Improving the alphabet-size in high noise, almost optimal rate list decodable codes

Eran Rom and Amnon Ta-Shma

Computer Science Department, Tel-Aviv University,
69978 Tel-Aviv, Israel
{eranrom,amnon}@post.tau.ac.il

Abstract. We revisit the construction of high noise, almost optimal rate list decodable code of Guruswami [1]. Guruswami showed that if one can explicitly construct optimal extractors then one can build an explicit $(1-\epsilon, O(\frac{1}{\epsilon}))$ list decodable codes of rate $\Omega(\frac{\epsilon}{log\frac{1}{\epsilon}})$ and alphabet size $2^{O(\frac{1}{\epsilon}\cdot log\frac{1}{\epsilon})}$. We show that if one replaces the expander component in the construction with an unbalanced disperser, then one can dramatically improve the alphabet size to $2^{O(log^2\frac{1}{\epsilon})}$ while keeping all other parameters the same.

1 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. 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 [2]. 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 $r\in\Sigma^N$, $|\{c\in C|\Delta(r,c)\leq pN\}|\leq K$, 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 smaller than 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,

[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.

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 $(\sqrt{1-d/N}, qN)$-list decodable (see, [3], Lecture 8). For Reed-Solomon codes there exists an efficient list decoding algorithm [4]. Thus, unique decoding is possible with at most $N/2$ errors, while by [4] list decoding is possible with up to $N - \sqrt{Nn}$ errors, and the number of all possible decodings is small.

We focus on the high noise regime, where $p = 1 - \epsilon$, for $\epsilon > 0$ being very small. 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 [1], 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 this construction are huge.

Although Guruswami's result suffers the drawbacks of huge decoding list size and huge alphabet size it is interesting as it improves our understanding of the relationships between extractors, expanders and codes. Specifically, it gives motivation for explicitly constructing better extractors which will yield better codes.

1.1 Our Results

The relationship that [1] has found between strong extractors[2] and high noise list decodable codes is given in the following theorem:

Theorem 1 (Old 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 $2^{O(\epsilon^{-1} log(1/\epsilon))}$*

We show:

Theorem 2 (New 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 $2^{2^{polyloglog(\frac{1}{\epsilon})}}$.*[3]

Note that all parameters are the same as in [1], except the significantly smaller alphabet size. Using the best explicit construction of strong extractors we have today [1] has shown:

Theorem 3 (Old connection between explicit strong extractors and L.D.C.). *For every constant $\epsilon > 0$, there is a polynomial time constructible*

[2] The definition of strong extractors and dispersers is given in Sect. 1.2

[3] If we further assume an optimal disperser then the alphabet size can be reduced to $2^{O(\log^2 \frac{1}{\epsilon})}$

family of $(1-\epsilon, 2^{O(\sqrt{n \log n})})$*-list decodable codes of rate* $\Omega(\frac{\epsilon}{polylog(1/\epsilon)})$ *over an alphabet of size* $2^{O(\epsilon^{-1}\log(1/\epsilon))}$*, where* n *is the dimension of the code.*

Again, we show that the alphabet size can be improved:

Theorem 4 (New connection between explicit strong extractors and L.D.C.). *For every constant* $\epsilon > 0$*, there is a polynomial time constructible family of* $(1-\epsilon, 2^{O(\sqrt{n \log n})})$*-list decodable codes of rate* $\Omega(\frac{\epsilon}{polylog(1/\epsilon)})$ *over an alphabet of size* $2^{2^{polyloglog(\frac{1}{\epsilon})}}$*, where* n *is the dimension of the code.*

1.2 The Technique

In order to understand our technical contribution we first need to understand what Guruswami did in [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 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, ζ_{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* $U_{[D]\times[M]}$*, where* x *is drawn from* X *and* y *is taken uniformly at random from* $[D]$*. The entropy loss of the strong extractor is* $\frac{K}{M}$*.* ζ_{ext} *is called the extractor error. The strong extractor is explicit if* $f(x,y)$ *can be computed in time polynomial in the input length, i.e., polynomial in* $\log N + \log D$*.*

That is 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, ζ_{disp})-disperser if for every $X \subseteq [L]$ with $|X| \geq H$ we have $|\{g(l,j) | l \in X, j \in [T]\}| \geq (1-\zeta_{disp})D$. 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.

