Expander-based Constructions of Efficiently Decodable Codes

(Extended Abstract)

Venkatesan Guruswami* Piotr Indyk*

Abstract

We present several novel constructions of codes which share the common thread of using expander (or expander-like) graphs as a component. The expanders enable the design of efficient decoding algorithms that correct a large number of errors through various forms of "voting" procedures. We consider both the notions of unique and list decoding, and in all cases obtain asymptotically good codes which are decodable up to a "maximum" possible radius and either (a) achieve a similar rate as the previously best known codes but come with significantly faster algorithms, or (b) achieve a rate better than any prior construction with similar error-correction properties. Among our main results are:

- *Codes of rate $\Omega(\varepsilon^2)$ over constant-sized alphabet that can be* list *decoded in quadratic time from $(1 - \varepsilon)$ errors. This matches the performance of the best algebraic-geometric (AG) codes, but with much faster encoding and decoding algorithms.*
- *Codes of rate $\Omega(\varepsilon)$ over constant-sized alphabet that can be* uniquely *decoded from $(1/2 - \varepsilon)$ errors in near-linear time (once again this matches AG-codes with much faster algorithms). This construction is similar to that of [1], and our decoding algorithm can be viewed as a positive resolution of their main open question.*
- *Linear-time encodable and decodable* binary *codes of positive rate*[1] *(in fact, rate $\Omega(\varepsilon^4)$) that can correct up to $(1/4 - \varepsilon)$ fraction errors. 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 [19] and the linear-time decodable codes of [18, 22].*

*MIT Laboratory for Computer Science, Cambridge, MA 02139. Email: {venkat,indyk}@theory.lcs.mit.edu. The research of the first author was supported in part by an IBM Graduate Fellowship and in part by funding from NSF CCR 9875511, NSF CCR 9912342, and NTT Award MIT 2001-04 .

[1] Very recently, the authors of this paper were able to improve the rate to $\Omega(\varepsilon^3)$. The details of this result will appear elsewhere.

1 Introduction

Error-correcting codes are combinatorial objects used for the purpose of reliable transmission of information on a noisy communication channel. Constructing efficiently decodable error-correcting codes that can correct a large number of errors is the central problem of coding theory, and is of tremendous practical importance as well. In this paper we present several novel constructions of error-correcting codes, together with very efficient decoding algorithms that correct a large fraction of errors. All of the constructions employ expander-like graphs, which facilitate efficient decoding algorithms through various forms of "voting" procedures.

We are interested in both binary codes and codes over large (but constant-sized) alphabets with good decodability. Within each class, we further consider both the unique and list decoding versions. Unique decoding algorithms are required to output a unique answer and their error-correction capability is limited by half the minimum distance of the code. Under list decoding, the decoder is allowed to output a small list of codewords which must include the transmitted codeword [4, 21]. This permits recovery from errors beyond half the minimum distance, and several recent results have given efficient list decoding algorithms to do so [17, 9, 16, 10]. For both unique and list decoding, we are interested in constructing codes and decoding algorithms that correct a "maximum" possible fraction of errors. Specifically, for codes over large alphabets we are interested in unique (resp. list) decoding up to $(1/2 - \varepsilon)$ (resp. $(1 - \varepsilon)$) fraction errors, while for binary codes our aim is to unique (resp. list) decode up to $(1/4 - \varepsilon)$ (resp. $(1/2 - \varepsilon)$) errors. For such code constructions, our goal is to achieve good rate together with fast encoding and decoding algorithms.

Our results can be grouped into four classes. Each class (save the first one) contains constructions of both binary and non-binary codes. The detailed specification of their parameters is presented in Figure 1. We next present a discussion of the individual results and compare them with previously known constructions.

