i.e., polynomial in $\log L + \log T$.

In Definition 1, we defined a *strong* extractor, while in Definition 2, we defined a (not necessarily strong) disperser. This is due to the way we use these objects later on.

[2]The definition of strong extractors and dispersers is given in Section I-B1

[3]Two distributions are ϵ-close if their statistical distance—defined in Section II—is smaller than ϵ.

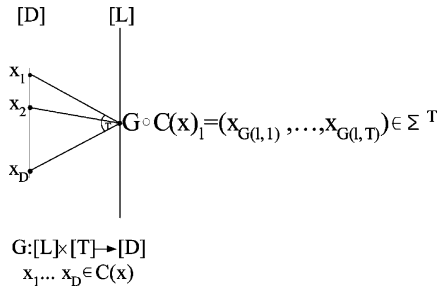

Fig. 1. The encoding: $x_1, \ldots, x_D \in \Sigma$ on the left—the coordinates of some codeword $C(x)$—are "put" along the disperser's output $[D]$, defining for each $\ell \in [L]$ an ordered vector of its neighbors: $(x_{G(\ell,1)}, \ldots, x_{G(\ell,T)}) \in \Sigma^T$, where $x_{G(\ell,t)}$ is the symbol that was matched to $G(\ell,t) \in [D]$. $(x_{G(\ell,1)}, \ldots, x_{G(\ell,T)})$ is the codeword $G \circ C(x)$.

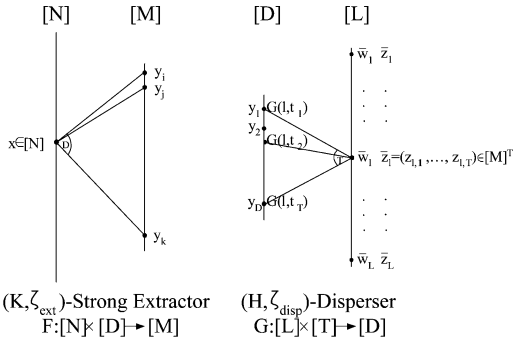

Fig. 2. The Decoding Procedure: Let $z = (\bar{z}_1, \ldots, \bar{z}_L)$ be an arbitrary word in $[M^T]^L$. For each $\ell \in [L], \bar{z}_\ell \in [M]^T$ defines a set $S_\ell \subset [D] \times [M]$ as follows. The pair $(i,s) \in S_\ell$ if $i \in [D]$ is say the tth neighbor of $\ell \in [L]$ and s is the tth symbol in the ordered vector $\bar{z}_\ell$.

- Extractor Codes. [8] observed a simple connection between strong extractors and list decodable codes. Given a strong extractor $F : [N] \times [D] \to [M]$, we define a code $C : [N] \to [M]^D$ as follows: $\forall\, x \in [N], C(x)_i = F(x,i)$. By definition the rate of the code is $\frac{\log N}{D \log M}$. The connection is summarized by the following lemma.

Lemma 1: If $F : [N] \times [D] \to [M]$ is a (K, ζ_{ext})-strong extractor then the extractor code $C(x)$ is $(1 - (\frac{1}{M} + \zeta_{\text{ext}}), K)$-list decodable code.

Also observed by [8] is that extractor codes meet a property stronger than list decoding, known as list recovering [9]. List recovering deals with the situation where the ith symbol of the received word is only known to be in some set $S_i \subseteq \Sigma$. The goal is to find a code $C \subseteq \Sigma^N$ such that for every given $S_1, \ldots S_N \subseteq \Sigma$ describing a received word, there are not to many codewords $C(x)$ with $C(x)_i \in S_i$ for many indices i. List decoding is the case where all sets S_i are of size 1.

