Linear time Encodable/Decodable Codes with Near-Optimal Rate[*]

Venkatesan Guruswami[†]   Piotr Indyk[‡]

May 18, 2004

Abstract

We present an *explicit* construction of *linear-time encodable and decodable* codes of rate r which can correct a fraction $(1 - r - \varepsilon)/2$ of errors over an alphabet of constant size depending only on ε, for every $0 < r < 1$ and arbitrarily small $\varepsilon > 0$. The error-correction performance of these codes is *optimal* as seen by the Singleton bound (these are "near-MDS" codes). Such near-MDS linear-time codes were known for the decoding from erasures [2]; our construction generalizes this to handle errors as well. Concatenating these codes with good, constant-sized binary codes gives a construction of linear-time binary codes which meet the Zyablov bound, and also the more general Blokh-Zyablov bound (by resorting to multilevel concatenation).

Our work also yields linear-time encodable/decodable codes which match Forney's error exponent for concatenated codes [8] for communication over the binary symmetric channel. The encoding/decoding complexity was quadratic in Forney's result, and Forney's bound has remained the best constructive error exponent for almost 40 years now.

In summary, our results match the performance of the previously known explicit constructions of codes that had polynomial time encoding and decoding, but in addition have linear time encoding and decoding algorithms.

1   Introduction

We present constructions of asymptotically good error-correcting codes whose rate vs. distance trade-off matches the best known trade-offs for explicit codes, and furthermore our codes come with linear time encoding and decoding algorithms (that correct up to almost half the minimum distance). Thus the encoding/decoding times are optimal up to constant factors in the running time, and there is essentially no compromise in terms of the rate of the codes compared to the previously known results with polynomial time encoding/decoding.

Our first result is a construction of a code family with the best possible asymptotic rate vs. distance trade-off (namely, the *Singleton bound*), together with linear time encoding and linear time decoding algorithms up to half the (designed) distance. Specifically, for each distance, the rate achieved by our code matches the best achievable rate up to a multiplicative factor of $(1+\varepsilon)$, for an arbitrarily small $\varepsilon > 0$. In addition, the codes have alphabet size a constant that depends only on

---

ε; however, and this is the main shortcoming of our result, the alphabet size grows exponentially in $1/\varepsilon$. Thus, our constructions are essentially optimal in three respects: they can achieve any desired distance including a distance arbitrarily close to the codeword length; they achieve rate arbitrarily close to the optimum for each value of the distance; and they are encodable and decodable in linear time. The only criterion with respect to which our codes are sub-optimal is the alphabet size, which (although constant) depends exponentially on $1/\varepsilon$ (the best alphabet size one can hope for is $O(1/\varepsilon)$). A result similar to ours was known for the easier problem of decoding from erasures [2], and indeed we use techniques from that paper in our construction.

Concatenating these with good binary inner codes enables us to match the *Zyablov* bound [27] with linear-time decoding. One can also use the multilevel concatenation technique of [5] to get linear-time codes that match the *Blokh-Zyablov* bound as well. We note that these bounds give essentially the best known rate vs. distance trade-off for explicit binary codes. (The only better bound known for constructive code is obtained by concatenating certain sophisticated algebraic-geometric codes with inner codes that lie on the Gilbert-Varshamov bound. But even with the current fastest known constructions [21], this has very high construction and decoding complexity; see [6] for further discussion about this point.)

We are able to attain the stated bounds together with linear time encoding and decoding algorithms. In comparison, the first constructions of linear time encodable/decodable codes [24, 23] could not achieve arbitrarily large distance. While this aspect was fixed in the recent work [12] which obtained linear-time codes of any desired distance, the rate of these codes was sub-optimal.

Our work also yields linear-time encodable/decodable codes which match Forney's error exponent for concatenated codes [8] for the binary symmetric channel (the encoding/decoding complexity was quadratic in Forney's result).

To summarize, we obtain what was earlier possible with reasonable polynomial time construction complexity with linear time algorithms. Our result is a major step towards the resolution of a grand project proposed by Spielman in his Ph.D. thesis [23], namely, to "obtain linear-time encodable and decodable error-correcting codes with rate and error tolerance as good as the best known codes, whatever their algorithmic efficiency".

The technique behind our construction gives a fairly general way of using highly expanding graphs to construct and decode codes from a large fraction of errors. In fact, in a preliminary version of this paper that appeared as [13], we also use it to construct a *list* decodable code with small alphabet size and good rate. In this paper, however, we focus only on results for unique decoding.

Subsequent Work. Our work was the first to attain the Zyablov bound together with linear complexity algorithms. Subsequent to our work, Barg and Zémor [4] gave a different construction of linear-time decodable codes that meets the Zyablov bound, by extending the expander codes scheme of Sipser-Spielman and Zémor [22, 26] using "parallel concatenation". We point out that their results do not give linear time encoding algorithms (but since the constructed codes are linear, a quadratic time encoding algorithm is immediate).

2   Linear-time Near-MDS codes over large alphabets

We now present constructions of linear-time encodable and decodable codes over large alphabets that get arbitrarily close to the Singleton bound and thus achieve the optimal rate vs. error-correction trade-off (not to forget the linear time encoding and decoding algorithms). Once we construct codes over the large alphabet that match the Singleton bound, getting binary codes that

meet the Zyablov bound is fairly standard and follows by concatenation with a constant-sized inner code that meets the "Gilbert-Varshamov bound". We will discuss this formally in Section 3. For now, we focus on the construction of the large alphabet code.

2.1   High-level view of the construction

Before delving into the formal construction, we describe the high-level idea behind the construction (reading what follows with an eye on Figure 1 might be useful). Our code is constructed by combining three objects, a "left" code C, a constant-sized MDS (say, Reed-Solomon) code $\tilde{C}$, and a suitable bipartite expander graph G (say, with n vertices on each side). The message will be first encoded by the left code C. The resulting codeword of C will then be broken into n blocks, each of constant size, and each of these blocks will be encoded by the Reed-Solomon code $\tilde{C}$. The symbols of the resulting string will then be redistributed using the edges of the expander G, the symbols in the encoding of the i'th block being sent to the neighbors of the i'th node on the left side of G. Now, the final codeword (of length n) is obtained by "juxtaposing" or "concatenating" the symbols received at each of the n vertices on the right. The construction scheme is similar in spirit to earlier expander-based code constructions in [1, 2, 12], and specifically the construction of near-MDS erasure codes in [2].