No	Alphabet	Decoding radius	Decoding method	Rate	Encoding time	Decoding time	Construction time (probabilistic)*
1	$2^{\varepsilon^{-1}\log(1/\varepsilon)}$	$1-\varepsilon$	list	ε^2	$n\log n$	n^2	$\log^2 n\cdot\log(1/\varepsilon)/\varepsilon$
2a	$2^{\varepsilon^{-1}\log(1/\varepsilon)}$	$1-\varepsilon$	list	ε	$n^{2(1-\gamma)}\log^2 n$	$2^{n^\gamma\log(1/\varepsilon)}$	$n^{2(1-\gamma)}/\varepsilon$
2b	2	$1/2-\varepsilon$	list	ε^3	$n^{2(1-\gamma)}\log^2 n$	$2^{n^\gamma\log(1/\varepsilon)}$	$n^{2(1-\gamma)}/\varepsilon$
3a	$2^{1/\varepsilon}$	$1/2-\varepsilon$	unique	ε	$n\log^{O(1)} n$	$n^{1+\gamma}$	explicit
3b	2	$1/4-\varepsilon$	unique	ε^3	$n\log^{O(1)} n$	$n^{1+\gamma}+2^{1/\varepsilon^3}$	$1/\varepsilon^3$
4a	$2^{1/\varepsilon^2}$	$1/2-\varepsilon$	unique	ε^2	n/ε^2	n/ε^2	explicit
4b	2	$1/4-\varepsilon$	unique	ε^4	$n+2^{1/\varepsilon^2}$	$n/\varepsilon^2+2^{1/\varepsilon^4}$	$1/\varepsilon^4$

Figure 1. The parameters of our codes. n stands for the length of the code. For readability, the $O(\cdot)$ and $\Omega(\cdot)$ notation has been omitted. The value of γ is in the interval $(0,1]$; its value influences the rate by a constant factor. The decoding radius shows the fraction of errors which the decoding algorithms can correct. .
$*$A detailed discussion on the construction times is presented later in this Section.

Our first code (1) enables efficient list-decodability from up to $(1-\varepsilon)$ fraction errors, for an arbitrary constant $\varepsilon > 0$. Its distinguishing feature is quadratic decoding time and fairly high ($\Omega(\varepsilon^2)$) rate, while maintaining constant (i.e., independent on n) alphabet size. The only known constructible codes with comparable parameters are certain families of algebraic-geometric codes (henceforth, AG-codes) [20, 7]. Such AG-codes have quadratic rate and constant (although polynomial in $1/\varepsilon$) alphabet size, but their (list) decoding, however, is much more difficult. It is known how to decode them in $O(n^3)$ time using certain auxiliary advice (of length $O(n^3)$) [16, 9, 11]; however, it not known how to compute this information in sub-exponential (randomized or deterministic) time. Only very recently [12] showed how to construct the generator matrix for certain AG codes in $O(n^3)$ time. In comparison, our construction time (although probabilistic) is essentially negligible.

The third code (2a) also enables list decodability up to an $(1-\varepsilon)$ fraction errors. Its distinguishing feature is the *optimal* $\Omega(\varepsilon)$ rate. The only previously known codes with such rate were purely random codes (even Reed-Solomon codes that have super-constant alphabet size only guarantee $\Omega(\varepsilon^2)$ rate). However, the best known decoding time for random codes is $2^{O(n)}$, and it is likely that no significantly better algorithm exists. Our codes also have significant random components; however, they can be decoded substantially faster in sub-exponential time. The binary version of the aforementioned codes, which correct $(1/2-\varepsilon)$ fraction errors, also beat the rate of best constructive codes of [8] (which is $\Omega(\varepsilon^4)$, improving earlier bound of $\Omega(\varepsilon^6)$ [10]). They are, however, not as good as random binary codes, which enjoy quadratic rate [8].

The remaining codes are supported by unique decoders (i.e., whose decoding radius is bounded by half of the minimum designed distance). The first two of them (3a,3b) provide a more efficient alternative to codes based on concatenation schemes involving AG-codes, which achieve a similar rate, but only have complicated $O(n^3)$-time decoding and construction algorithms. In comparison, our decoding algorithms are simple and have running times arbitrarily close to linear; however, the constant in the rate is affected by the exponent in the running time. We also mention that our codes are very similar to the codes constructed in [1]. In the latter paper the authors asked if there is a polynomial time decoding algorithm for their codes; thus our results can be viewed as positive resolution of the main open problem from their paper.