2) Error Reduction Using Expanders: As we said before, the technique of code amplification using expanders was introduced in [1]. Here is how the amplification is done: Assume $C : \Sigma^n \to \Sigma^D$ is a $[D, n, \delta D]_q$ code, with rate $r = \frac{n}{D}$. Let $G : [L] \times [T] \to [D]$ be a $(H = \epsilon L, \zeta_{\text{disp}})$-disperser with entropy loss Λ. Define a code $G \circ C : \Sigma^n \to [\Sigma^T]^L$ as follows. For $x \in \Sigma^n$ let $C(x)$ be its encoding using C. Given $C(x) \in \Sigma^D$, we put its symbols along the output vertices of G, such that the i'th symbol of the codeword is matched with the i'th output element of G. We now look at the input elements in $[L]$, each such element has T neighbors each matched with a symbol from Σ. For each input element we collect the symbols of its neighbors and get a symbol in Σ^T. Altogether, we get a code $G \circ C : \Sigma^n \to [\Sigma^T]^L$. The encoding is illustrated in Fig. 1.

A simple argument shows the following.

Lemma 2: If $\zeta_{\text{disp}} < \delta$ then $G \circ C(x)$ is a $[L, \frac{r\epsilon}{\Lambda} L, (1-\epsilon)L]_{q^T}$ code.

The simple proof is given for completeness in Section III. Thus, we increase the relative distance from δ in the original code C to $1 - \epsilon$ in the new code $G \circ C$, at the expense of enlarging the alphabet size from q to q^T. We therefore see that controlling T should be a major goal for us. We also see that the quality of the new rate is largely influenced by the entropy-loss of the disperser.

3) Decoding the New Code: The code constructed above is explicit and Lemma 2 shows it can tolerate high noise. It does not admit, however, explicit decoding. Guruswami and Indyk [2] were the first to propose a decoding mechanism for the code, and for that they replaced the original code C with a list-decodable code. This was further generalized in [3] where C is list-recoverable. We now describe the decoding mechanism.

Imagine we start with some corrupted codeword $z_1, \ldots, z_L$. Each z_i is in Σ^T, and the T symbols (in a correct codeword) are supposed to come from T values of $C(x)$. We can therefore think of z_i as voting for the values of its neighbors. We therefore do the following. For each y_i $(i = 1, \ldots, D)$ we form a set S_i with all the votes about its value. We then use the fact that C is list-recoverable to deduce that there are only few possible code-words having much agreement with $S_1, \ldots, S_D$. The decoding process is formally explained in Section IV-A and is illustrated in Fig. 2.

4) Our Improvement: What we do is replace the expander component in [1] and [3] with a good *unbalanced* disperser. As we discussed above, what is needed in both applications is a disperser for the high min-entropy case that has optimal entropy loss and an extremely small degree. Surprisingly, such objects are possible and nonexplicitly exist. Furthermore, and fortunately, an explicit construction of such a graph was given recently [10] and using such a graph, our improvement over the construction in [3] can be made explicit. For every code built using Guruswami's scheme, the expander component can be replaced with the explicit disperser and improve the alphabet size. As the disperser is explicit, the decoding scheme mentioned in [3] and the time it takes do not change.

II. Preliminaries

A. Statistical Distance

We need the following standard definitions. A probability distribution X over Ω is a function $X : \Omega \to [0,1]$, such that $\sum_{x \in \Omega} X(x) = 1$. U_n is the uniform distribution over $\{0,1\}^n$. The statistical distance between two probability distributions , distributed X, Y over Ω, denoted $|X - Y|$, is $\frac{1}{2} \sum_{x \in \Omega} |X(x) - Y(x)| = \max_{S \subseteq \Omega} |X(S) - Y(S)|$. X, Y are ϵ-close if $|X - Y| \le \epsilon$.

B. Bounds of the Parameters Achievable for Extractors

Ta-Shma and Radhakrishnan [11] show that a (K, ζ_{ext})-strong extractor $F : [N] \times [D] \to [M]$ must have degree $D = \Omega(\frac{1}{\zeta_{\text{ext}}^2} \log \frac{N}{K})$, and entropy loss $\frac{K}{M} = O(\frac{1}{\zeta_{\text{ext}}^2})$. Also shown in [11] are matching implicit upper bounds. The degree lower bound gives the minimal true randomness needed for extracting randomness from a weak source. The entropy loss lower bound gives the amount of randomness lost by the process.