We now elaborate a bit on how we pick each of these components. The left code C will be a linear-time code of rate very close to one, say, $(1 - \gamma)$ for some small $\gamma > 0$, which can correct a fraction $\Theta(\gamma^2)$ of errors in linear time. The code $\tilde{C}$ will be a Reed-Solomon code of rate (very close to) r. Its block length will be equal to the degree D of the expander. For the graph G, we can take any expander whose second eigenvalue λ is much smaller than its degree D; in order to get the best parameters (specifically, alphabet size), we use a Ramanujan graph which satisfies $\lambda = O(\sqrt{D})$.

The code $\tilde{C}$ and the expander are standard and we just use them "off-the-shelf". For the left code C, the existing construction of linear-time encodable/decodable codes due to Spielman [24, 23] do not give this directly, as even to correct a very small fraction of errors, the rate has to be an absolute constant bounded away from 1. However, as Spielman [23] remarks it is possible to pick parameters differently in his construction and achieve any rate, though the formal details have not been made explicit anywhere. Here, we present a new construction which has the property necessary to us; our construction is obtained by combining ideas from [2] and [26]. Our construction also achieves a slightly better dependence between the fraction of errors corrected and the rate (compared to what can be deduced by working through the construction in [24]); this translates into a slightly better alphabet size for our overall construction. The construction of this left code is in fact our main technical contribution in this section, as the rest of the construction pretty much follows the scheme used in [2] for near-MDS erasure codes. We discuss this construction next, before moving on to the construction of the final near-MDS code.

2.2   Linear-time codes with rates close to 1

In this section, we describe a code construction that will serve the role of the "left code" in the construction scheme of Figure 1. The required qualitative properties from these codes is that they be able to correct a small constant fraction β of of errors and have rate approaching 1 as $\beta \to 0$; the exact dependence of how close the rate is to 1 as a function of the fraction of errors corrected is not important. In fact it is this trade-off that we will improve to near-optimal in Section 2.3.

Lemma 1 *For every $\gamma > 0$, there is an explicitly specified code of rate $1/(1 + \gamma)$ over an alphabet of size $q = O(1/\gamma^2)$ such that a code of block length N in the family can be encoded in $O(N/\gamma)$ time*

Expander Graph

< *, *, a, *, * >

Each symbol of C^* is the juxtaposition of the symbols it receives from the left.

Codeword of linear–time left code C broken into blocks

Each block is encoded by constant–sized Reed–Solomon code $\tilde{C}$
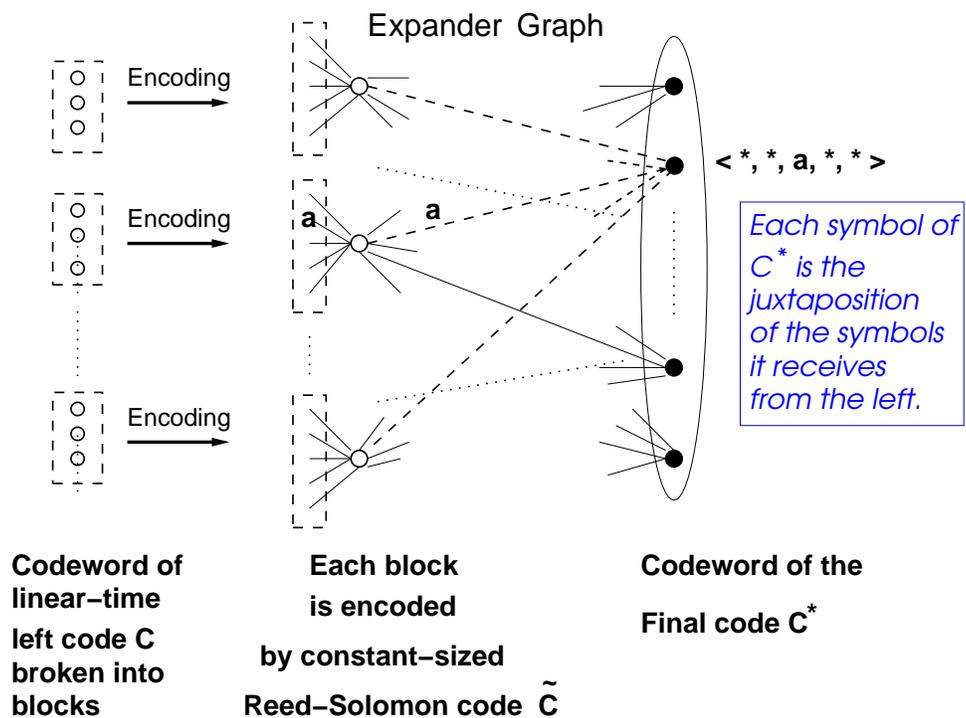
Codeword of the Final code C^*

Figure 1: Basic structure of the construction of near-MDS linear time codes. The "left" code is first broken into blocks and each block encoded by a constant-sized Reed-Solomon code $\tilde{C}$. Note that the second symbol **a** of the encoded block is sent to the second neighbor of the corresponding node of the expander. This is in general how symbols are redistributed from the left to the right using the expander. On the right side, the symbol at each position is the juxtaposition of the symbols received from the neighbors on the left. (For example, in the figure the second position receives **a** from its third neighbor on the left, and therefore has **a** at the third position of the 5-tuple of symbols that it receives.) This yields the overall encoding, and we denote by C^* the code obtained by the combination of all the encoding steps.

and can be decoded from a fraction $\beta = O(\gamma^2)$ of errors in $O(N/\gamma^2)$ time.