Our last class of codes comes with linear-time encoding and decoding algorithms (call such codes *linear-time codes*). Although the rate is not optimal,[2] they are the first known asymptotically good linear-time codes that 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 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 codes (due to Spielman [19]), or for that matter over that of the linear-time decodable codes of Sipser and Spielman [18], Zémor [22] and Barg and Zémor [3]. To make a concrete comparison, the largest fraction of errors corrected by the linear-time codes in [19] (at least the bound directly deducible from the paper) seems to be about 4.6×10^{-7}. The linear-time decodable codes from [22] can correct about 3.025×10^{-3} fraction errors with positive rate. In comparison, we can correct a $1/4-\varepsilon$ fraction of errors for arbitrary $\varepsilon > 0$.

Finally, we stress that though we state our results for

[2] Recently the authors improved the rate of the linear time constructions to the "optimal" values achieved by constructions (3a,3b). These results will appear elsewhere.

binary codes, analogous results and bounds for unique decoding up to $\frac{(1-1/q)}{2} - \varepsilon$ and list decoding up to $(1 - 1/q - \varepsilon)$ fraction errors, also hold for codes over alphabets of size q, for any fixed $q \geq 2$.

Construction times. Almost all of our codes (3a,4a are the exceptions) use the probabilistic method to obtain certain gadgets needed for the code construction. These gadgets are then used together with explicitly specified objects. The probabilistic method generates such building blocks with high probability. However, there is no easy way to *verify* that a given combinatorial gadget has the required properties. Therefore, our construction algorithms are *randomized Monte Carlo*. We note, however, that our probabilistic algorithms using R random bits can be trivially converted into deterministic algorithms using $O(R)$ space and running in $2^{O(R)}$ time. In many cases the resulting *deterministic* construction time is polynomial, or even sub-linear. It is not the case for codes 1 (the time is quasi-polynomial) and 2 (the time is exponential). Nevertheless, we show (in Section 3.3) that the former code can be constructed deterministically in time $n^{O(\varepsilon^{-1}\log(1/\varepsilon))}$.

We stress that modulo the gadget construction, generating each symbol of a codeword can be done in polylogarithmic time.

1.1 Our techniques

At a high level, all of our codes are constructed using a similar scheme. The basic components of the constructions are: a "left" code (say, C), a "dispersing" graph G, and (in case of binary codes) an efficient "right" binary code (say C'). The left code C is typically a concatenation of efficient list-decodable codes, i.e., either the Reed-Solomon codes or certain highly decodable "pseudolinear" codes whose existence we prove in Section 3. Such codes can either be found by brute-force or, one can pick a code at random and thus get a much faster probabilistic construction that works with high probability. The graph G is an *expander* of sufficiently large degree.

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 specific implementations of this idea depend on the code). The juxtaposition of symbols "sent" to each right node of G forms a symbol of the codeword y. If we want to construct a binary code, we add a final level of concatenation with the binary code C'.

The purpose of the distribution step is, roughly speaking, to convert an arbitrary distribution of errors that could exist between the various blocks of the concatenated code into a near-uniform distribution. This permits recovery of a (somewhat corrupted) codeword x from (heavily corrupted) y by a variety of voting schemes. In the simplest scheme (used in codes (4a,4b)), each left node assumes the symbol which is the most popular among its neighbors on the right. In more elaborate schemes, a list of potential symbols is constructed and possibly propagated through the concatenation levels of the left code.

The specifics of the implementation of the above ideas depend on the actual code construction. For the code (1), we take the left code C to be a concatenation of a Reed-Solomon code and a pseudolinear code. Such a code can be list decoded in quadratic time using the Reed-Solomon decoding algorithm of [14]. The codes (2a,2b) are constructed by picking C to be a multilevel concatenation of a constant number of pseudolinear codes. These codes can perform list decoding when given, as an input, a vector of lists (one per codeword position) such that at least half of the lists contain the correct symbol. More importantly, they have fixed constant rate *independent* from the length of the lists. This allows the decoding algorithm to propagate the candidate symbols through the concatenation levels without decreasing the rate by a large factor at each level. The parameters are so picked that the decoding of each of these pseudolinear codes and the final code can be done in sub-exponential time.