C. Some Dispersers' Parameters

We give below the parameters of the dispersers we use throughout the correspondence.

Alon *et al.* [1] use a balanced expander denoted $G_{\text{balaced}} : [D] \times [T] \to [D]$, which is a $(\epsilon D, \zeta_{\text{disp}})$-disperser with degree

$$T \geq \frac{4\left(\frac{1}{\zeta_{\text{disp}}} - 1\right)}{\epsilon} \tag{1}$$

and with entropy loss

$$\Lambda = \epsilon T \geq 4\left(\frac{1}{\zeta_{\text{disp}}} - 1\right) \tag{2}$$

[11] show the existence of optimal $G_{\text{opt}} : [L] \times [T] \to [D]$ $(\epsilon L, \zeta_{\text{disp}})$-disperser with degree

$$T = O\left(\frac{1}{\zeta_{\text{disp}}} \log \frac{1}{\epsilon}\right) \tag{3}$$

and with entropy loss

$$\Lambda = O\left(\log \frac{1}{\zeta_{\text{disp}}}\right). \tag{4}$$

Finally, Reingold *et al.* [10] give the following explicit extractor (which we use for its expansion properties) based on the zig-zag construction $G_{\text{explicit}} : [L] \times [T] \to [D]$, which is a $(\epsilon L, \zeta_{\text{disp}})$-disperser with degree

$$T = 2^{O(\log^3\left(\frac{1}{\zeta_{\text{disp}}} \log(\frac{1}{\epsilon})\right)} \tag{5}$$

and with entropy loss

$$\Lambda = O\left(\frac{1}{\zeta_{\text{disp}}^2}\right). \tag{6}$$

In the cases we study we demonstrate the improvement in the alphabet size when replacing G_{balanced} with G_{opt} and with G_{explicit}.

D. Reverse Expansion

A basic property of dispersers is that expansion works for both sides, as demonstrated in the following lemma.

Lemma 3: (Reverse expansion) If $G : [L] \times [T] \to [D]$ is a $(\epsilon L, \zeta_{\text{disp}})$-disperser then for any subset $Y \subset [D], |Y| \geq \zeta_{\text{disp}} D$, we have $|\Gamma_G(Y)| \geq (1-\epsilon)L$.

Proof: Any $X \subset [L], |X| \geq \epsilon L$ has $|\Gamma_G(X)| \geq (1-\zeta_{\text{disp}})D$. This implies that for any subset $Y \in [D], |Y| \geq \zeta_{\text{disp}} D$ there can be a set of size at most ϵL in $[L]$ missed by Y. Thus, $|\Gamma_G(Y)| \geq (1-\epsilon)L$. □

E. The Mixing Property

An important property of extractors is mixing [12, Chap 9]. We introduce some notation. For $x \in [N]$ we define $\Gamma_F(x)$ to be the ordered neighbors of x. Formally

$$\Gamma_F(x) = \{(i, F(x,i)) \,|\, i \in [D]\}. \tag{7}$$

The mixing property says that

Fact 1: If $F : [N] \times [D] \to [M]$ is a (K, ζ_{ext})-strong extractor, then for every $S \subseteq [D] \times [M]$, there are at most K elements $x \in [N]$ satisfying

$$\frac{|\Gamma_F(x) \cap S|}{D} - \frac{|S|}{D \cdot M} \geq \zeta_{\text{ext}}. \tag{8}$$

III. Improving the Alphabet Size in High-Noise Unique Decodable Codes

We begin with proving Lemma 1.

Lemma 1: Let $G : [L] \times [T] \to [D]$ be a $(\epsilon L, \zeta_{\text{disp}})$-disperser with entropy loss Λ. Let C be a $[D, rD, \delta D]_q$ code. If $\zeta_{\text{disp}} < \delta$ then $G \circ C$ is a $[L, \frac{r\cdot\epsilon}{\Lambda}L, (1-\epsilon)L]_{q^T}$ code.

