Linear-time Codes to Correct a Maximum Possible Fraction of Errors

Venkatesan Guruswami* Piotr Indyk†

October 2, 2001

Abstract

We construct linear-time encodable and decodable *binary* codes of positive rate (in fact, rate $\Omega(\varepsilon^4)$) that can correct up to $(1/4-\varepsilon)$ fraction errors, for an arbitrarily small positive constant ε. Note that this is the best error-correction one can hope for using unique decoding of binary codes. This significantly improves the fraction of errors corrected by the earlier linear-time codes of Spielman [11] and the linear-time decodable codes of Sipser and Spielman [10] and Zémor [13].

1 Introduction

In this paper, we are interested in the construction of codes of good rate and large relative distance which can be encoded and decoded up to close to half the minimum distance in linear time. In other words, our goal is to construct *linear-time* codes that can be *uniquely* decoded up to the "maximum" possible radius one can hope for.

For unique decoding, the maximum number of errors that can be corrected is limited by half the minimum distance of the code. In particular, it is never possible to correct more than a fraction $1/2$ of errors. In this paper, we are interested in both binary codes and codes over large alphabets. Since binary code families of positive rate have relative distance less than $1/2$,the half-the-minimum-distance barrier implies that the maximum possible fraction of errors that can be uniquely decoded is $(1/4-\varepsilon)$ for binary codes. For codes over large alphabets, the maximum unique decoding radius is $(1/2-\varepsilon)$ (this requires an alphabet size of $\Omega(1/\varepsilon)$, though).

Therefore, we are interested in codes over large alphabets that are uniquely decodable up to a fraction $(1/2-\varepsilon)$ of errors, and in binary codes that are uniquely decodable up to a fraction $(1/4-\varepsilon)$ of errors. These represent the "high-noise" (or "high-distance") regime for unique decoding. For such code constructions, our goal is to achieve good rate together with highly efficient encoding and decoding algorithms.

The best known explicit constructions of codes with such a large relative distance achieve a rate of $\Omega(\varepsilon)$ and $\Omega(\varepsilon^3)$ respectively. However, such codes either use algebraic-geometric codes [9] and hence suffer from complicated decoding procedures (in particular the decoding time is at least cubic in the blocklength), or as in the constructions of [1], are explicitly constructible but lack a polynomial time algorithm to decode up to half the minimum distance (or for that matter to decode up to any positive constant, no matter how small, fraction of errors).

*University of California at Berkeley, Computer Science Division, Berkeley, CA 94720. Work done while at MIT. Email: venkat@lcs.mit.edu

†MIT Laboratory for Computer Science, 200 Technology Square, Cambridge, MA 02139. Email: indyk@theory.lcs.mit.edu

In this paper, we present novel constructions of large distance codes of close to the above "best known" rates, together with *linear time* unique decoding algorithms, as well as linear time encoding algorithms. Specifically, we construct codes over a constant-sized alphabet that have $\Omega(\varepsilon^2)$ rate and which are uniquely decodable from a fraction $(1/2-\varepsilon)$ of errors in linear time and which are further encodable in linear time. We also construct similar linear-time binary codes that can correct a fraction $(1/4-\varepsilon)$ of errors and have $\Omega(\varepsilon^4)$ rate. Our constructions use the earlier linear-time codes of Spielman [11] as a black-box, and given this our codes are extremely simple to describe, and so are the decoding algorithms.

The parameters achieved by our code constructions are tabulated in Figure 1. The binary codes are obtained by a "standard" concatenation of our large alphabet codes.

No	Alphabet	Decoding radius	Rate	Encoding time	Decoding time	Construction time (probabilistic)
A	$2^{1/\varepsilon^2}$	$1/2-\varepsilon$	ε^2	n/ε^2	n/ε^2	explicit
B	2	$1/4-\varepsilon$	ε^4	$n+2^{1/\varepsilon^2}$	$n/\varepsilon^2+2^{1/\varepsilon^4}$	$1/\varepsilon^4$

Figure 1: The parameters of our linear-time codes for unique decoding. n stands for the length of the code. For readability, the $O(\cdot)$ and $\Omega(\cdot)$ notation have been omitted, and $\log(1/\varepsilon)$ factors have been omitted in the encoding and decoding times. The decoding radius shows the fraction of errors which the unique decoding algorithms can correct.