For the constructions (3a,3b), we use for left code a list-decodable code that can correct a number of errors at most, say, $3/4$ times the minimum distance (any fraction greater than $1/2$ is sufficient) – such a code of constant rate and with a near-linear time list decoding algorithm can be built a concatenation of several levels of Reed-Solomon codes followed by any large distance code. Finally, the codes (4a,4b) use the linear-time encodable/decodable codes of Spielman [19] as the left code. By taking the graph G to be an expander with good "mixing" properties, we can transform a heavily corrupted codeword y to a much less corrupted codeword x of C. The latter can be decoded using the linear-time decoding algorithm for C.

1.2 Prior work

In recent years, there have been several papers which construct codes using expanders. These use expanders in one of two ways: either to construct the parity check matrix [18, 19, 22] or the generator matrix [1]. Our codes constructions follow the spirit of the second approach, in the sense that we also use expander-like graphs to distribute the symbols of the message. However, our constructions are more involved than the construction of [1], since we want to make the codes efficiently decodable. Also, we use much stronger properties of the graphs than just (vertex) expansion: specifically we use the *mixing* properties of expanders based on their spectral gap.

There has also been work on sub-exponential time *unique* decoding algorithms. In particular, the algorithm of [23] can unique decode certain large distance binary codes in $2^{O(\sqrt{n})}$ time. In contrast, our codes (2a,2b) can be *list* decoded in $2^{O(n^\delta)}$ time for arbitrary $\delta > 0$.

Due to space limitations, we omit several of the proofs in this extended abstract. The proofs will soon appear in the first author's Ph.D. dissertation as well as the full version of the paper.

2 Preliminaries

2.1 Definitions and Notation

For a prime power q, let $\mathbb{F}_q$ denote a finite field of cardinality q. When the field structure is not used, we denote an alphabet of size q by $[q] = \{1, 2, \ldots, q\}$. A q-ary code C of *blocklength* n is simply a subset of $\mathbb{F}_q^n$ (or $[q]^n$). An important class of codes are linear codes which are linear subspaces of $\mathbb{F}_q^n$. We refer to a code which is a k-dimensional subspace of $\mathbb{F}_q^n$ as an $[n, k]_q$ code, and k is called the *dimension* of the code. By abuse of notation, for a general non-linear q-ary code C with $|C| = M$, we will call the quantity $\log_q M$ the dimension of the code, and refer to such a code as an $(n, \log_q M)_q$ code. The *rate* of a q-ary code C of blocklength n is defined as $R(C) = (\log_q |C|)/n$.

For vectors $x, y \in \mathbb{F}_q^n$, let $\mathsf{dist}(x, y)$ denote the Hamming distance between them, and let $\Delta(x, y) = \frac{\mathsf{dist}(x,y)}{n}$ be the relative (fractional) distance between x and y. The *minimum distance* (or simply, *distance*) of a code C, denoted $\mathsf{dist}(C)$, is the quantity $\min_{x,y \in C, x \neq y}\{\mathsf{dist}(x, y)\}$. The *relative distance* of a code of blocklength n is defined as $\delta(C) = \mathsf{dist}(C)/n$.

Since the main thrust of this paper is the asymptotic performance of the codes, we define analogs of the quantities above for infinite families of codes. An infinite family of (binary) codes is a family $\mathcal{C} = \{C_i | i \in \mathbb{Z}\}$ where C_i is a code of blocklength n_i with $n_i > n_{i-1}$. We define the *rate* of an infinite family of codes $\mathcal{C}$ to be $R(\mathcal{C}) = \liminf_i \{R(C_i)\}$. We define the *(relative) distance* of an infinite family of codes $\mathcal{C}$ to be $\delta(\mathcal{C}) = \liminf_i \{\delta(C_i)\}$. We also denote by $q(\mathcal{C})$ the size of the alphabet over which the code family $\mathcal{C}$ is defined.

We now define the list decoding radius of a code. For non-negative integer r and $x \in \mathbb{F}_q^n$, let $B(x, r)$ denote the ball of radius r around x, i.e., $B(x, r) = \{y \in \mathbb{F}_q^n | \mathsf{dist}(x, y) \leq r\}$. For integers e, ℓ, a code $C \subseteq \mathbb{F}_q^n$ is said to be (e, ℓ)-*list decodable* if every ball of radius e has at most ℓ codewords, i.e. $\forall\ x \in \mathbb{F}_q^n$, $|B(x, e) \cap C| \leq \ell$. Note that in this terminology a code of distance d is $((d-1)/2, 1)$-list decodable, which is the same as saying such a code can *uniquely decoded* up to $(d-1)/2$ errors.