Proof: The alphabet size of $G \circ C$ is immediate from the construction. The rate of the construction is also immediate

$$r \cdot \frac{D \log q}{L \log(q^T)} = r \cdot \frac{D}{LT} = \frac{r \cdot \epsilon}{\Lambda}. \tag{9}$$

For the relative distance, let $C(x)$ be a nonzero codeword of C. C is linear with relative distance δ, and so there are at least δD coordinates which are not zero in $C(x)$. By the reverse expansion of dispersers (Lemma 3), these $\delta D > \zeta_{\text{disp}} D$ coordinates have at least $(1-\epsilon)L$ neighbors in G, yielding $(1-\epsilon)L$ coordinates different from zero in $G \circ C(x)$. Thus, the minimal weight of nonzero codeword in $G \circ C$ is at least $(1-\epsilon)L$, and therefore $G \circ C$ has relative distance $(1-\epsilon)$. □

Theorem 1 immediately follows from the above lemma.

Proof: Let $\epsilon > 0$. Take C to be a $[D, r_{\text{Jus}} D, \delta_{\text{Jus}} D]_{q_{\text{Jus}}}$ Justesen code having constant rate, constant relative distance and constant alphabet size. Take $G : [L] \times [T] \to [D]$ to be a $(\epsilon L, \delta_{\text{Jus}})$-disperser. By the above lemma the resulting code $G \circ C$ is a $[L, \frac{r_{\text{Jus}}\cdot\epsilon}{\Lambda}L, (1-\epsilon)L]_{q_{\text{Jus}}^T}$ code. Plugging in the degree and entropy loss of the dispersers G_{balanced} (1), (2), G_{explicit} (5), (6) and G_{opt} (3), (4) gives the claimed rate and alphabet size. □

IV. A Better Connection Between Strong Extractors and L.D.C.

We now turn to our second case study, where we improve on the connection between strong extractors and List decodable codes shown by Guruswami in [3]. Guruswami uses the basic construction of [1], described in Section I-B2, where the code used is an extractor code (mentioned in Section I-B), and the expanding graph is the balanced expander used in [1]. The lemma below was shown by Guruswami for a balanced expander, and we restate it for unbalanced dispersers:

Lemma 4: Let $F : [N] \times [D] \to [M]$ be a strong (K, ζ_{ext})-extractor and $G : [L] \times [T] \to [D]$ be a $(\epsilon L, \zeta_{\text{disp}})$-disperser. If $M \cdot D \geq \frac{L \cdot T}{1-\zeta_{\text{ext}}-\zeta_{\text{disp}}}$, then $G \circ C_F$ is a $(1-\epsilon, K)$-list decodable, where $G \circ C_F$ is the composition of the extractor code C_F with the disperser G as described in Section I-B2.

The proof is practically the same, and we give it below for completeness.

A. The Decoding Procedure

For the proof we first define the decoding procedure for $G \circ C_F$. An eye on Fig. 2 might be helpful. The decoding procedure gets as input an arbitrary word in $[M^T]^L$ and outputs a set $S \subseteq [D] \times [M]$ as follows: Let $z = (\bar{z}_1, \ldots, \bar{z}_L) \in [M^T]^L$. For each $1 \leq \ell \leq L$, the ℓth coordinate $\bar{z}_\ell \in [M]^T$ defines a subset $S_\ell \subseteq [D] \times [M]$ as follows:

$$S_\ell = \{(g(\ell, t), z_{\ell,t}) \,|\, 1 \leq t \leq T\}. \tag{10}$$

To see why we choose these sets, notice that when z is a legitimate codeword of $G \circ C_F$, all pairs in S_ℓ are in the set $\{(i, y_i)\}_{i=1}^D$. When z is an arbitrary word, not necessarily a codeword, two different coordinates $\bar{z}_{\ell_1}$ and $\bar{z}_{\ell_2}$ having a mutual neighbor $i \in [D]$ can "vote" differently about which symbol should reside in the ith coordinate of $[D]$. In other words if i is the jth neighbor of $\bar{z}_{\ell_1}$ and the kth neighbor of $\bar{z}_{\ell_2}$ then it may happen that $\bar{z}_{\ell_{1_j}} \neq \bar{z}_{\ell_{2_k}}$. We define the set S of

$z = (\bar{z}_1, \ldots, \bar{z}_L)$ to be $\bigcup_{\ell=1}^{L} S_\ell$. We now turn to the proof of the above lemma.