Elaboration on above results. The rate of the above codes is an ε factor worse compared to the best known rate for constructive codes with the necessary relative distance. However, these codes are simple to describe, and can correct a fraction of errors arbitrarily close to the maximum possible (i.e., $1/4$ for binary codes and $1/2$ for large alphabet codes) in *linear* time. In fact, by decoding up to the maximum possible radius, these codes exhibit a *dramatic* improvement in error-correction capability over that of the only prior linear-time binary codes (due to Spielman [11, 12]), or for that matter over that of the linear-time decodable binary codes of Sipser and Spielman [10], Zémor [13], and Barg and Zémor [3]. To make a concrete comparison, the largest fraction of errors corrected by the linear-time codes in [11] (at least the bound directly deducible from the paper) seems to be about 4.6×10^{-7}. The linear-time decodable codes from [13] can correct about a fraction 3.1×10^{-3} of errors with positive rate. In comparison, we can correct a fraction $(1/4-\varepsilon)$ of errors for arbitrary $\varepsilon>0$.

Subsequent Improvements. We should remark that recently we were able to improve the rate of our constructions to $\Omega(\varepsilon)$ and $\Omega(\varepsilon^3)$ respectively for the large alphabet and binary codes. Note that these are the *asymptotically best known rates* for explicitly specified codes of relative distance $(1-\varepsilon)$ and binary codes of relative distance $(1/2-\varepsilon)$ respectively, and are we are able to achieve these rates *together with linear time encoding and decoding procedures.* These improvements will appear elsewhere — we just mention here that the basic construction philosophy of the improved codes is similar to the constructions used in this paper, though some of the details are more complicated.

A Remark on Construction Times. Our code construction (A) over a large alphabet is *explicit.*[1] Our binary codes (B) are obtained by concatenating the codes (A) with suitable binary codes. These

[1]We do not try to formally define the notion of a construction being "explicit". We refer to a construction as explicit only if it is "constructive" in the strictest sense of the term, namely if it is specifiable by a formula of an acceptable type.

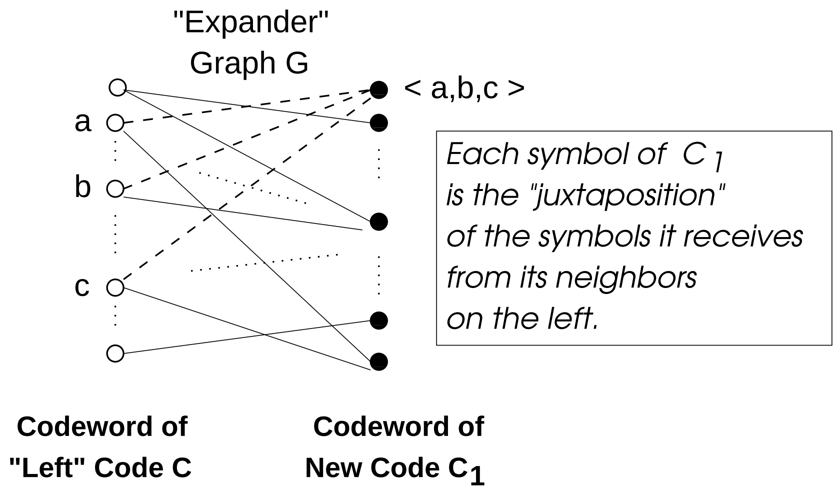

Figure 2: Basic structure of our code constructions. For binary codes, each symbol of C_1 is further concatenated with a good, constant-sized binary code.

binary codes can either be picked randomly or found deterministically by a constant amount of brute-force search. If one considers codes with only a constant amount of "non-constructivity" as still being "explicit", then our binary code construction is also explicit. Moreover, we can make the construction completely explicit at the expense of a (moderate) worsening of the rate, by picking some explicit binary code for the final concatenation instead of the "best" non-explicit ones. We point out here that the linear-time codes of Spielman [11] also require a constant amount of non-constructivity, which can eliminated at the expense of worsening the parameters (see the discussion in [11, Sec. 5, Remark 20]).