Definition 1 (List Decoding Radius) *For a code C and list size ℓ, the* list of ℓ decoding radius *of C, denoted* $\mathsf{radius}(C, \ell)$ *is defined to be the* maximum *value of e for which C is (e, ℓ)-list decodable. We also normalize this quantity and define $\mathsf{LDR}_\ell(C) = \mathsf{radius}(C, \ell)/n$, if n is the blocklength of the code.*

Definition 2 (List Decoding Radius for code families) *For an infinite family of codes $\mathcal{C} = \{C_i\}_{i \geq 1}$ and an integer ℓ, define the* list of ℓ decoding radius *of $\mathcal{C}$, denoted $\mathsf{LDR}_\ell(\mathcal{C})$, to be $\mathsf{LDR}_\ell(\mathcal{C}) = \liminf_i \mathsf{LDR}_\ell(C_i)$.*

For some our results, the following alternate notion of good list decodability proves extremely useful – for purposes of disambiguation from (e, ℓ)-list decodability, we call this notion "list recoverability".

Definition 3 *For α, $0 < \alpha < 1$, and integers $L \geq \ell \geq 2$, a q-ary code C of blocklength n is said to be (α, ℓ, L)-*list recoverable *if given arbitrary "lists" $L_i \subseteq \mathbb{F}_q$ of size at most ℓ for each i, $1 \leq i \leq n$, the number of codewords $\mathbf{c} = \langle c_1, \ldots, c_n \rangle \in C$ such that $c_i \in L_i$ for at least αn values of i, is at most L.*

Remark: A code of blocklength n is $(\alpha, 1, L)$-list recoverable then it is also $(\alpha n, L)$-list decodable.

3 Basic Techniques and Existence results for list-decodable codes

In this section, we prove existence results using the probabilistic method for codes which serve as inner codes in our concatenated code constructions. We first define the notion of "pseudolinear" code families. This notion appears to be new, and it plays a critical role in translating list-decodability results for general, non-linear codes into similar results for codes, which albeit not linear, still have a succinct description, and allow for efficient encoding. We believe this is a simple yet powerful tool which will find applications in contexts outside those discussed in this paper. In our applications, these pseudolinear codes, which are typically used as inner codes in suitable concatenated schemes, are critical in getting efficient constructions for our codes.

3.1 Pseudolinear (Code) Families

Informally, define an L-wise independent $(n, k)_q$-code family $\mathcal{F}$ to be a sample space of codes C that map k symbols over $\mathbb{F}_q$ to n symbols over $\mathbb{F}_q$ such that any L of the (non-zero) codewords are completely independent (for a code picked uniformly at random from $\mathcal{F}$). A random code picked from such a family often tends to have very good list decoding properties for decoding with list size L, owing to the mutual independence of any set of L codewords. An example of such a family is the space of all general, non-linear q-ary codes of blocklength n and with q^k codewords, which is clearly L-wise independent, for all L, $1 \le L \le q^k$. While a random, non-linear code has excellent randomness properties, it comes from a very large sample space and there is no succinct representation of a general code from the family. [3] We now define a family of codes which we call *pseudolinear* that has the desired L-wise independence property and in addition is succinct. Thus a random code drawn this family has the desired randomness properties, can be succinctly represented and has an efficient encoding procedure.

Definition 4 *For a prime power q, integer $L \ge 1$, and positive integers k, n with $k \le n$, an (n, k, L, q)-pseudolinear family $\mathcal{F}(n, k, L, q)$ of codes is defined as follows. Let H be the parity check matrix of any q-ary linear code of blocklength $(q^k - 1)$, minimum distance at least $(L+1)$ and dimension $q^k - 1 - O(kL)$ (for example, one can use parity check matrices of q-ary BCH codes of designed distance $(L+1)$). A random code C_A in the pseudolinear family $\mathcal{F}(n, k, L, q)$ is specified by a random $n \times O(kL)$ matrix A over $\mathbb{F}_q$. Under the code C_A, a message $x \in \mathbb{F}_q^k \setminus \{0\}$ is mapped to $A \cdot H_x \in \mathbb{F}_q^n$ where $H_x \in \mathbb{F}_q^{O(kL)}$ is the column of H indexed by x (viewed as an integer in the range $[1, q^k)$). (We also define $H_0 = \mathbf{0}$ to be the all-zeroes message vector.)*