Proof: Let $z = (\bar{z}_1, \ldots, \bar{z}_L) \in [M^T]^L$ be an arbitrary word in $[M^T]^L$. Let $S \subseteq [D] \times [M]$ be the output of the above decoding procedure above when given z. Suppose a codeword $G \circ C_F(x) \in [M^T]^L$ agrees with z on a set $\mathcal{H} \subseteq [L]$ of size at least ϵL. Now, if $\ell \in \mathcal{H}$ then $(i, f(x,i)) \in S_\ell$ for every $i \in [D]$ such that i is a neighbor of ℓ in G. Since G is a $(H, \zeta_{\rm disp})$-disperser, the set of neighbors of $\mathcal{H}$ has at least $(1-\zeta_{\rm disp})D$ elements. Hence, $|\Gamma_F(x) \cap S| \geq (1-\zeta_{\rm disp})D$. Noting that $|S| \leq L \cdot T$, and using the assumption $M \cdot D \geq \frac{L \cdot T}{1-\zeta_{\rm ext}-\zeta_{\rm disp}}$, we see that $\frac{|S|}{MD} \leq 1 - \zeta_{\rm ext} - \zeta_{\rm disp}$ and together

$$\frac{|\Gamma_F(x) \cap S|}{D} - \frac{|S|}{MD} \geq \zeta_{\rm ext}. \tag{11}$$

By Fact 1 we conclude that there are at most Ks's for which $G \circ C_F(x)$ agrees with z on at least H coordinates. Hence, the code is $(1-\epsilon, K)$-list decodable. □

B. What Makes the Difference

First, let us have a second look at Guruswami's construction. A strong extractor gives a list decodable code that can correct $1-\alpha$ noise with α^2 penalty in the rate, and so we do not lose much when α is a constant. Indeed, on the left of Fig. 2 we use a strong extractor for a constant error rate.

We are then left with the task of amplifying the error. For that Guruswami uses a balanced expander. The property that we need from the expander, is that every set (of relatively small cardinality ϵL) on the right hand side (of Fig. 2) sees almost all of the vertices on the left hand side as its neighbors (more precisely $1-\zeta_{\rm disp}$ of them).

Taking a balanced expander does the job, but at the cost of enlarging the disperser degree T. This is because ϵL vertices can have at most ϵLT neighbors, and so if ϵLT is almost D and the graph is balanced $L = D$, it must be that the degree T is an order of $(\frac{1}{\epsilon})$. This makes the alphabet size exponential in $\frac{1}{\epsilon}$. On the other hand, if we take a larger right hand side L (such that ϵL is roughly D) we can use a much smaller degree T (an order of $\log(\frac{1}{\epsilon})$) and still have the same property.

One can worry what happens to the rate when taking an unbalanced disperser. However, as we saw before the new rate is $\frac{r \cdot \epsilon}{\Lambda_G}$, where r in this case is the rate of the extractor code. Thus, by taking a disperser with optimal entropy loss, we don't lose on the rate, while dramatically improving the alphabet size.

For this to work we need a good disperser that works for the high min-entropy setting and tiny degree. Luckily, the recent Zig-Zag construction [10] explicitly constructs such a graph. We mention that ϵL min-entropy is considered high, as it has $\log(L) - \log(\frac{1}{\epsilon})$ min-entropy, and the entropy deficiency is only $\log(\frac{1}{\epsilon})$.

C. Analyzing the Parameters

We now find out the parameters of the extractor and disperser to be used in the construction, so as to get a $(1-\epsilon, O(\frac{1}{\epsilon}))$-list decodable code. These parameters must not violate the lower bounds of the extractor and disperser, and the condition of Lemma 4. Since the lower bounds match nonexplicit upper bounds, the parameters we find give a nonexplicit construction for the desired code.