Extractor Codes. [5] 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 $(1-(\frac{1}{M}+\zeta_{ext}), K)$-list decodable code.*

Also observed by [5] is that extractor codes meet a property stronger than list decoding, known as list recovering [6]. List recovering deals with the situation where the i^{th} 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.

Error Amplification Using Expanders. The technique of code amplification using expanders was introduced in [7], where it is used to amplify Justesen code. Justesen code rate vanishes as the error rate grows. [7] take Justesen code of constant error rate and amplify it using an expander to get a code with large distance and constant rate over a large alphabet. Looking back, the amplification in [7] can be done using any *disperser* (a good expander is just a special case).

Here is how the amplification is done: Assume $C : \Sigma^n \to \Sigma^D$ is a (p,K)-list decodable code. Let $g : [L] \times [T] \to [D]$ be a (H, ζ_{disp})-disperser. Define a code $C_g : \Sigma^n \to [\Sigma^T]^L$. 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 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 $C_g : \Sigma^n \to [\Sigma^T]^L$. A simple argument shows:

Lemma 2. *If $\zeta_{disp} \leq p$, then $C_g(x)$ is $(1-\frac{H}{L}, K)$-list decodable.*

Guruswami's Construction. Lemma 1 shows that an extractor code already has a good error rate. However, the lower bound of $\Omega(\frac{1}{\zeta_{ext}^2})$ on the extractor degree (see Sect. 2.1) implies an upper bound of $O(\zeta_{ext}^2)$ on the extractor code's rate. To overcome this obstacle, Guruswami uses a strong extractor with a constant error and amplifies it using a balanced expander, see figure 1. As shown in Section 3, the construction takes advantage of the list recovering property of extractor codes.

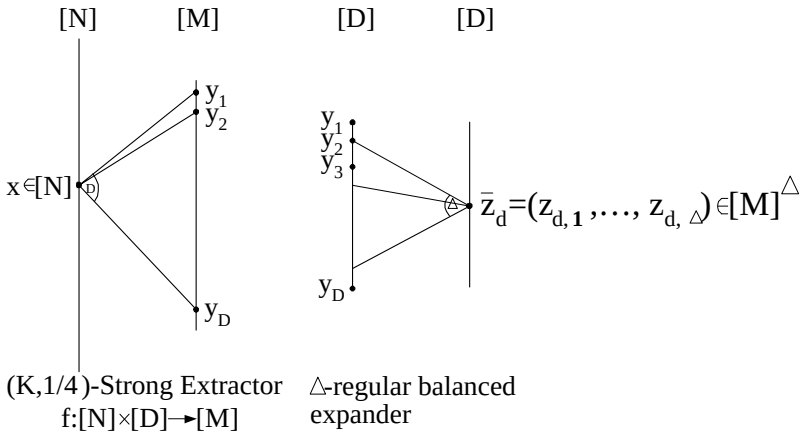

Fig. 1. *The strong extractor on the left has a constant error, giving an extractor code with* $\frac{3}{4} - \frac{1}{M}$ *error rate. The code produced by the extractor is then amplified using the balanced expander on the right. The balanced expander used is equivalent to some* $(\epsilon D, \zeta_{disp})$*-disperser* $g : [D] \times [\Delta] \to [D]$.

Our Improvement. What we do is replace the expander component in [1] with a good *unbalanced* disperser. Studying the problem we discover that what is needed is a disperser for the high min–entropy rate, that has optimal entropy loss and a surprisingly small degree. Fortunately, an explicit construction of such a graph was given recently [8]. Using such a graph, our improvement over the construction in [1] 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 [1] and the time it takes does not change.

2 Preliminaries

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 X, Y distributed 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 $|X-Y|\leq\epsilon$.