Given $1 \le x < q^k$, a description of the column H_x can be obtained in time polynomial in k and $\log q$, since there are explicit descriptions of the parity check matrices of BCH codes of distance at least $(L+1)$ and blocklength $(q^k - 1)$, in terms of the powers of the generating element of $\mathrm{GF}(q^k)$ over $\mathrm{GF}(q)$. Hence encoding as per these codes is an efficient operation. In addition to these complexity issues, the crucial combinatorial property about these pseudolinear codes that we exploit is that every set of L fixed non-zero codewords of the code C_A, for a random A, are completely independent. This is formalized below and the easy proof follows from the fact that any L columns of the parity check matrix H are linearly independent.

Lemma 1 *For every n, k, L, q, an (n, k, L, q)-pseudolinear family is an $(n, k)_q$ L-wise independent family of codes.*

[3] The space of random $[n, k]_q$ linear codes has the desired succinctness properties, but however is in general not even 3-wise independent (it is 2-wise (or pairwise) independent, though). This is because for any linear map $E : [q]^k \to [q]^n$ and every $x, y \in [q]^k$, $E(x + y) = E(x) + E(y)$.

We next define the notion of an infinite family of (L, q)-pseudolinear codes of increasing blocklength. Since we are interested in the asymptotic performance of codes, we will be interested in such code families of a certain rate.

Definition 5 *An infinite family of (L, q)*-pseudolinear codes $\mathcal{C}_{L,q}$ *is obtained by picking codes $\{C_{A_i}\}_{i \ge 1}$ of blocklengths n_i (with $n_i \to \infty$ as $i \to \infty$) where C_{A_i} belongs to the (n_i, k_i, L, q)-pseudolinear family.*

3.2 Existence results for certain list-decodable codes

Lemma 2 *For every prime power $q \ge 2$, every integer ℓ, $1 \le \ell \le q$ and $L > \ell$, and every α, $0 < \alpha < 1$, there exists an infinite family of (L, q)-pseudolinear codes of rate r given by $r = \frac{1}{\lg q}\Big(\alpha \lg(q/\ell) - H(\alpha) - H(\ell/q) \cdot (q/L)\Big)$, such that every code in the family is (α, ℓ, L)-list recoverable. (Here, as elsewhere, for $0 \le x \le 1$, $H(x) = -x \lg x - (1-x)\lg(1-x)$ denotes the binary entropy function of x).*

Proof: The proof follows by an application of the probabilistic method and is omitted. □

3.3 "Derandomizing" constructions of pseudolinear codes

The above probabilistic constructions can be derandomized using the method of conditional probabilities. Details are omitted. For our purposes, the necessary result is stated below (obtained by setting $\ell = 1/\varepsilon$, $q = O(1/\varepsilon^2)$ and $L = O(1/\varepsilon)$ in above probabilistic result).

Lemma 3 *For every $\alpha > 0$, there exist positive constants a_α, b_α such that for every $\varepsilon > 0$, there exist $q = O(1/\varepsilon^2)$, $L = a_\alpha/\varepsilon$ and a family PL_ε of (L, q)-pseudolinear codes of rate b_α, such that a code of blocklength n in the family is (a) constructible in deterministic time $q^{O(n\varepsilon^{-1})} = 2^{O(n\varepsilon^{-1}\log(1/\varepsilon))}$ or probabilistically in $O(n^2\varepsilon^{-1}\log(1/\varepsilon))$ time, (b) can be represented in $O(n^2\varepsilon^{-1}\log(1/\varepsilon))$ space, and encoded in $O(n^2 \log q \log^{O(1)} \log q)$ time, and (c) is $(\alpha, 1/\varepsilon, L)$-list-recoverable.*

Remark: Even though the above results are stated for code families over a fixed constant-sized alphabet, a variant of it holds equally well also for alphabet size that grows with the length of the code (in some sense the large alphabet only "helps" these results). This is used in our multilevel concatenated code constructions.