1) The Constraints: First, we write down all the constraints. The bounds we give are both lower bounds, and achievable by nonexplicit constructions. We have

$$D = \Omega\left(\frac{1}{\zeta_{\rm ext}^2} \cdot \log \frac{N}{K}\right). \tag{12}$$

$$M = O\left(K \zeta_{\rm ext}^2\right). \tag{13}$$

$$T = \Omega\left(\frac{1}{\zeta_{\rm disp}} \cdot \log(\frac{1}{\epsilon})\right). \tag{14}$$

$$D = O\left(\frac{\epsilon LT}{\log \frac{1}{\zeta_{\rm disp}}}\right). \tag{15}$$

$$M \cdot D \geq \frac{L \cdot T}{1 - \zeta_{\rm ext} - \zeta_{\rm disp}}. \tag{16}$$

The first two equations are the degree and entropy loss of the extractor, the third and fourth are the degree and entropy loss of the disperser, and the fifth is the construction bound that guarantees that the set S is small in $[D] \times [M]$.

2) A Specific Choice of Parameters: We now choose parameters. We first set $\zeta_{\rm ext}, \zeta_{\rm disp}$ to be small constants, say we set both to be $\frac{1}{4}$. In order to get a $(1-\epsilon, O(\frac{1}{\epsilon}))$-list decodable code we set $K = \Theta(\frac{1}{\epsilon})$. With these choices we have $D = \Theta(\log(N)), M = \Theta(K) = \Theta(\frac{1}{\epsilon})$, and $T = \Theta(\log \frac{1}{\epsilon})$. To satisfy (15) we need to take L such that $\epsilon L = \Theta(\frac{D}{T}) = \Theta(\frac{\log(N)}{\log \frac{1}{\epsilon}})$ which implies that $L = \Theta(\frac{\log(N)}{\epsilon \cdot \log(\frac{1}{\epsilon})})$. Finally, we check (16). We see that $M \cdot D = \Theta(\frac{\log(N)}{\epsilon})$ and $L \cdot T = \Theta(\frac{\log(N)}{\epsilon})$, so with the proper choice of constants the equation holds. We let $N = 2^n$ and $\epsilon > 0$ be our basic parameters. We summarize all other parameters as functions in n and ϵ. We have

$$K = \Theta\left(\frac{1}{\epsilon}\right). \tag{17}$$

$$D = \Theta(n). \tag{18}$$

$$M = \Theta\left(\frac{1}{\epsilon}\right). \tag{19}$$

$$L = \Theta\left(\frac{n}{\epsilon \cdot \log(\frac{1}{\epsilon})}\right). \tag{20}$$

$$H = \Theta\left(\frac{n}{\log(\frac{1}{\epsilon})}\right). \tag{21}$$

$$T = \Theta\left(\log \frac{1}{\epsilon}\right). \tag{22}$$

Thus, rate $= \frac{\log N}{L \cdot T \log M}$ is $\Theta(\frac{\epsilon}{\log(\frac{1}{\epsilon})})$, and the alphabet size $|\Sigma| = M^T$ is $2^{O(\log^2(\frac{1}{\epsilon}))}$. This proves that using the best implicit disperses one gets the parameters stated in Theorem 2 for the best possible disperser.

V. Explicit Constructions

We now make the construction explicit by plugging in explicit disperser and explicit strong extractor. Naturally, the parameters deteriorate. As before, we set the extractor and disperser errors to be constants, say $\zeta_{\rm ext} = \zeta_{\rm disp} = \frac{1}{4}$. We note that (16) now becomes, $L \cdot T \leq \frac{1}{2} \cdot M \cdot D$ For the explicit disperser we use the Zig-Zag construction, mentioned in Section II-C. By (5), (6), having constant $\zeta_{\rm disp}$ the disperser $G_{\rm explicit} : [L] \times [T] \to [D]$ has degree

$$T = 2^{O(\text{polyloglog}(\frac{1}{\epsilon}))} \tag{23}$$

and entropy loss

$$\Lambda = O(1). \tag{24}$$

For the explicit extractor $F : [N] \times [D] \to [M]$ we use the best explicit construction to date of a strong extractor with almost linear seed length due to [13].