Comment on meaning of "linear time": When we claim that our encoding and decoding algorithms run in linear time, we must be careful to specify exactly what is meant, since the meaning of "linear time" depends on the model of sequential computation considered. We just remark that our codes use the linear-time codes due to Spielman [11] as a black-box, and use expanders on top of it to boost the number of errors tolerated by the code. Our algorithms run in linear time in *any* model under which the algorithms of Spielman run in linear time. In particular, they can be implemented to run in linear time on a RAM in the uniform cost model, or on a Pointer Machine (see [10, Sec. 2] for details and pointers on these models).

2 High-level structure of codes

The basic components of the constructions are: a "left" code (say, C), an expander-like graph G, and (in case of binary codes) an efficient "right" binary code (say C').

Given the above components, the codes are constructed as follows. For each codeword x of C, we construct a new codeword y by distributing the symbols of x from left to right according to the edges in G. The juxtaposition (or concatenation) of symbols "sent" to each right node of G forms a symbol of the codeword y of the final large alphabet code C_1. See Figure 2 for an illustration of the basic construction scheme. If we want to construct a binary code, we add a final level of concatenation with the binary code C'.

In our constructions, we take the left code C to be the explicit linear-time binary codes of Spielman [11] which can correct some constant fraction of errors. The properties of the expander graph G allows us to transform a heavily corrupted codeword y of C_1 to a much less corrupted codeword x of C. The latter can be decoded using the linear-time decoding algorithm for C. The encoding can also be accomplished in linear time using the linear-time encoder for C.

Our constructions are similar to those in [1] where expanders were used to "amplify" the distance and thus construct codes of large minimum distance. However, while the results of [1] use only the vertex-expansion properties of the graph G, here we use much stronger *isoperimetric* properties offered by expander graphs. We need these stronger properties because we are also interested in good decoding algorithms for our codes. Informally, the isoperimetric property just states that the fraction of edges between two sufficiently large subsets of vertices is approximately the product of their densities. The isoperimetric property of expanders is implied by the spectral definition of expanders in terms of the second largest eigenvalue of the adjacency matrix. It is this definition of "expanders" that we use in this paper. In the next section, we give the necessary background on expanders that will be necessary for our code constructions (A,B) (discussed in Sections 4 and 5 respectively).

3 Background on the necessary expanders

There are several ways in which expander graphs are defined in the literature. One possible definition just relates to the "vertex-expansion" of the graph, and under this definition a graph is an expander if the neighborhood of every large enough subset of vertices is large. For our application here we will also need some *isoperimetric* properties offered by expanders, and therefore we use a spectral definition of expanders based on the second largest eigenvalue of the normalized adjacency matrix. Under this definition a Δ-regular graph H on n vertices with adjacency matrix $\mathbf{A}$ is an expander if $\lambda(H) < 1$, where $\lambda(H) \stackrel{\text{def}}{=} \max\{\lambda_2, |\lambda_n|\}$ is defined to be the second largest eigenvalue in magnitude and $1 = \lambda_1 \geq \lambda_2 \geq \cdots \lambda_n \geq -1$ are the n eigenvalues of $\frac{1}{\Delta} \cdot \mathbf{A}$.[2]

The following result relating the second eigenvalue to vertex expansion is well-known and has appeared in many places (see, for example, Theorem 2.4 of [2, Chap. 9]).

Lemma 1 *Let $H = (V, E)$ be a Δ-regular graph with $n = |V|$ and $\lambda(H) = \lambda$, and let $T \subseteq V$ with $|T| = bn$. Let $t = |v \in V : N(v) \cap T = \emptyset\}|$ be the number of vertices of H that have* no *neighbors in T. Then*

$$t \leq \frac{\lambda^2(1-b)n}{b} \; . \tag{1}$$

The above lemma applies to a general graph while we are interested in bipartite graphs. But this is easily fixed. One can define a $n \times n$ bipartite graph $G = (A, B, E')$ from the above graph H by letting A, B to be copies of V and connecting a vertex $a \in A$ with $b \in B$ iff the corresponding vertices in V are adjacent in H. We call such a graph G the *double cover* of H. Together with the above lemma, this gives us the desired bipartite expanders, stated in the form of the following corollary.