4 Expander based list-decodable code constructions

We need the following basic definition about expanders.

Definition 6 *For integers $N, d \geq 1$ and $0 < \varepsilon, \alpha < 1$, an $(N, d, \alpha, \varepsilon)$-expander is a d-regular $N \times N$ bipartite graph $H = (A, B, E)$ (where A, B with $|A| = |B| = N$ are the two sets in the bipartition and E is the edge set) with the property that given any subset $X \subseteq B$ with $|X| \geq \varepsilon|B|$, the number of vertices in A with some neighbor in X is at least $\alpha|A|$.*

The following result on existence of expanders is well known; see for instance [1, Sec. II] where an explicit construction using the Ramanujan graphs of [13] is discussed.

Fact 1 *There is a constant c such that for every $\varepsilon > 0$ and for infinitely many n, there exists an explicitly constructible $(n, c/\varepsilon, 1/2, \varepsilon)$-expander.*

4.1 The "basic" code construction using expanders

We now present our code construction (number 1) which has rate ε^2 and is list decodable in quadratic time from up to $(1-\varepsilon)$ fraction errors. The formal result is stated below.

Theorem 4 *For all $\varepsilon > 0$, there exists a code family with the following properties:*

(i) It has rate $\Omega(\varepsilon^2)$ and is defined over an alphabet of size $2^{O(\varepsilon^{-1}\log(\varepsilon^{-1}))}$.

(ii) A description of a code of blocklength N in the family can be constructed in deterministic $N^{O(\varepsilon^{-1}\log(1/\varepsilon))}$ time or probabilistically in $O(\varepsilon^{-1}\log^2 N \log(1/\varepsilon))$ time. Such a code can be list decoded in $O(N^2)$ time using lists of size $O(1/\varepsilon)$ from up to $(1-\varepsilon)N$ errors.

For reasons of space, we only sketch the construction idea. The basic idea behind the construction is to use as "left" code C a concatenated code based on an outer Reed-Solomon code C_{RS} of rate $\Theta(\varepsilon)$ and an inner code C_{in} as guaranteed in Lemma 3. The final code C^* is then constructed by redistributing the symbols of a codeword of C using a degree $O(1/\varepsilon)$ expander G at the final stage. To encode a message x, we first encode it by C, and then push each symbol of $C(x)$ along the edges of G and form $C^*(x)$. To perform decoding, given a string z that agrees with $C^*(x)$ in at least ε fraction of positions, each symbol of C^* "votes" for every symbol of C that it receives a symbol from. By the expansion property, at least, say $1/2$, fraction of positions of C receive a list of size $O(1/\varepsilon)$ that includes the "correct" symbol $C(x)$. We can now use the list-recoverability properties of C to complete the decoding and output x.

4.2 Constructions using multilevel concatenated schemes

We now introduce a code construction where an outer Reed-Solomon code is concatenated with multiple levels of inner codes (as guaranteed by Lemma 2, albeit over large, growing sized alphabets which decrease in size from the outermost to innermost levels) followed by a redistribution of symbols via a expander (this part is as in the earlier construction). By using constant rate Reed-Solomon codes over very large fields at the outermost level followed by concatenation by these multilevel codes, allows us improve the rate in Theorem 4 by an ε factor at the expense of the decoding time becoming sub-exponential in the blocklength. Thus these give the best rate codes that have list decoding algorithms of reasonable complexity for up to $(1-\varepsilon)$ fraction of errors.

4.2.1 The basic multilevel scheme

We now describe these results in further detail. We first describe the construction of these multilevel codes and their properties.

Lemma 5 *For every $p \geq 1$ and every $\varepsilon > 0$, there exist a code family with the following properties:*

(i) (Rate and alphabet size) The family has rate $2^{-O(p^2)}$ and is defined over an alphabet of size $O(1/\varepsilon^2)$.