2.1 Bounds of the Parameters Achievable for Extractors and Dispersers

Ta-Shma and Radhakrishnan [9] show that a (K,ζ_{ext})-strong extractor $f:[N]\times[D]\rightarrow[M]$ must have degree $D=\Omega(\frac{1}{\zeta_{ext}^2}\log\frac{N}{K})$, and entropy loss $\frac{K}{M}=O(\frac{1}{\zeta_{ext}^2})$. Also shown in [9] 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. [9] also give matching lower bounds and non-explicit upper bounds for dispersers. Specifically, a (H,ζ_{disp})-disperser $g:[L]\times[T]\rightarrow[D]$ must have $T=\Omega(\frac{1}{\zeta_{disp}}\log\frac{L}{H})$, and entropy loss $\frac{HT}{D}=\Omega(log\frac{1}{\zeta_{disp}})$.

2.2 The Mixing Property

An important property of extractors is mixing (see, [10], Chap 9). We introduce some notation. For $x\in[N]$ we define $\Gamma_f(x)$ to be the ordered neighbors of x. Formally,

$$\Gamma_f(x)=\{(i,f(x,i)|i\in[D]\}\ . \tag{1}$$

The mixing property says that:

Fact 5 *If $f:[N]\times[D]\rightarrow[M]$ 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*

$$\frac{|\Gamma_f(x)\cap S|}{D}-\frac{|S|}{D\cdot M}\geq\zeta_{ext}\ . \tag{2}$$

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

The connection shown below is basically Guruswami's, except that Guruswami uses a balanced, good expander and we use a slightly unbalanced good disperser. Let: $f:[N]\times[D]\rightarrow[M]$ be a (K,ζ_{ext})-strong extractor, and let $g:[L]\times[T]\rightarrow[D]$ be a (H,ζ_{disp})-disperser. We define the code $C_{f,g}:[N]\rightarrow[M^T]^L$ as follows:

1. Given $x\in[N]$, denote by $\overline{y}=(y_1,\ldots,y_D)\in[M]^D$ where $y_i=f(x,i)$.
2. Put the symbols $(y_1,\ldots,y_D)\in[M]^D$ along g's range $[D]$. Each element $\ell\in[L]$ has T neighbors in $[D]$. Collect from each neighbor the symbol that was put along it. I.e., for each $\ell\in[L]$ define $\overline{w}_\ell=(w_{\ell,1},\ldots,w_{\ell,T})\in[M]^T$, where $w_{\ell,t}=y_{g(\ell,t)}$.
3. The encoding of x is defined to be

$$C_{f,g}(x)=(\overline{w}_1,\ldots,\overline{w}_L)\ . \tag{3}$$

See figure 2 for an illustration of the construction. We claim:

Lemma 3. *If the extractor f and the disperser g are as above, and if $M \cdot D \geq \frac{L \cdot T}{1-\zeta_{ext}-\zeta_{disp}}$, then $C_{f,g}$ is $(1-\frac{H}{L}, K)$-list decodable.*

An eye on figure 2 might be helpful while reading the proof.

Proof. Let $z = (\overline{z}_1, \ldots, \overline{z}_L) \in [M^T]^L$ be an arbitrary word in $[M^T]^L$. From z we build a set S as follows. For each $1 \leq \ell \leq L$, we look at $\overline{z}_\ell = z_{\ell,1}, \ldots, z_{\ell,T}$ and we build the set $S_\ell \subseteq [D] \times [M]$ by:

$$S_\ell = \{(g(\ell,t), z_{\ell,t}) | 1 \leq t \leq T\} \ . \tag{4}$$

S_ℓ represents what $\overline{z}_\ell$ thinks $y_1, \ldots, y_D$ are in locations $g(\ell,1), \ldots, g(\ell,T)$. We define the set S of $z = (\overline{z}_1, \ldots, \overline{z}_L)$ to be $\bigcup_{\ell=1}^{L} S_\ell$.

Suppose a codeword $C_{f,g}(x) \in [M^T]^L$ agrees with z on a set $\mathcal{H} \subseteq [L]$ of size at least H. 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 (because the l^{th} coordinate is the concatenation of all the symbols along the neighbors of ℓ in g). Since g is a (H, ζ_{disp})-disperser, the set of neighbors of $\mathcal{H}$ has at least $(1-\zeta_{disp})D$ elements. Hence, $|\Gamma_f(x) \cap S| \geq (1-\zeta_{disp})D$. Noting that $|S| \leq L \cdot T$, and using the assumption $M \cdot D \geq \frac{L \cdot T}{1-\zeta_{ext}-\zeta_{disp}}$, we see that $\frac{|S|}{MD} \leq 1-\zeta_{ext}-\zeta_{disp}$ and together

$$\frac{|\Gamma_f(x) \cap S|}{D} - \frac{|S|}{MD} \geq \zeta_{ext} \ . \tag{5}$$

By Fact 5 we conclude that there are at most K x's for which $C_{f,g}(x)$ agrees with z on at least H coordinates. Hence the code is $(1-\frac{H}{L}, K)$-list decodable. □