Fact 2 [[13]]: For Every $m = m(n), k = k(n)$, and $\zeta_{\rm ext} = \zeta_{\rm ext}(n)$ such that $3m\sqrt{n \log(n/\zeta_{\rm ext})} \leq k \leq n$, there is an explicit family of $(k, \zeta_{\rm ext})$-strong extractors $E_n : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = \log n + O(\log \frac{m}{\zeta_{\rm ext}})$.

Denoting $N = 2^n, K = 2^k, D = 2^d$, and $M = 2^m$, and taking a $\zeta_{\text{ext}} = \frac{1}{4}$, the above fact states that for every $M^{O(\sqrt{\log N \log\log N})} \leq K \leq N$, there exists an explicit $(K, \frac{1}{4})$-strong extractor with degree

$$D = \log N \log^{O(1)} M. \tag{25}$$

Again, we take $M = \Theta(\frac{1}{\epsilon})$, meaning $K = 2^{O(\log(\frac{1}{\epsilon})\sqrt{\log N \log\log N})}$, and $D = \log N \log^{O(1)}(\frac{1}{\epsilon})$. It can be easily verified that by taking an appropriate constant in the $\Theta(\cdot)$ notation in the choice of M, the construction constraint: $L \cdot T \leq \frac{1}{2} \cdot M \cdot D$ is satisfied. Plugging the above disperser and extractor in the construction we get alphabet size

$$M^T = 2^{2^{\text{polyloglog}(\frac{1}{\epsilon})}} \tag{26}$$

and rate

$$\frac{\log N}{D \log M} \frac{\epsilon}{\Lambda} = \frac{\epsilon}{\log^{O(1)}(\frac{1}{\epsilon})}. \tag{27}$$

VI. On the Optimality of the Parameters Choice

We now take a closer look at the parameters. Specifically, we show that the parameters chosen in Section IV-C2, which give good but sub optimal rate and alphabet size w.r.t. the non explicit construction, are the best possible in the above construction:

Lemma 5: In the construction given in Section IV, for any choice of parameters satisfying error rate of $1 - \epsilon$ and strictly positive rate, it must be that $\Sigma = \Omega(2^{\log^2(\frac{1}{\epsilon})})$, and $r = O(\frac{\epsilon}{\log(\frac{1}{\epsilon})})$.

Proof: Having an error rate of $1 - \epsilon$, the min-entropy of the disperser must be ϵL. Thus, by (14) $T = \Omega(\log(\frac{1}{\epsilon}))$. The construction constraint $M > \frac{LT}{D} = \frac{\Lambda_G}{\epsilon}$ implies that $M = \Omega(\frac{1}{\epsilon})$. Finally, having rate strictly bigger that zero, implies $D = O(\log N)$. Altogether, we get

$$M^T = \Omega(2^{\log^2(\frac{1}{\epsilon})}) \tag{28}$$

and

$$r = \frac{\log N}{D \log M} \frac{\epsilon}{\Lambda_G} = O\left(\frac{\epsilon}{\log\left(\frac{1}{\epsilon}\right)}\right). \tag{29}$$

□

References

[1] N. Alon, J. Bruck, J. Naor, M. Naor, and R. Roth, "Construction of asymptotically good, low-rate error-correcting codes through pseudo-random graphs," *IEEE Trans. Inf. Theory*, vol. 38, pp. 509–516, 1992.

[2] V. Guruswami and P. Indyk, "Expander-based constructions of efficiently decodable codes," in *Proc. 42nd Annu. IEEE Symp. Found. Comput. Sci.*, 2001, pp. 658–667 [Online]. Available: citeseer.ist.psu.edu/guruswami01expanderbased.html

[3] V. Guruswami, "Better extractors for better codes?," in *Proc. 36th Annu. ACM Symp. Theory Comput.*, 2004, pp. 436–444.