Corollary 2 *Let H be a Δ-regular graph on n vertices with $\lambda(H) = \lambda$. Let $G = (A, B, E)$ be the double cover of H. Then for every subset $X \subseteq A$ with $|X| \geq bn$, we have $|\Gamma(X)| \geq (1 - \frac{\lambda^2}{b})n$ where $\Gamma(X) \subseteq B$ is the set of all nodes with some neighbor in X.*

[2]Of course, for this definition to be non-trivial to achieve, we need to define it for an infinite family of regular graphs. Throughout the paper, we always implicitly assume that we are concerned with an infinite family of codes and expanders.

Expander graphs with $\lambda \ll 1$ also have good *isoperimetric* properties. Loosely speaking this means that the fraction of edges between two large sets of vertices approximately equals the product of the densities of those sets. The formal lemma, stated below, is folklore (see for example Corollary 2.5 in [2, Chap. 9]).

Lemma 3 *Let H be a Δ-regular graph with $\lambda(H) = \lambda < 1$. Let $G = (A, B, E)$ be the double cover of H. Then for every pair of subsets $X \subseteq A$ and $Y \subseteq B$, we have*

$$\left| \frac{E(X:Y)}{\Delta |X|} - \frac{|Y|}{|B|} \right| \le \lambda \sqrt{\frac{|Y|}{|X|}} \ .$$

Thus a low value of λ achieves both good vertex expansion and isoperimetric properties. It is known, however, that the best value of λ one can hope for in an infinite family of Δ-regular graphs is $\frac{2\sqrt{\Delta-1}}{\Delta} - o(1)$. Amazingly enough, there are explicitly known constructions of an infinite family of Δ-regular graphs $\{G_i\}_{i\ge1}$ with $\limsup_{i\to\infty} \lambda(G_i) = \frac{2\sqrt{\Delta-1}}{\Delta} < \frac{2}{\sqrt{\Delta}}$. These graphs, which are called Ramanujan graphs, were constructed independently in [7] and [8].

4 Codes with rate $\Omega(\varepsilon^2)$ decodable up to a fraction $(1/2 - \varepsilon)$ of errors

Theorem 4 *For any $\varepsilon > 0$ there is an explicitly specified code family with rate $\Omega(\varepsilon^2)$, relative distance at least $(1-\varepsilon)$ and alphabet size $2^{O(1/\varepsilon^2)}$, such that a code of blocklength n from the family can be (a) encoded in $O(n/\varepsilon^2)$ time, and (b) uniquely decoded from up to a fraction $(1/2-\varepsilon)$ of errors in $O(n/\varepsilon^2)$ time.*

Proof: We need the following two combinatorial objects for our code construction:

(1) A binary asymptotically good $[n,k]_2$ linear code C, encodable and uniquely decodable from a fraction $\gamma > 0$ of errors in *linear time* (here γ is an absolute positive constant). An explicit construction of such a code is known [11, 12].

(2) A Δ-regular bipartite graph $G = (A, B, E)$ with $|A| = |B| = n$, such that:

   (a) for every set $X \subset A$ with $|X| \ge \gamma n$, if Y is the set of neighbors of X in G, then $|Y| \ge (1-\varepsilon)|B|$.

   (b) for every set $Y \subset B$ with $|Y| \ge (1/2+\varepsilon)n$, the set $X' \subseteq A$ defined by

$$X' = \{x \in A : \ x \text{ has as many neighbors in } B \setminus Y \text{ as in } Y\} \tag{2}$$

   has size at most γn.