3.1 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 figure 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 H) on the right hand side (of figure 2) sees almost all of the vertices on the left hand side as its neighbors (more precisely $1-\zeta_{disp}$ of them).

Taking a balanced expander does the job, but at the cost of enlarging the disperser degree T. This is because H vertices can have at most HT neighbors, and so if H is small and HT is almost D, we must have a disperser with a large degree T. On the other hand, if we take a larger right hand side L (such that H is roughly D) we can use a much smaller degree T and still have the same property.

To see how the parameters behave, we notice that the size of the alphabet is determined by T (and so we get a much smaller alphabet size) and the rate is

determined by $L \cdot T$ and L should be H/ϵ (to provide the necessary amplification). The fact that L is larger does not translate to an inferior rate, because T is smaller and so $L \cdot T$ stays exactly as in Guruswami's construction. We thus keep the rate while dramatically reduce the alphabet size.

For this to work we need a good disperser that works for the high min–entropy setting (where H is very close to L) and tiny degree (the optimal is $O(\log(L/H))$. Luckily, the recent Zig-Zag construction [8] explicitly constructs such a graph.

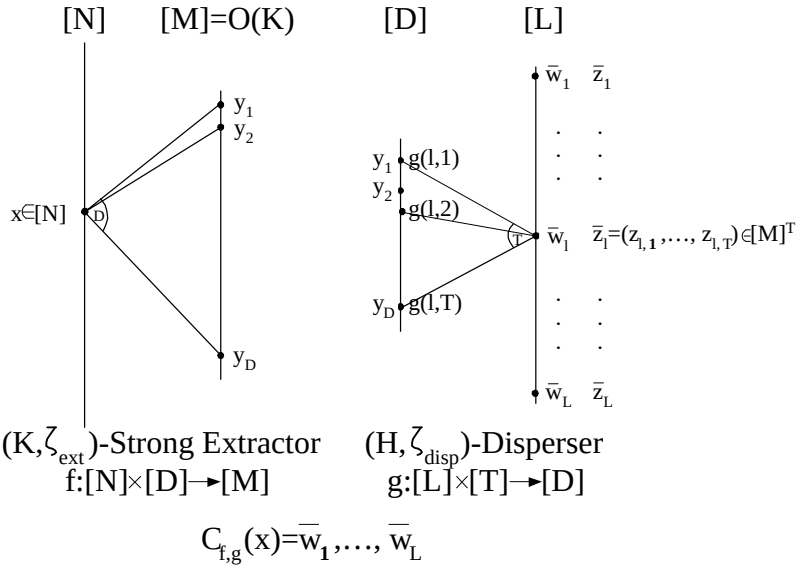

Fig. 2. *The neighbors of $x \in [N]$ on the left: $(y_1, \ldots, y_D)$ are "put" along the disperser's output $[D]$, defining for each $l \in [L]$ an ordered vector of its neighbors $\overline{w}_l = (w_{l,1}, \ldots, w_{l,T}) \in [M]^T$. The vector $\overline{w}_l$ is the l^{th} symbol in the codeword $C_{f,g}(x)$. $z = (\overline{z}_1, \ldots, \overline{z}_L)$ is an arbitrary word in $[M^T]^L$. $\overline{z}_l \in [M]^T$ is the l^{th} coordinate of z and $g(l,1), \ldots, g(l,T)$ are the neighbors of $l \in [L]$ in g. Each neighbor $g(l,t)$ of l is associated with $z_{l,t} \in [M]$, the t^{th} element of $\overline{z}_l$. The pairs $\{(g(l,t), z_{l,t})\}_{t=1,\ldots,T}$ form the set $S_l \subset [D] \times [M]$. An agreement between $\overline{z}_l$ and $\overline{w}_l$ implies that $y_t = z_{l,t}$ for all $1 \leq t \leq T$, i.e. agreement on t out of the D symbols $(y_1, \ldots, y_D)$. An agreement between z and $C_{f,g}(x)$ on H coordinates will thus translate to an agreement on $(1 - \zeta_{disp})D$ of the symbols $(y_1, \ldots, y_D)$.*

3.2 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 3. Since the components' lowers bounds have matching non-explicit upper bounds, the parameters we find give a non-explicit construction for the desired code.