Proof: For infinitely many values of m and for some fixed $q = O(1/\gamma^2)$, we will construct a code over $\mathrm{GF}(q)$ of dimension m and block length $N = (1 + \gamma)m$ which can be encoded in linear time and can be decoded from βm errors in linear time for $\beta = \Theta(\gamma^2)$. The encoding will work in two steps. In the first step, the message is encoded by a code C_1 into a string of length $(1 + 2\gamma')m$ comprising of the m message symbols and $2\gamma'm$ check symbols (we take $\gamma' = \gamma/8$). This code has the property that given the correct values to all of the check symbols, an arbitrary set of βm errors in the message symbols can be corrected. In the second step, the check symbols are further encoded by a linear-time rate $1/4$ code C_2 that can correct up to βm errors. The combined code thus maps m symbols into $(1 + 8\gamma')m = (1 + \gamma)m$ symbols and can correct up to βm errors. The decoding algorithm for the combined code from βm errors is the obvious one: first decode C_2 to correct any errors in the check bits, and then decode C_1 to correct, using the correct values of the check bits, the up to βm errors that could exist in the message bits.

For the code C_2, we can use the codes due to Spielman which have some constant rate. Specifically, as stated in [2], there is an explicit such code C_2 over $\mathrm{GF}(q)$ of rate $1/4$ which can correct a fraction b of errors for some absolute constant $b > 0$ that is independent of γ. The qualitative feature that is important about C_2 is that its rate and fraction of correctable errors both be absolute constants (independent of γ); the exact values of these constants are not important and therefore we can get away with just using the original Spielman code. It remains to describe the code C_1. The code C_1 must encode m symbols into $(1 + 2\gamma')m$ symbols such that the encoding can be performed in linear time and moreover C_1 can be decoded from up to βm errors in the message bits, where $\beta = O(\gamma'^2)$, in linear time.

Let H be a d-regular bipartite "Ramanujan" expander with m edges and $n = m/d$ vertices on each side, such that the second largest eigenvalue λ of its adjacency matrix satisfies $\lambda \leq 2\sqrt{d}$. Here d is a constant that is independent of n, i.e., we use a family of constant-degree expanders (jumping ahead $d = O(1/\gamma^2)$ will suffice). The m positions of the message to be encoded are identified with the edges of H. For each vertex v of H, we compute $\gamma'd$ check symbols corresponding to the message symbols on edges incident upon v. These are computed using some systematic MDS code C' of dimension d, block length $(1 + \gamma')d$, and which can correct fewer than $\gamma'd/2$ errors; for example we can use a Reed-Solomon code over a field of size $O(d)$. In all, this gives $2n(\gamma'd) = 2\gamma'm$ check symbols, as required.

It is clear that C_1 can be encoded in linear time, since each of the n MDS codes is of constant-size. We now discuss the linear-time decoding algorithm for C_1 that corrects up to βm errors in the message symbols, given the correct values of all check symbols. This algorithm and its analysis follows along the lines of Zemor's recent improvement [26] of the analysis of Sipser and Spielman [22]. For completeness sake, we next present the details of this analysis.

Let the two sides of the bipartition of H be A and B. For each $v \in A \cup B$ denote by E_v the set of edges of H incident on v. Let $x \in \mathrm{GF}(q)^m$ be the portion of the received word corresponding to the m message symbols — by hypothesis, x is the message vector corrupted by at most βm errors. Let $y \in \mathrm{GF}(m)^{\gamma'm}$ be the vector of the check symbols. Denote by x_{E_v} the projection of x on the d edges in E_v, and by y_{E_v} the projection of y to the $\gamma'd$ check symbols that correspond to the encoding by the MDS code C' of the symbols on the edges in E_v. The decoding algorithm proceeds in rounds, and in each round does the following in sequence:

(a) (Left wing decoding) For each $v \in A$ in parallel, check if there exists a vector $z \in \mathrm{GF}(q)^d$ within distance $\gamma'd/2$ of x_{E_v} and whose check bits agree with y_{E_v}; if so, set x_{E_v} to z.

(b) (Right wing decoding) For each $v \in B$ in parallel, check if there exists a vector $z \in \mathrm{GF}(q)^d$ within distance $\gamma'd/2$ of x_{E_v} and whose check bits agree with y_{E_v}; if so, set x_{E_v} to z.

To analyze the algorithm, by linearity it suffices to consider the case when the correct message is the all-zeroes string (which also implies that all check symbols equal 0). Let $X = \{e : x_e \neq 0\}$ be the set of edges whose symbols are in error in the original received word x. For $i \geq 1$, let $Y^{(i)}$ (resp. $Z^{(i)}$) be the set of edges in error, i.e. edges e so that $x_e \neq 0$, after the left wing (resp. right wing) of the i'th round of decoding (we use the convention $Y^{(0)} = Z^{(0)} = X$). Define the set $A^{(i)}$ and $B^{(i)}$ for $i \geq 1$ as follows:

- $A^{(i)} = \{v \in A : E_v \cap Y^{(i)} \neq \emptyset\}$

- $B^{(i)} = \{v \in B : E_v \cap Z^{(i)} \neq \emptyset\}$

Now comes the crucial part of the analysis. Let $i \geq 1$ be fixed. For each $v \in A^{(i)}$ (i.e., vertices on the left which are incident to some uncorrected edge after the left wing decoding of the i'th round), we have $|E_v \cap Z^{(i-1)}| \geq \gamma'd/2$, as otherwise the left wing decoding of the i'th round would have corrected the fewer than $\gamma'd/2$ errors that remained in the edges of E_v. We also have, for the same reason, $|E_v \cap Y^{(i)}| \geq \gamma'd/2$ for every $v \in B^{(i)}$.

Our goal is to now prove that the size of the $A^{(i)}$'s and $B^{(i)}$'s decreases geometrically, which will imply that the algorithm converges in $O(\log n)$ rounds. Note that this immediately implies only an $O(n \log n)$ complexity decoding algorithm, but not a linear upper bound on the decoding time, since each round itself appears to require linear runtime. However, there is a linear-time implementation of the algorithm by carefully considering only "relevant" subsets of A, B which decrease in size geometrically when implementing the successive decoding rounds. We omit the details here and point the reader, for example, to [3, Sec. V], where explicit details on this aspect appear.