A graph as in (2) above with $\Delta = O(\frac{1}{\gamma\varepsilon^2})$ can be obtained from a Ramanujan graph (i.e., an expander with second largest eigenvalue $O(1/\sqrt{\Delta})$). Indeed let $H = (V, E')$ be a Δ-regular Ramanujan graph with $\lambda(H) = \lambda = O(1/\sqrt{\Delta})$. Take G to be the double cover of H. We will prove that G both the properties (a) and (b) described above. For property (a), we apply Corollary 2 with the choice $b = \gamma$. This gives that for all $X \subseteq A$ with $|X| \ge \gamma n$, the set $Y \subseteq B$ of all nodes with neighbors in X satisfies

$$|Y| \ge \Big(1 - \frac{\lambda^2}{\gamma}\Big) n \ge \Big(1 - O\Big(\frac{1}{\Delta\gamma}\Big)\Big) n \ge (1-\varepsilon^2) n > (1-\varepsilon) n \ ,$$

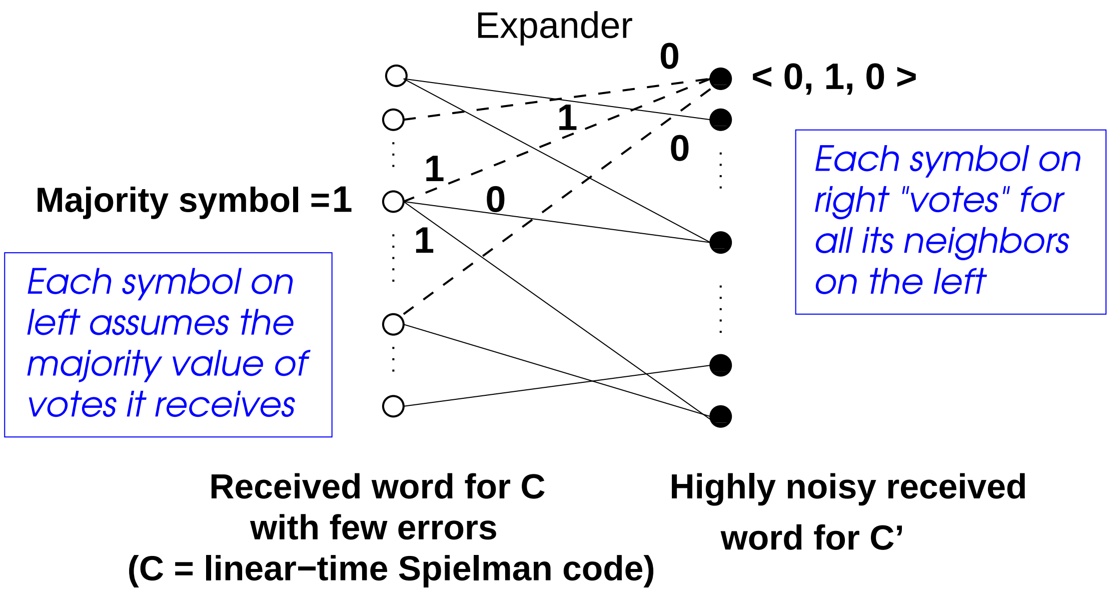

Figure 3: The majority voting based decoding algorithm

for $\Delta = \Omega(\frac{1}{\gamma\varepsilon^2})$.

For the second property (b), assume that $|Y| \geq (1/2+\varepsilon)n$ and let X' be defined as in (2). We need to prove that $|X'| \leq \gamma n$. By the definition of X', we have $E(X' : Y) \leq \Delta|X'|/2$. Applying the result of Lemma 3, we know that

$$\begin{aligned} \frac{E(X':Y)}{\Delta|X'|} &\geq \frac{|Y|}{n} - \lambda\sqrt{\frac{|Y|}{|X'|}} \\ &\geq \left(\frac{1}{2}+\varepsilon\right) - \lambda\sqrt{\frac{|Y|}{|X'|}} \ . \end{aligned}$$

Together with $E(X' : Y) \leq \Delta|X'|/2$, this implies that

$$|X'| \leq \frac{\lambda^2|Y|}{\varepsilon^2} = O\Big(\frac{n}{\Delta\varepsilon^2}\Big) \leq \gamma n \ ,$$

for $\Delta = \Omega(\frac{1}{\gamma\varepsilon^2})$. Hence we conclude that the graph G required in (2) above exists with degree $\Delta = O(\frac{1}{\gamma\varepsilon^2}) = O(1/\varepsilon^2)$, since γ is an absolute constant.

Given the code C and graph G, our final code, call it C', is constructed as follows: to encode a message x according to C', we first encode it into $C(x)$, and then push symbols of $C(x)$ along the edges of G. The i'th symbol of the codeword $C'(x)$, for $1 \leq i \leq n$, comprises of the collection of the symbols received at the i'th node of the right side B of G.