The Constraints. First, we write down all the constraints. The bounds we give are both lower bounds, and achievable by non-explicit constructions. We have:

$$D = \Omega\left(\frac{1}{\zeta_{ext}^2} \cdot \log\frac{N}{K}\right) \quad . \tag{6}$$

$$M = O\left(K\zeta_{ext}^2\right) \quad . \tag{7}$$

$$T = \Omega\left(\frac{1}{\zeta_{disp}} \cdot \log\frac{L}{H}\right) \quad . \tag{8}$$

$$D = O\left(\frac{HT}{\log\frac{1}{\zeta_{disp}}}\right) \quad . \tag{9}$$

$$M \cdot D \geq \frac{L \cdot T}{1 - \zeta_{ext} - \zeta_{disp}} \quad . \tag{10}$$

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 $[D] \times [M]$.

A Specific Choice of Parameters. We now choose parameters. We first set $\zeta_{ext}, \zeta_{disp}$ 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 to be $K = \Theta(\frac{1}{\epsilon})$. With these choices we have $D = \Theta(\log(N))$, and $M = \Theta(K) = \Theta(\frac{1}{\epsilon})$. We also set $\frac{H}{L} = \epsilon$. This implies that $T = \Theta(\log(\frac{L}{H})) = \Theta(\log\frac{1}{\epsilon})$. To satisfy Equation (9) we need to take $H = \Theta(\frac{D}{T}) = \Theta(\frac{\log(N)}{\log\frac{1}{\epsilon}})$ which implies that $L = \frac{H}{\epsilon} = \Theta(\frac{\log(N)}{\epsilon \cdot \log(\frac{1}{\epsilon})})$. Finally, we check Equation (10). 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,

$$K = \Theta\left(\frac{1}{\epsilon}\right) \quad . \tag{11}$$

$$D = \Theta\left(n\right) \quad . \tag{12}$$

$$M = \Theta\left(\frac{1}{\epsilon}\right) \quad . \tag{13}$$

$$L = \Theta\left(\frac{n}{\epsilon \cdot \log(\frac{1}{\epsilon})}\right) \quad . \tag{14}$$

$$H = \Theta\left(\frac{n}{\log(\frac{1}{\epsilon})}\right) \quad . \tag{15}$$

$$T = \Theta\left(\log\frac{1}{\epsilon}\right) \quad . \tag{16}$$

Thus, $rate = \frac{\log N}{L \cdot T \log M}$ is $\Theta\left(\frac{\epsilon}{\log(\frac{1}{\epsilon})}\right)$, 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 footnote of Theorem 2.

4 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 $\zeta_{ext} = \zeta_{disp} = \frac{1}{4}$. We note that (10) now becomes, $L \cdot T \leq \frac{1}{2} \cdot M \cdot D$

4.1 Using Explicit High Min–entropy Optimal Loss Disperser

As suggested by the parameters chosen in 3.2, the construction requires a high min–entropy disperser with optimal entropy loss. Such a disperser is given by [8] in the extractors' analogue of the Zig-Zag construction. Specifically:

Fact 6 ([8]) *For every L and $\epsilon > \frac{1}{\sqrt{L}}$, there exists an explicit construction of $(\epsilon L, \frac{1}{4})$-disperser $g : [L] \times [T] \to [D]$, with $T = 2^{polyloglog(\frac{1}{\epsilon})}$, and entropy loss $\frac{\epsilon L \cdot T}{D} = O(1)$.*

We now prove Theorem 2:

Proof. The disperser from Fact 6 has $\Theta(1)$ entropy loss. Let C_1, C_2 be the constants which bound this entropy loss from below and from above accordingly. Let C_3 be the constant behind the extractor degree $O(\cdot)$ notation from the assumption, where $D = O(\log N)$.