Now, consider the subgraph of H induced by the edges in $Y^{(i)}$. By definition, each such edge must be incident upon a vertex in $A^{(i)}$. Furthermore, every vertex in $B^{(i)}$ is incident upon at least $\gamma'd/2$ edges of $Y^{(i)}$. Applying Lemma 2 stated at the end of this section to this situation (with the choice $S = A^{(i)}$, $T = B^{(i)}$ and $Y = Y^{(i)}$), the expansion property of the graph H implies that $B^{(i)}$ has to be small provided $A^{(i)}$ is small. Specifically, $|B^{(i)}| \leq \zeta|A^{(i)}|$ for some $\zeta < 1$, provided $|A^{(i)}| \leq \rho n(\frac{\gamma'}{4} - \frac{2}{\sqrt{d}})$ for some $\rho < 1$. This condition will be satisfied provided $d \geq 64/\gamma'^2$ and $|A^{(i)}| \leq \gamma'n/16$. By the same argument, we will also have $|A^{(i+1)}| \leq \zeta|B^{(i)}|$ for $i \geq 1$. Hence, we would have proved the geometrically decreasing property, provided we can get an upper bound of $\gamma'n/16$ on $|A^{(1)}|$ to start with.

By definition each vertex of $A^{(1)}$ is adjacent to at least $\gamma'd/2$ erroneous edges, and hence we have $|X| \geq |A^{(1)}|\gamma'd/2$. Also, by hypothesis there are at most $\beta m = \beta nd$ errors, and so $|A^{(1)}| \leq \frac{2\beta n}{\gamma'}$. Therefore, if $\beta \leq \gamma'^2/32$, then $|A^{(1)}| \leq \gamma'n/16$ as desired. $\qquad \square$ (Lemma 1)

Lemma 2 ([26]) *Let $\rho < 1$ be arbitrary. Let $H = (A, B, E)$ be a d-regular bipartite expander with n vertices on each side and whose adjacency matrix has second largest eigenvalue $\lambda \leq d/3$. Let S be a subset of vertices of A such that $|S| \leq \rho n\left(\frac{\alpha}{2} - \frac{\lambda}{d}\right)$. Let T be a subset of vertices of B and suppose that there exists a set $Y \subseteq E$ of edges such that:*

(a) every edge in Y has one of its endpoints in S, and

(b) every vertex in T is incident to at least αd edges of Y.

Then, $|T| \leq \frac{1}{2-\rho}|S|$.

2.3 Linear-time error-correcting codes meeting the Singleton bound

We now use the codes from the previous section as the "left code" in our general construction scheme to obtain linear time encodable/decodable codes whose rate vs. error-correcting trade-off approaches the Singleton bound (we call such codes *near-MDS* codes). Below we state a more general result that handles both errors and erasures. This will help us deduce the result for binary codes in the next section very easily, since the GMD algorithm for concatenated codes that we will employ requires an errors-and-erasures decoding algorithm for the outer code.

Theorem 3 *For every* r, $0 < r < 1$, *and all sufficiently small* $\varepsilon > 0$, *there exists an* explicitly *specified family of* $\mathrm{GF}(2)$*-linear (also called additive)*[1] *codes of rate* r *and relative distance at least* $(1 - r - \varepsilon)$ *over an alphabet of size* $2^{O(\varepsilon^{-4} r^{-1} \log(1/\varepsilon))}$ *such that codes from the family can be encoded in linear time and can also be (uniquely) decoded in linear time from a fraction* e *of errors and* s *of erasures provided* $2e + s \leq (1 - r - \varepsilon)$.

Proof: We will use the construction outlined in Section 2.1 with left code being the code from Lemma 1 for the choice $\gamma = \varepsilon/4$. Let x be a message of length m over $\mathrm{GF}(q)$ for some constant q (jumping ahead, q will be a power of two large enough for the left code and the Reed-Solomon code $\tilde{C}$ to exist). The message is first encoded by C to give a string $y = C(x)$ of length $n' = (1 + \varepsilon/4)m$ over $\mathrm{GF}(q)$. We assume, by Lemma 1, that C can correct $\beta n'$ errors in linear time for $\beta = O(\varepsilon^2)$. The symbols of y will be broken up into $n = n'/b$ blocks consisting of b symbols each for a block size $b = \Theta(1/\varepsilon^4)$. Each of these n blocks will undergo encoding by a Reed-Solomon code $\tilde{C}$ over $\mathrm{GF}(q)$ of dimension b and rate $r' = r(1 + \varepsilon/4)$, to give n blocks $B_1, \ldots, B_n$ each consisting of $\Delta = b/r'$ symbols over $\mathrm{GF}(q)$ (if we pick $q = \Omega(r^{-1}\varepsilon^{-4}) \geq \Delta$, both the left code as well as the Reed-Solomon code will exist over an alphabet of size q).

Let $G = (A, B, E)$ be a Δ-regular bipartite expander with n vertices on each side with the following property:

(*) For every subset $X \subset A$ with $|X| \geq \beta n/2$ and every $Y \subseteq B$, we have $\left| \frac{|E(X:Y)|}{|X|\Delta} - \frac{|Y|}{|B|} \right| \leq \varepsilon/4$.

One can show that Ramanujan graphs, namely graphs whose second largest eigenvalue satisfies $\lambda = O(\sqrt{\Delta})$, of degree $\Delta = O(1/\beta\varepsilon^2) = O(1/\varepsilon^4)$, give bipartite graphs with the above property (see for example [12] or [10, Chap. 11] for a proof). Explicit constructions of Ramanujan graphs are known [16] and since $C, \tilde{C}$ are explicitly specified as well, our overall construction is explicit.[2] The symbols of the i'th block will be redistributed to the neighbors of the i'th vertex on the left side of G (the j'th symbol going to the j'th neighbor of the vertex, for $1 \leq j \leq \Delta$, as per some arbitrary ordering of the neighbors of each vertex). This gives, for each vertex on the right side, a collection of Δ $\mathrm{GF}(q)$-symbols obtained from its Δ neighbors on the left, which, equivalently, can be viewed as a single symbol over $\mathrm{GF}(q^\Delta)$. The string (of length n) consisting of these symbols forms the encoding of x by our overall code over $\mathrm{GF}(q^\Delta)$, call it C^*. (Taking another quick look at Figure 1 before reading on might be useful to the reader.)

---

[1] A code C over a field of characteristic 2 is said to $\mathrm{GF}(2)$-linear or additive if $x + y \in C$ whenever both $x \in C$ and $y \in C$. The codes we construct have this property, but they are not in general linear over the larger field.