Since C has constant rate, clearly C' has rate $\Omega(1/\Delta) = \Omega(\varepsilon^2)$. Since C is uniquely decodable up to a fraction γ of errors, its relative distance must be at least 2γ, and this together with the expansion property (a) of G clearly implies that C' has relative distance at least $(1-\varepsilon)$.

The encoding time for C' is the same as for C (i.e., linear), plus $O(n\Delta) = O(n/\varepsilon^2)$. In order to decode a received word z which differs from a codeword $C'(x)$ in at most a fraction $(1/2-\varepsilon)$ of positions, we first perform the following key voting step, which is illustrated in Figure 3: *Each node v in A recovers the bit which is the majority of the neighbors of v in B (ties broken arbitrarily).*

Since z and $C'(x)$ agree on at least $(1/2+\varepsilon)n$ positions, appealing to the property (b) of the graph G, we conclude that at most γn nodes in A recover incorrect bits of $C(x)$ in the above voting procedure. Then, by the property of the code C, we can decode x in linear time. The total decoding time is again equal to $O(n/\varepsilon^2)$ for the first stage and then a further $O(n)$ time for the decoding of C. Hence the total decoding time is $O(n/\varepsilon^2)$, as claimed. □

5 Binary codes with rate $\Omega(\varepsilon^4)$ decodable up to a fraction $(1/4-\varepsilon)$ of errors

In this section we show how to augment the linear-time codes from the previous section in order to obtain binary codes with linear-time encoding, and linear-time decoding up to a fraction $(1/4-\varepsilon)$ of errors.

Theorem 5 *For every $\varepsilon > 0$ there is a binary linear code family of rate $\Omega(\varepsilon^4)$ and relative distance at least $(1/2-O(\varepsilon))$, such that a code of blocklength N from the family can be uniquely decoded from up to a fraction $(1/4-\varepsilon)$ of errors in $O(N/\varepsilon^2+2^{O(1/\varepsilon^4)})$ time, and can be encoded in $O(N+2^{O(1/\varepsilon^2)})$ time. The code can be constructed in probabilistic $O(1/\varepsilon^4)$ or deterministic $2^{O(1/\varepsilon^4)}$ time.*

Proof: The code is constructed by concatenating the code from Theorem 4 with a suitable binary code. Details follow. Let C' be the code from the Theorem 4.[3] The alphabet size of C' is $Q = 2^{O(1/\varepsilon^2)}$. Let C_3 be any $[O(\lg Q/\varepsilon^2), \lg Q]_2$ linear code with relative distance at least $(1/2-\varepsilon)$. Such a code can be constructed by a picking random linear code from a "Wozencraft ensemble" in probabilistic $O(1/\varepsilon^4)$ time or by a brute-force search in such an ensemble in $2^{O(1/\varepsilon^4)}$ time. We concatenate C' with C_3 obtaining a binary linear code , say C^*, of blocklength $N = O(n/\varepsilon^4)$, rate $\Omega(\varepsilon^4)$ and relative designed distance at least $\delta \stackrel{\text{def}}{=} (1-\varepsilon)(1/2-\varepsilon) = (1/2-O(\varepsilon))$.[4] Since C' can be encoded in $O(n/\varepsilon^2)$ time, the encoding of C^* can be performed in $O(n/\varepsilon^4)$ time (since each encoding by C_3 can be done in $1/\varepsilon^4$ time using a look-up table building which takes a one-time cost of $2^{O(1/\varepsilon^2)}$ time and space). As the overall blocklength of C^* equals $N = O(n/\varepsilon^4)$, the claimed encoding time holds.

We will now show how to unique decode C^* from a fraction $\delta/2$ of errors in linear-time. Since $\delta = (1-\varepsilon)(1/2-\varepsilon)$ and $\varepsilon > 0$ is arbitrary, this will imply the claimed result. To this end, we use the following general result on decoding concatenated codes. This result appears to be folklore and has been observed and used by several authors, including Justesen [6] who gave an algorithm to decode his asymptotically good concatenated code construction up to the product bound (i.e., half the product of the distances of the outer and inner codes). The basic idea and inspiration behind this method comes from the work of Forney [4] on Generalized Minimum Distance (GMD) decoding.