Let $\epsilon > 0$, and $T = 2^{polyloglog(\frac{1}{\epsilon})}$ as in Fact 6. We let $M = \frac{2 \cdot C_2}{\epsilon}$, $N > 2^{\frac{2T}{\epsilon \cdot C_3 (C_1 + C_2)}}$, $D = C_3 \log N$, $L = \frac{C_1 + C_2}{2} \cdot \frac{D}{\epsilon T}$ and $K = \Theta(M)$. By the choice of N, we have that $\epsilon > \frac{1}{\sqrt{L}}$. Thus, by Fact 6 there is an explicit construction of $(\epsilon L, \frac{1}{4})$-disperser $g : [L] \times [T] \to [D]$ and by the assumption there is a $(K, 1/4)$-strong extractor $f : [N] \times [D] \to [M]$. Finally, by the choice of M, $LT \leq \frac{1}{2} MD$, and we satisfy constraint (10). Lemma (3) now gives the desired list decodable code. □

4.2 Using an Explicit Extractor

As mentioned in [1], and shown in Sect. 5, in order to keep the rate strictly greater than zero, we need an extractor with degree $D = O(\log N)$. The best explicit construction to date of a strong extractor, which achieves the required degree is due to [11].

Fact 7 ([11]) *For Every m $=m(n)$, $k = k(n)$ and $\zeta = \zeta(n)$ such that $3m\sqrt{n\log(n/\zeta)} \leq k \leq n$, there is an explicit family of (k,ζ)-strong extractors $E_n : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = \log n + O(\log \frac{m}{\zeta})$.*

Denoting $N = 2^n$, $K = 2^k$, $D = 2^d$, and $M = 2^m$, Plugging the above extractor in the construction, we prove Theorem 4:

Proof. Let $\epsilon > 0$, $T = 2^{polyloglog(\frac{1}{\epsilon})}$ as in Fact 6. Let C_1, C_2 be as in the proof of Theorem 2. Let C_3 be the constant behind the extractor degree $O(\cdot)$ notation from Fact 7, where $d = \log n + O(\log \frac{m}{\zeta})$.

Let $M = \frac{2 \cdot C_2}{\epsilon}$, $N > 2^{\frac{2T}{\epsilon \cdot (C_1+C_2) \cdot 2^{2 \cdot C_3} \cdot (\log M)^{C_3}}}$, such that $2^{\sqrt{\frac{\log N}{\log\log N}}} > M^6$. Let $K = M^6\sqrt{\log N \log\log N}$, $D = 2^{2 \cdot C_3} \log N (\log M)^{C_3}$ and $L = \frac{C_1+C_2}{2} \cdot \frac{D}{\epsilon T}$. By Fact 7 and by the choice of N, K and M there is an explicit $(K, \frac{1}{4})$-strong extractor $f : [N] \times [D] \to [M]$. By the choice of N, $\frac{1}{\sqrt{L}} < \epsilon$ and there is an explicit $(\epsilon L, \frac{1}{4})$ disperser $g : [L] \times [T] \to [D]$. Finally, by the choice of M we have $LT \leq \frac{1}{2}MD$, and constraint (10) is satisfied. Applying Lemma 3 gives the desired code.

Encoding an element $x \in [N]$ is consisted of finding $\Gamma_f(x)$, and finding $\Gamma_g(l) = \{g(l,j) | j \in [T]\}$ for each $l \in [L]$. Thus, the explicitness of the code is straightforward from the explicitness of the disperser g and the extractor f above. □

5 On the Optimality of the Parameters Choice

We now take a closer look at the parameters. Specifically, we show that the parameters chosen in Sect. 3.2, 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. The following claim summarizes the various relations between the parameters and their optimality.