[2] Here we assume that parameters have been so picked that there is an explicit Ramanujan graph, eg. the construction of [16], with exactly n vertices. Since there is a lot of flexibility in the choice of parameters of the left code C and the Reed-Solomon code $\tilde{C}$, and since the sequences of vertex sizes of known explicit constructions of Ramanujan graphs form a dense sequence, this can be easily ensured. For sake of simplicity, we ignore this issue and simply assume that expanders with exactly the required number of vertices exist.

RATE AND ALPHABET SIZE. This gives a code over an alphabet of size $q^\Delta = 2^{O(b \lg q/r')} = 2^{O(\varepsilon^{-4} r^{-1} \log(1/\varepsilon))}$ and which has rate $\frac{m/\Delta}{n} = \frac{mr'}{bn} = \frac{mr'}{n'} = r$ (since $n' = (1 + \varepsilon/4)m$ and $r' = r(1 + \varepsilon/4)$). It is also clear that C^* has a linear time encoding algorithm.

DECODING COMPLEXITY. Using the Property (*) of G, it is also easy to show that the relative distance of C^* is at least $(1 - r - \varepsilon/2)$. In fact, we next prove that C^* can be uniquely decoded from a fraction e of errors and s of erasures provided $2e + s \leq (1 - r - \varepsilon)$.

Let z be a received word for C^* with a fraction s of erasures and a fraction e of errors, where $2e + s \leq (1 - r - \varepsilon)$. Since the relative distance of C^* is greater than $(1 - r - \varepsilon)$, there is a unique message x that is solution to the decoding problem. Let S be the set of erasures in the received word z, and let F be the set of errors (i.e., the positions where $C^*(x)$ and z differ). We have $|S| = sn$ and $|F| = en$.

Given the received word z, the decoding algorithm proceeds as follows. In the first step, the word z is used to compute certain "received words" z_i, $1 \leq i \leq n$, for the n encodings by $\tilde{C}$ (corresponding to the n blocks into which a codeword of C is broken into). This is done as follows. For each i, j, $1 \leq i \leq n$ and $1 \leq j \leq \Delta$, if the j'th neighbor of the i'th node of A has an unerased symbol, say $\zeta \in \mathrm{GF}(q^\Delta)$, then the j'th symbol of z_i is set to the symbol in the appropriate coordinate of ζ (namely, the coordinate which received that symbol through the expander). If the j'th neighbor of the i'th node of A has an erased symbol, then we declare an erasure at the j'th position of z_i.

For each i, $1 \leq i \leq n$, let z_i be the received word thus obtained for the encoding of i'th block. Let s_i be the fraction of positions in z_i which are erased, and let e_i be the fraction of positions in z_i which are set to a wrong symbol. With the z_i's computed, the algorithm continues as follows. For each i, we run a unique error-erasure decoding algorithm for the Reed-Solomon code $\tilde{C}$ with received word z_i. If it succeeds in decoding, we let $y_i \in \mathrm{GF}(q)^b$ be the message it outputs, otherwise we let y_i be an arbitrary string in $\mathrm{GF}(q)^b$. Finally, the decoding is completed by running the linear time unique decoding algorithm for C on the received word $y = \langle y_1, y_2, \ldots, y_n \rangle$, and outputting whatever message x it outputs.

It is clear that the algorithm runs in linear time. We now prove the correctness of this procedure. We claim that it suffices to prove that the received words z_i (obtained from the first stage of the decoding that uses the expander) satisfy $2e_i + s_i < (1 - r - \varepsilon/4)$ for at least $(1 - \beta)n$ values of i. Indeed, for any such i, the Reed-Solomon decoder will succeed in finding the correct block y_i (as the relative distance of each Reed-Solomon code is at least $(1 - r(1 + \varepsilon/4)) \geq 1 - r - \varepsilon/4$). Hence the received word y passed to the decoding algorithm for C will agree with $C(x)$ entirely on a fraction $(1 - \beta)$ of the blocks, or in other words y and $C(x)$ will differ in at most $\beta n'$ positions. Since the assumed decoding algorithm for C can correct up to a fraction β of errors, we will correctly find and output the message x.

It remains to prove that $2e_i + s_i < (1 - r - \varepsilon/4)$ for all but βn values of i. Define $X' \subset A$ to be the set of nodes which have at least a fraction $(s + \varepsilon/4)$ of neighbors in the set S (the set of erasures in the received word z). Also define $X'' \subset A$ to be the nodes which have at least a fraction $(e + \varepsilon/4)$ of neighbors in F (the set of erroneous positions in z). It easily follows from the Property (*) of the expander G that $|X'|, |X''| \leq \beta n/2$.

Now consider any node $i \in A \setminus (X' \cup X'')$. It has less than a fraction $(e + \varepsilon/4)$ of neighbors in F. These correspond to the errors in the received word z_i, and hence we have

$$e_i < e + \varepsilon/4 \quad \text{for every } i \in A \setminus (X' \cup X'') . \tag{1}$$

A node $i \in A \setminus (X' \cup X'')$ also has less than a fraction $(s + \varepsilon/4)$ of neighbors in S. These correspond to the erasures in the received word z_i, and hence we have

$$s_i < s + \varepsilon/4 \quad \text{for every } i \in A \setminus (X' \cup X'') . \tag{2}$$

Since $2e + s \leq (1 - r - \varepsilon)$ by hypothesis, we have, combining (1) and (2) that $2e_i + s_i < (1 - r - \varepsilon/4)$, for each $i \in A \setminus (X' \cup X'')$. Since $|X'|, |X''| \leq \beta n/2$, we have proved that the condition $2e_i + s_i < (1 - r - \varepsilon/4)$ holds for all but a fraction β of i's in the range $1 \leq i \leq n$. This completes the proof of correctness of the decoding algorithm. $\qquad\square$

3 Linear time encodable/decodable binary codes

3.1 Binary linear-time codes meeting the Zyablov bound

We now construct binary codes which have excellent rate vs. error-correction trade-off and further have linear time encoding and decoding algorithms. Our codes meet the *Zyablov* bound which is the best trade-off known with reasonable construction complexity (and the best known for concatenated codes).