[3] Actually, we will need to make slight changes in the assumptions about the components used in the construction of C' in Theorem 4, namely in the assumptions about the expander graph G. But the construction of C' itself (given the left code C and the expander G), as well all its properties claimed in Theorem 4, remain unaltered — we will only pose some stronger requirements on G and the decodability of C'. We will discuss these and justify how they can be achieved without any loss in rate later in the proof.

[4] The code C^* will be linear since C_3 is linear and it is easy to check that the construction from Theorem 4 gives an additive code C'.

Proposition 6 (GMD decoding of concatenated codes) *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 $\tilde{D} \le \mathsf{dist}(C_{\text{out}})$. Then there exists an algorithm $\mathcal{A}$ running in $O(NT_{\text{in}} + dT_{\text{out}})$ time that uniquely decodes $\mathbf{C}$ from any pattern of less than $\frac{d\tilde{D}}{2}$ errors.*

The number of times the decoding algorithm for C_{out} is invoked by the final algorithm $\mathcal{A}$ is at most d (and hence the dT_{out} term in the runtime), and *independent* of the outer blocklength N. This will be crucial for us. The fact that GMD decoding allows for such a few invocations of the outer decoding algorithm has been observed before (for example, by Justesen [6] who makes a one line comment on it). A formal proof of the above proposition can be found, for example, in [5, Appendix A].

Since the relative distance of the inner code C_3 in our concatenated construction C^* is at least $(1/2-\varepsilon)$, it can be uniquely decoded up to a fraction $(1/2-\varepsilon)/2$ of errors. Since the blocklength of C_3 is $O(1/\varepsilon^4)$, this can be accomplished in $O(1/\varepsilon^4)$ time using table look-up where building up the table takes a one-time cost of $2^{O(1/\varepsilon^4)}$ time. Below we will show that outer code C' can be decoded in $O(n/\varepsilon^2)$ time from any pattern of fraction s of erasures and fraction e of errors as long as $2e+s \le (1-\varepsilon)$. Appealing to the above lemma, therefore, we will have an $O(n/\varepsilon^4 + 2^{O(1/\varepsilon^4)} + \varepsilon^{-4} \cdot n/\varepsilon^2)$ time algorithm to decode C^* up to a fraction $(1/2-\varepsilon)(1-\varepsilon)/2 = \delta/2$ of errors (the distance d of the inner code C_3 equals $O(1/\varepsilon^4)$). Hence the total decoding time is $O(n/\varepsilon^6 + 2^{O(1/\varepsilon^4)}) = O(N/\varepsilon^2 + 2^{O(1/\varepsilon^4)})$ (since $N = O(n/\varepsilon^4)$).

It remains to show that the outer code C' can be decoded, in linear time, from a fraction s of erasures and a fraction e of errors for any e,s that satisfy $2e+s \le (1-\varepsilon)$. To this end, recall the construction of C' from Theorem 4 using a binary asymptotically good code C and the expander G. For this proof, we assume some small changes in the parameters of the expander $G = (A,B,E)$. Specifically, we assume that G satisfies a property (b'), defined below, which is stronger than the property (b) from Equation (2) that was used in Theorem 4.[5]

(b') For every subset $X \subset A$ with $|X| \ge \gamma n/2$ and every $Y \subseteq B$, we have

$$\left| \frac{|E(X:Y)|}{|X|\Delta} - \frac{|Y|}{|B|} \right| \le \varepsilon/3 \; . \tag{3}$$

Note the above strongly resembles the isoperimetric property of Lemma 3, and in fact is implied by it provided $\lambda = O(\varepsilon\sqrt{\gamma})$. Therefore, by using Ramanujan graphs which achieve $\lambda = O(1/\sqrt{\Delta})$, we can achieve the above property with $\Delta = O(\frac{1}{\gamma\varepsilon^2})$. Thus, we can achieve the stronger property (b') above with essentially no asymptotic increase in the degree Δ. Hence the rate of C' is still $\Omega(\varepsilon^2)$.