[4] P. Elias, "List decoding for noisy channels," in *1957-IRE WESCON Conv. Rec., Pt. 2*, 1957, pp. 94–104.

[5] M. Sudan, *Lecture Notes on Algorithmic Introduction to Coding Theory*. : [Online]. Available: http://theory.lcs.mit.edu/~madhu/FT01/scribe/overall.ps

[6] V. Guruswami and M. Sudan, "Improved decoding of reed-solomon and algebraic-geometric codes," in *Proc. 39th Annu. IEEE Symp. Found. Comput. Sci.*, 1998, p. 28.

[7] A. Ta-Shma, "Storing information with extractors," *Information Processing Letters*, vol. 83, no. 5, pp. 267–274, 2002 [Online]. Available: http://dx.doi.org/10.1016/S0020-0190(02)00206-5, [Online]. Available

[8] A. Ta-Shma and D. Zuckerman, "Extractor codes," in *Proc. 33rd Annu. ACM Symp. Theory Comput.*, 2001, pp. 193–199.

[9] V. Guruswami and P. Indyk, "Near-optimal linear-time codes for unique decoding and new list-decodable codes over smaller alphabets," in *Proc. 34th Annu. ACM Symp. Theory Comput.*, 2002, pp. 812–821.

[10] O. Reingold, S. Vadhan, and A. Wigderson, "Entropy waves, the zig-zag product, and new constant-degree expanders and extractors," in *Proc. 41st Annu. IEEE Symp. Found. Comput. Sci.*, 2000.

[11] J. Radhakrishnan and A. Ta-Shma, "Bounds for dispersers, extractors, and depth-two superconcentrators," *SIAM J. Discr. Math.*, vol. 13, no. 1, pp. 2–24, 2000.

[12] N. Alon, J. H. Spencer, and P. Erdös, *The Probabilistic Method*. New York: Wiley-Interscience, 1992.

[13] A. Ta-Shma, D. Zuckerman, and S. Safra, "Extractors from Reed-Muller codes," in *Proc. 42nd Annu. IEEE Symp. Found. Comput. Sci.*, Oct. 2001, pp. 638–647.

A New Bound for the Minimum Distance of a Cyclic Code From Its Defining Set

Emanuele Betti and Massimiliano Sala

***Abstract*—A new lower bound for the distance of cyclic codes is proposed. This bound depends on the defining set of the code, like several other bounds. The proposed bound improves upon the Bose–Chaudhuri–Hocquehghen (BCH) bound and, for some codes, improves upon the Hartmann–Tzeng bound and the Roos bound as well.**

***Index Terms*—Bose–Chaudhuri–Hocquehghen (BCH) bound, cyclic codes, Hamming distance, Hartmann–Tzeng bound, Roos bound.**

I. Introduction

Many lower bounds exist for the distance of cyclic codes, among others the Bose–Chaudhuri–Hocquehghen (BCH), Hartmann–Tzeng, and Roos bounds (see [1]–[5]). They are usually based on patterns in the complete defining set of the code. We present a similar bound, which is based on a pattern which has never been noted before. The proof is quite technical and relies on a method proposed by Ponchio and Sala in [6]. We will give a brief description of their method, which is called "single procedure" in [6]. Our lower bound is stronger than the BCH bound, but its relation with other classical bounds is not clear: for some codes it performs better than the Roos and the Hartmann–Tzeng bounds, while for others it performs worse.

Manuscript received December 3, 2004; revised November 23, 2005. The material in this correspondence was presented in part at the Effective Methods in Algebraic Geometry (MEGA'05) in Alghero, Italy, May 2005.

E. Betti is with the Department of Mathematics, University of Pisa, Pisa, Italy.

M. Sala is with the Boole Centre for Research in Informatics, University College Cork, Cork, Ireland.

Communicated by K. A. S. Abdel-Ghaffar, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2006.876240