Our code constructions are obtained by concatenating the near-MDS codes from Theorem 3 with a binary inner code which meets the Gilbert-Varshamov bound. Such a code can be constructed by picking a linear code at random and checking that it has the necessary distance property, or a deterministic construction can be obtained by searching for the inner code (since it is of constant size, this takes only $O(1)$ time). Linear time encoding is clear, and for decoding we use Generalized Minimum Distance (GMD) decoding [7], which decodes a concatenated code up to the "product bound" (i.e., half the product of the designed distances of the outer and inner codes) by running several instances of the errors-and-erasures algorithm for the outer near-MDS code. The number of such runs needed is bounded from above by half the distance of the inner code and therefore by a fixed constant as the inner code is of constant size. Since each run takes linear time by Theorem 3, the overall decoding time is linear. The statement we need about GMD decoding is formally stated for completeness below.

Lemma 4 *Let C_{out} be an $(N, K)_Q$ code where $Q = q^k$ and let C_{in} be an $(n, k)_q$ code with minimum distance at least d. Let $\mathbf{C}$ be the $(Nn, Kk)_q$ code obtained by concatenating C_{out} with C_{in}. Assume that there exists an algorithm running in time T_{in} to uniquely decode C_{in} up to less than $d/2$ errors. Assume also the existence of an algorithm running in time T_{out} that uniquely decodes C_{out} from S erasures and E errors as long as $2E + S < \tilde{D}$ for some $\tilde{D} \leq \operatorname{dist}(C_{\text{out}})$. Then there exists an algorithm $\mathcal{A}$ running in $O(N T_{\text{in}} + d T_{\text{out}})$ time that uniquely decodes $\mathbf{C}$ from any pattern of less than $\frac{d\tilde{D}}{2}$ errors.*

Using the above result for concatenated codes with outer codes from Theorem 3 and inner code being one of the appropriate dimension that meets the Gilbert-Varshamov bound, we get our result for linear-time binary codes below. We omit the full details which are standard and easy to fill in. The reader may find all the details about how GMD decoding together with Theorem 3 will give the result below, by referring to similar results in [10, Chap. 11].

Theorem 5 *For every $\varepsilon > 0$ and for any code rate $0 < R < 1$, there exists a family of* binary linear *concatenated codes of rate R which can be encoded in linear time and can be decoded in linear time from up to a fraction e of errors, where*

$$e \geq \max_{R < r < 1} \frac{(1 - r - \varepsilon) H^{-1}(1 - R/r)}{2} \tag{3}$$

($H^{-1}(y)$ is defined to be the unique x in the range $0 \leq x \leq 1/2$ that satisfies $H(x) = y$). Every code in the family is explicitly specified given a constant sized binary linear code which can be constructed in probabilistic $O(\varepsilon^{-4} \log(1/\varepsilon))$ or deterministic $2^{O(\varepsilon^{-4} \log(1/\varepsilon))}$ time.

The bound of Equation (3) is half the Zyablov bound [27], and thus these codes match the best error-correction performance known for constructive binary concatenated codes. We remark that the first explicit construction of codes meeting the Zyablov bound for all rates was due to Shen [19]. These were based on certain algebraic-geometric codes as outer codes and the encoding and decoding times were at least quadratic in the block length.

3.2 Linear-time codes beyond the Zyablov bound using multilevel concatenation

By using the multilevel generalizations of concatenated codes (see [6] for a detailed discussion) we can construct linear-time binary codes which get arbitrarily close to the so-called *Blokh-Zyablov* bound [5] and thus lie above the Zyablov bound. Specifically, for any $\varepsilon > 0$, we can correct a fraction e of errors with rate

$$R \geq 1 - \varepsilon - H(2e) - 2e \int_0^{1-H(2e)} \frac{dx}{H^{-1}(1-x)} \ . \tag{4}$$

For a comparative plot of the Blokh-Zyablov and Zyablov bounds, we refer the reader to the survey paper by Dumer [6, Section 4.1]. We also refer the reader to [6] for detailed information on multilevel concatenated codes, and simply sketch the basic idea here. For an integer $s \geq 1$, an s-*level concatenated* binary code, say $\mathbf{C}^{(s)}$, is obtained by combining s "outer" codes $C_1, C_2, \ldots, C_s$ of the same block length, say N, over large alphabets of size, say, $2^{m_1}, 2^{m_2}, \ldots, 2^{m_s}$ respectively, with a suitable "inner" binary code. The inner code is of dimension $m_1 + m_2 + \cdots + m_s$. Given messages $x_1, x_2, \ldots, x_s$ for the s outer codes, the encoding as per the s-level generalized concatenated code proceeds by first encoding each x_i as per C_i (note that since the C_i's encode different messages, the rate "adds up"). Then for every j, $1 \leq j \leq N$, the collection of the j'th symbols of $C_i(x_i)$ for $1 \leq i \leq s$, which can be viewed as a binary string of length $m_1 + m_2 + \cdots + m_s$, is encoded by the inner code. Note that for $s = 1$ this reduces to the usual definition of code concatenation.

By using as outer codes the codes from Theorem 3 of suitably varying rates and an inner code based on a sequence of nested binary linear codes that each meets the Gilbert-Varshamov bound (see Lemma 7 for a formal statement), and arguing as in Section 4.2 of [6], we can prove the following:

Theorem 6 *For every integer $s \geq 1$, every r, $0 < r < 1$, any $\delta \leq H^{-1}(1 - r)$, the following holds for every $\varepsilon > 0$: There exists an infinite family of s-level concatenated (binary linear) codes of relative distance $\geq \delta$ and rate*

$$R = r(1 - \varepsilon) - (r/s) \sum_{i=0}^{s-1} \frac{\delta}{H^{-1}(1 - r + ri/s)} \ ,$$

which can be encoded as well as decoded up to a fraction $\delta/2$ of errors in linear time. Every code in the family is explicitly specified given a constant sized binary linear code which can be constructed in deterministic $2^{O(\varepsilon^{-4} \log(1/\varepsilon))}$ time (the constants in the big-Oh notation depend on s, r, δ).

Remark: We note here that the codes guaranteed by the above theorem, even without the statement about encoding and decoding complexity, give the first binary codes that lie above the Zyablov bound and that have a reasonable asymptotic construction complexity. The best known trade-off between rate and distance for constructive binary codes is given by concatenated codes with outer

codes being certain algebraic-geometric codes of high (at least cubic in block length) construction complexity. As far as codes that meet the Blokh-Zyablov bound are concerned, prior constructions of s-level concatenated codes which approached this bound were based on outer Reed-Solomon codes, and hence picking a nested sequence of inner codes that lie on the Gilbert-Varshamov bound (as guaranteed by Lemma 7 below) led to construction complexity of about $O(N^s)$. In contrast the codes from Theorem 6 are almost explicit and only involve a constant (independent of the block length) amount of search for the inner codes.