We now proceed to describe the claimed $O(n/\varepsilon^2)$ time error-erasure decoding algorithm for C' that can successfully decode from a fraction s of erasures and a fraction e of errors for any s,e satisfying $2e+s \le (1-\varepsilon)$. Let S be the set of erasures in the received word z, and let F be the set of errors. The decoding algorithm proceeds by the same *majority voting* scheme as before.

[5] Of course we could have defined and used this stronger property in the proof of Theorem 4 itself. But, we chose to avoid this since we wanted to state only the minimum required property on the expander to make the construction work. This has the advantage that a better construction of the combinatorial object might be possible which in turn might lead to improvements in the code construction.

That is, each node on the left side A of the bipartite graph G assumes the bit which is majority of the values of the neighbors of that node on the right side B. The difference now is that the nodes corresponding to the set S of erased symbols do not participate in the voting. In order to prove the correctness of the decoding algorithm, we would like to prove that all but a fraction γ of the positions on the left (that of the code C) receive the correct bit as the majority vote by this procedure. Then we can complete the decoding using the linear-time decoder for C (which by assumption can correct up to a fraction γ of errors). Note that the decoding is the time to decode C, i.e. $O(n)$, plus the time to perform the majority votings, which is $O(n/\varepsilon^2)$, for a total of $O(n/\varepsilon^2)$ time, as desired.

To prove that at most a fraction γ of the nodes in A receive the incorrect bit as the majority vote by the above voting procedure, define $X \subseteq A$ to be the set of nodes which have at most a fraction $(1 - s - \varepsilon/3)$ of neighbors in the set $B \setminus S$. Also, define $X' \subseteq A$ to be the set of nodes which have at least a fraction $(e + \varepsilon/3)$ of neighbors in F. It is easy to infer using the property (b') of Equation (3) that $|X|, |X'| \le \gamma n/2$.

Now consider any node from $A \setminus (X \cup X')$. It has less than a fraction $(e + \varepsilon/3)$ of neighbors in F. Moreover, it has more than a fraction $(1 - s - \varepsilon/3) - (e + \varepsilon/3) = (1 - s - e - 2\varepsilon/3)$ of neighbors in $B \setminus (S \cup F)$ (i.e., in the set of correct positions). Since $2e + s \le (1 - \varepsilon)$, we have $(e + \varepsilon/3) \le (1 - s - e - 2\varepsilon/3)$. Therefore, every node in $A \setminus (X \cup X')$ has more neighbors among the correct positions (i.e., those in $B \setminus (S \cup F)$) than among the incorrect positions (i.e. those in F). It follows that the the voting procedure works correctly for all nodes in $A \setminus (X \cup X')$, and hence for all but a fraction γ of the codeword positions of C (since $|X|, |X'| \le \gamma n/2$). This is exactly what we desired to show, and the proof is complete. $\square$

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. on Information Theory*, 38 (1992), pp. 509-516.

[2] N. Alon and J. Spencer. *The Probabilistic Method.* John Wiley and Sons, Inc., 1992.

[3] A. Barg and G. Zémor. Error exponents of expander codes. *IEEE Transactions on Information Theory*, to appear, 2001.

[4] G. D. Forney. Generalized Minimum Distance Decoding. *IEEE Trans. Inform. Theory*, Vol. 12, pp. 125-131, 1966.

[5] V. Guruswami. *List Decoding of Error-Correcting Codes.* PhD thesis, Massachusetts Institute of Technology, August 2001.

[6] J. Justesen. A class of constructive asymptotically good algebraic codes. *IEEE Transactions on Information Theory*, 18:652–656, 1972.

[7] A. Lubotzky, R. Phillips and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3): 261-277, 1988.

[8] G. A. Margulis. Explicit group-theoretical constructions of combinatorial schemes and their applications to the design of expanders and superconcentrators. *Problems of Information Transmission*, 24:39–46, 1988.

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

[10] M. Sipser and D. Spielman. Expander Codes. *IEEE Transactions on Information Theory*, Vol 42, No 6, pp. 1710-1722, 1996.

[11] D. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, Vol 42, No 6, pp. 1723-1732, 1996.

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

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