Lemma 4. *In the construction given in Sect. 3, for any choice of parameters satisfying error rate of $1-\epsilon$ the following holds:*

1. *Decoding list size of $O(\frac{1}{\epsilon})$ and rate bounded away from zero implies that the rate and alphabet size cannot be better (up to constant factor) than those in Sect. 3.2.*
2. *Decoding list size of $O(\frac{1}{\epsilon})$ implies disperser and extractor with optimal entropy loss (namely, $\frac{HT}{D} = O(1)$ and $\frac{K}{M} = O(1)$).*
3. *An almost optimal rate of $O(\frac{\epsilon}{\log \frac{1}{\epsilon}})$ implies extractor's degree $D = O(\log N)$.*
4. *Almost optimal rate of $O(\frac{\epsilon}{\log \frac{1}{\epsilon}})$ and $|\Sigma| = 2^{O(\log^2 \frac{1}{\epsilon})}$ imply an optimal entropy loss disperser (namely, $\frac{H \cdot T}{D} = O(1)$).*

For lack of space we give only the proof of 1.

Proof. Having an error rate of $1-\epsilon$, we have $H = \epsilon L$. We first show that M must be $\Theta(\frac{1}{\epsilon})$: Decoding list of size $\frac{1}{\epsilon}$ implies that $K = \frac{1}{\epsilon}$, and by (7) $M = O(K) =$

$O(\frac{1}{\epsilon})$. By (10) $M \geq \frac{L \cdot T}{D}$. Since $L = \frac{H}{\epsilon}$, and since (9) implies $T = \Omega(\frac{D}{H})$, we have $M \geq \frac{L \cdot T}{D} = \Omega(\frac{1}{\epsilon})$. Altogether $M = \Theta(\frac{1}{\epsilon})$. By the construction, $rate = \frac{\log N}{L \cdot T \log M}$, (10) implies $L \cdot T \leq M \cdot D$, and so $rate \geq \frac{\log N}{M \cdot D \log M} = \Theta(\frac{\epsilon \log N}{D \log \frac{1}{\epsilon}})$. Thus, in order to bound the rate away from zero, we must take $D = O(\log N)$, which gives $rate = \Omega(\frac{\epsilon}{\log \frac{1}{\epsilon}})$. As for the alphabet size $|\Sigma| = M^T$. $M = \Omega(\frac{1}{\epsilon})$, and by (8) $T = \Omega(\log \frac{1}{\epsilon})$, thus, $|\Sigma| = (\frac{1}{\epsilon})^{\Omega(\log \frac{1}{\epsilon})}$ □

Acknowledgements

We would like to thank the anonymous referees for numerous comments which improved and clarified the final version of this paper a lot.

References

1. Guruswami, V.: Better extractors for better codes? In: Proceedings of the 36th Annual ACM Symposium on Theory of Computing, ACM Press (2004) 436–444
2. Elias, P.: List decoding for noisy channels. In: 1957-IRE WESCON Convention Record, Pt. 2. (1957) 94–104
3. Sudan, M.: Lecture Notes on Algorithmic Introduction to Coding Theory. (http://theory.lcs.mit.edu/∼madhu/FT01/scribe/overall.ps)
4. Guruswami, V., Sudan, M.: Improved decoding of reed-solomon and algebraic-geometric codes. In: Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society (1998) 28
5. Ta-Shma, A., Zuckerman, D.: Extractor codes. In: Proceedings of the 33rd Annual ACM Symposium on Theory of Computing. (2001) 193–199
6. Guruswami, V., Indyk, P.: Near-optimal linear-time codes for unique decoding and new list-decodable codes over smaller alphabets. In: Proceedings of the 34th Annual ACM Symposium on Theory of Computing, ACM Press (2002) 812–821
7. Alon, N., Bruck, J., Naor, J., Naor, M., Roth, R.: Construction of asymptotically good, low-rate error-correcting codes through pseudo-random graphs. IEEE Transactions on Information Theory **38** (1992) 509–516
8. Reingold, O., Vadhan, S., Wigderson, A.: Entropy waves, the zig-zag product, and new constant-degree expanders and extractors. In: Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science. (2000)
9. Radhakrishnan, J., Ta-Shma, A.: Bounds for dispersers, extractors, and depth-two superconcentrators. SIAM Journal on Discrete Mathematics **13** (2000) 2–24
10. Alon, N., Spencer, J.H., Erdős, P.: The Probabilistic Method. Wiley–Interscience Series, John Wiley & Sons, Inc., New York (1992)
11. Ta-Shma, A., Zuckerman, D., Safra, S.: Extractors from Reed-Muller codes. In: Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science. (2001) 638–647