From Theorem 6 it follows that one can construct a family of linear time encodable/decodable codes that can correct a fraction e of errors with rate

$$R = \max_{r \le 1 - H(2e)} r(1 - \varepsilon) - (r/s) \sum_{i=0}^{s-1} \frac{2e}{H^{-1}(1 - r + ri/s)} \ , \tag{5}$$

for any $\varepsilon > 0$. Note that for concatenated codes ($s = 1$) this is just the Zyablov bound (3) in another guise. As the order s grows, the optimal value r in (5) tends to the maximal possible value $1 - H(\delta)$, and the sum in (5) is converted into an integral. This leads to the Blokh-Zyablov bound (4) discussed above. We now sketch the proof of Theorem 6. We first state a useful lemma that guarantees the necessary inner code for the multilevel concatenated scheme.

Lemma 7 *([Lemma 4.10] in [6])   For every r, $0 < r < 1$ and every integer $s \ge 1$, there is a family of binary linear codes of rate r and relative distance at least $H^{-1}(1-r)$ with the following property. A code C of block length n in the family can be constructed with complexity of order 2^n and for $0 \le p \le s - 1$, given two distinct messages which agree in the first prn symbols, their encoding by C differs in at least $H^{-1}(1 - r(1 - p/s))n$ positions.*

Proof of Theorem 6: The proof follows along the lines of Theorem 4.11 in [6] — instead of the Reed-Solomon codes used in [6] we will use the near-MDS codes of Theorem 3 as the outer codes in a suitable multilevel concatenation scheme. Since multilevel concatenated codes are not as well known as the usual concatenated codes, we give further details on the code construction below.

Given ε, r, s, define

$$r_i = r\Big(1 - \frac{i-1}{s}\Big); \qquad R_i = 1 - \frac{\delta}{H^{-1}(1 - r_i)} - \varepsilon$$

for $i = 1, 2, \ldots, s$. Pick $Q = O(\varepsilon^{-4} \log(1/\varepsilon))$ (where the constant in the big-Oh notation can depend on s, r, δ) large enough so that Theorem 3 guarantees the existence of linear-time codes C_i of rate R_i of relative distance at least $(1 - R_i - \varepsilon)$ over an alphabet of size 2^Q, for $i = 1, 2, \ldots, s$. Let N be the common block length of the C_i's. These codes will be used as the outer codes in constructing our multilevel concatenated code C^*.

The inner code C_{in} of the s-level concatenated code is as guaranteed by Lemma 7. Given a binary string $\mathbf{x}$ of length Qs (which is the length of the string obtained by the collection of the j'th symbols of the codewords of $C_1, C_2, \ldots, C_s$ for any j, $1 \le j \le N$), it is encoded by C_{in} into a string of length Qs/r. Note that C^* is linear-time encodable since each of the outer codes C_i are linear-time encodable and the inner code is of constant size. The rate of C^* equals

$$\frac{r}{s} \sum_{i=1}^{s} R_i = r(1 - \varepsilon) - \frac{r}{s} \sum_{i=1}^{s} \frac{\delta}{H^{-1}(1 - r_i)} \ ,$$

11

as claimed in the theorem. It remains to show that C^* can be decoded in linear time from up to a fraction $\delta/2$ of errors.

The relative distance of C_{in} is at least $H^{-1}(1-r)$. Furthermore, as per Lemma 7, given two distinct strings $\mathbf{x}_1$ and $\mathbf{x}_2$ which have a common initial block of $Q(i-1)$ bits (for $1 \le i \le s$), their encoding by C_{in} differ in at least a fraction $H^{-1}(1-r_i)$ of positions. By a standard bound on distance of multilevel concatenated codes (see [6, Section 2.2]), it follows that the relative distance of C^* is at least

$$\min_{1 \le i \le s} (1 - R_i - \varepsilon)H^{-1}(1-r_i) \ge \delta .$$

A natural generalization of Generalized Minimum Distance decoding called cascaded decoding of multilevel concatenations (see [6, Theorem 5.7]) enables decoding of C^* up to a fraction $\delta/2$ of errors. As with GMD decoding, this involves running an errors-and-erasures decoding algorithm for each C_i at most $H^{-1}(1-r_i)Qs/r$ times (this number is the designed distance of the inner code once the first $Q(i-1)$ bits are fixed which will be the case once the first $(i-1)$ outer codes are decoded). Since each C_i has a linear time decoding algorithm to handle a combination of errors and erasures up to the distance $(1 - R_i - \varepsilon)$, and s, Q, r are constants, the overall runtime of the cascaded decoding algorithm is linear in the block length (with leading constants that depend on s, r, ε). This completes the proof of Theorem 6. $\qquad\square$

3.3 Achieving Forney's error exponent with linear complexity

We conclude the paper by noting that our results can also be used to achieve Forney's error exponent for concatenated codes for the binary symmetric channel together with *linear time encoding/decoding algorithms*. Our formal result is stated in Theorem 8 below. The best previous complexity bound that achieved this error exponent was quadratic in the block length, dating back all the way to Forney's original work [8]. In fact, [8] was the first work to give binary codes together with a polynomial time decoding algorithm with exponentially small error probability.

Theorem 8 *For each $R < 1$ and every $\varepsilon > 0$, there is a polynomial-time constructible family of codes of rate R which is linear time encodable and also decodable in linear time with error probability $2^{-Nf(R,p)}$, where N is the block length, p is the crossover probability of the binary symmetric channel, and*

$$f(R,p) = \max_{R \le R_0 < 1 - H(p)} E_L(R_0, p)(1 - R/R_0) - \varepsilon , \tag{6}$$

with $E_L(R_0, p)$ being the well-known lower bound on "random coding exponent" [9] which we spell out for completeness below:

$$E_L(R_0, p) = \begin{cases} -H^{-1}(1-R_0)\log\sqrt{4p(1-p)} & \text{if } 0 \le R_0 \le R_x \\ 1 - \log(1+\sqrt{4p(1-p)}) - R_0 & \text{if } R_x \le R_0 \le R_{\text{crit}} \\ T(H^{-1}(1-R_0), p) + R_0 - 1 & \text{if } R_{\text{crit}} \le R \le 1 - H(p) \end{cases}$$

where $T(x,y) = -x \log y - (1-x)\log(1-y)$, $R_x = 1 - H\left(\frac{\sqrt{4p(1-p)}}{1+\sqrt{4p(1-p)}}\right)$, and $R_{\text{crit}} = 1 - H\left(\frac{\sqrt{p}}{\sqrt{p}+\sqrt{1-p}}\right)$.

We point the reader to [15] for detailed pointers to the above bound on $E_L(R_0, p)$ as well as some of the more recent work on error exponents for binary symmetric channel.

The above result follows similarly to the proof of Theorem 5, namely by concatenating the near-MDS codes (of rate R/R_0) from Theorem 3 with suitable inner codes of rate R_0 whose error

exponent equals the random coding exponent $E(R_0, p)$. The decoding algorithm performs a brute-force maximum likelihood decoding of the inner codes, followed by a GMD decoding of the outer code. Forney's analysis of GMD decoding [8, Sec. 4.2] shows that this achieves the error exponent $f(R, p)$ above. We note that our inner codes are of fixed length where as they were of growing length in Forney's work (since the outer codes used therein where Reed-Solomon codes), and this necessitates our worsening the error exponent by ε that is an arbitrarily small but fixed constant (independent of the block length N of the code). Likewise, our outer codes are only near-MDS as opposed to Reed-Solomon codes which are MDS, and this again costs us an ε in the error exponent. Since each of these ε losses can be made arbitrarily small, we are able to combine them and state a bound like (6). Linear time encoding follows since the outer code can be encoded in linear time and the inner codes are of size that is a large enough constant. Since we have a linear-time errors and erasure decoder for our outer codes, the overall decoding time is linear in the block length.

Acknowledgments

References

[1] Noga Alon, Jehoshua Bruck, Joseph Naor, Moni Naor, and Ronny Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Transactions on Information Theory*, 38:509–516, 1992.

[2] Noga Alon, Jeff Edmonds and Michael Luby. Linear time erasure codes with nearly optimal recovery. *Proceedings of FOCS'95*.

[3] Alexander Barg and Gillés Zémor. Error exponents of expander codes. *IEEE Transactions on Information Theory*, 48(6):1725-1729, June 2002.

[4] Alexander Barg and Gillés Zémor. Concatenated codes: serial and parallel. Manuscript, 2003.

[5] E. L. Blokh and V. V. Zyablov. Linear concatenated codes. *Nauka, Moscow*, 1982 (in Russian).

[6] Ilya I. Dumer. Concatenated codes and their multilevel generalizations. In V. S. Pless and W. C. Huffman, editors, *Handbook of Coding Theory*, volume 2, pages 1911–1988. North Holland, 1998.

[7] G. David Forney. Generalized Minimum Distance decoding. *IEEE Transactions on Information Theory*, 12:125–131, 1966.

[8] G. David Forney. *Concatenated Codes*, MIT Press, Cambridge, MA, 1966.

[9] Robert G. Gallager. *Low-density parity-check codes*. MIT Press, Cambridge, MA, 1963.

[10] Venkatesan Guruswami. *List Decoding of Error-Correcting Codes*. Ph.D thesis, Massachusetts Institute of Technology, August 2001.

[11] Venkatesan Guruswami. List decoding from erasures: Bounds and code constructions. *Proceedings of the 21st Foundations of Software Technology and Theoretical Computer Science*, Bangalore, India, pages 195-206, December 2001.

[12] Venkatesan Guruswami and Piotr Indyk. Expander-based constructions of efficiently decodable codes. *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science*, Las Vegas, NV, pages 658-667, October 2001.

[13] Venkatesan Guruswami and Piotr Indyk. Near-optimal linear time codes for unique decoding and new list decodable codes over smaller alphabets. *Proceedings of STOC'02*, Montreal, Canada, May 2002.

[14] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45:1757–1767, 1999.

[15] Simon Litsyn. New upper bounds on error exponents. *IEEE Transactions on Information Theory*, 45(2):385–398, March 1999.

[16] Alex Lubotzky, R. Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[17] Rasmus R. Nielsen and Tom Høholdt. Decoding Reed-Solomon codes beyond half the minimum distance. *Coding Theory, Cryptography and Related areas, (eds. Buchmann, Hoeholdt, Stichtenoth and H. tapia-Recillas)*, pages 221–236, 1999.

[18] Ronny Roth and Gitit Ruckenstein. Efficient decoding of Reed-Solomon codes beyond half the minimum distance. *IEEE Transactions on Information Theory*, 46(1):246–257, January 2000.

[19] Ba-Zhong Shen. A Justesen construction of binary concatenated codes that asymptotically meet the Zyablov bound for low rate. *IEEE Transactions on Information Theory*, 39:239–242, 1993.

[20] M. Amin Shokrollahi and Hal Wasserman. List decoding of algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45(2):432–437, 1999.

[21] Kenneth Shum, Ilia Aleshnikov, P. Vijay Kumar, Henning Stichtenoth, and Vinay Deolalikar. A low-complexity algorithm for the construction of algebraic-geomteric codes better than the Gilbert-Varshamov bound. *IEEE Transactions on Information Theory*, 47:2225-2241, 2001.

[22] Michael Sipser and Daniel Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.

[23] Daniel Spielman. *Computationally Efficient Error-Correcting Codes and Holographic Proofs*. Ph.D thesis, Massachusetts Institute of Technology, June 1995.

[24] Daniel Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, 42(6):1723–1732, 1996.

[25] Madhu Sudan. Decoding of Reed-Solomon codes beyond the error-correction bound. *Journal of Complexity*, 13(1):180–193, 1997.

[26] Gillés Zémor. On expander codes. *IEEE Transactions on Information Theory*, 47(2):835–837, 2001.

[27] Victor V. Zyablov. An estimate of the complexity of constructing binary linear cascaded codes. *Problemy Peridachi Informatsii*, 15(2):58-70, 1971.