(ii) (List decodability property) Each member of the code family is $(\frac{1}{2}, \frac{1}{\varepsilon}, \frac{2^{O(p^2)}}{\varepsilon})$-list recoverable. Furthermore such list decoding can be accomplished in $2^{O(N^{1/p}\log(1/\varepsilon))}$ time, where N is the blocklength of the concerned code.

(iii) (Constructibility) A code of blocklength N in the family can be constructed in probabilistic $O(N^2\log(1/\varepsilon))$ time, or deterministically in $2^{O(N\varepsilon^{-1}\log(1/\varepsilon))}$ time. Also, encoding can be performed in $O(N^2\log^{O(1)}(1/\varepsilon))$ time.

Comparison with Lemma 3: Note that the above lemma is similar to Lemma 3 in terms of the parameters of the codes it guarantees. But the important difference is that they come equipped with a sub-exponential time list decoding algorithm (Property (ii) above). The proof of the above lemma is omitted.

4.2.2 Codes of rate $\Omega(\varepsilon)$ with sub-exponential list decoding for $(1-\varepsilon)$ fraction errors

We now improve the rate of the codes from Theorem 4 using the multilevel concatenated codes we introduced in the previous section, at the cost of sub-exponential (as opposed to polynomial) decoding time.

Theorem 6 *For every constant $\gamma > 0$ the following holds: for all $\varepsilon > 0$, there exists a code family with the following properties:*

- *(i) (Rate and alphabet size) The code has rate $\Omega(\varepsilon 2^{-O(\gamma^{-2})})$ and is defined over an alphabet of size $2^{O(\varepsilon^{-1}\log(1/\varepsilon))}$.*
- *(ii) (Construction complexity) A description of a code of blocklength N in the family can be constructed in probabilistic $O(N^{2-2\gamma}\log(1/\varepsilon))$ time, or deterministically in time $2^{O(N^{1-\gamma}\varepsilon^{-1}\log(1/\varepsilon))}$. Moreover the code can be encoded in $O(N^{2(1-\gamma)}\log^2 N \log^{O(1)}(1/\varepsilon))$ time.*
- *(iii) (List decodability) The code can be list decoded in $2^{O(N^{\gamma}\log(1/\varepsilon))}$ time from up to $(1-\varepsilon)N$ errors.*

4.2.3 Binary codes of rate $\Omega(\varepsilon^3)$ with sub-exponential list decoding up to $(1/2-\varepsilon)$

We now use the code construction from the previous section with a final level concatenation with a binary code to get constructions of good list-decodable binary codes. Our result is stated formally below – we omit the proof.

Theorem 7 *For every constant $\gamma > 0$ the following holds: for all $\varepsilon > 0$, there exists a* binary *code family with the following properties:*

- *(i) (Rate) It has rate $\Omega(\varepsilon^3 2^{-O(\gamma^{-2})})$.*
- *(ii) (Construction Complexity) A description of a code of blocklength N in the family can be constructed in probabilistic $O((N^{2(1-\gamma)}+\varepsilon^{-6})\log(1/\varepsilon))$ time or deterministically in time $\left(2^{O(N^{1-\gamma}\varepsilon^{-3}\log(1/\varepsilon))}+2^{O(\varepsilon^{-3}\log(1/\varepsilon))}\right)$. The code can be encoded in $O(N^{2(1-\gamma)}\log^2 N\log^{O(1)}(1/\varepsilon))$ time.*
- *(iii) (List decodability) The code can be list decoded from up to $(1/2-\varepsilon)N$ errors in $2^{O(N^{\gamma}\log(1/\varepsilon))}$ time.*

5 Expander-based code constructions for unique decoding

We next turn to the constructions of codes with large minimum distance and very efficient unique decoding algorithms for up to half the distance. We obtain constructions both over large alphabets (where our aim is to decode $(1/2-\varepsilon)$ fraction errors, and binary codes (where our aim is to decode $(1/4-\varepsilon)$ fraction errors). We give two classes of constructions which have similar structure. The first one achieves the best known rates together with near-linear time decoding (actually it can perform list decoding beyond half the distance). The second one achieves linear decoding time, but works only for unique decoding and achieves a slightly worse rate.

5.1 Near-linear time decodable codes

In this section we show another expander based construction of codes that can be unique decoded from a large fraction of errors. The construction is almost identical to that given in [1]. Our modification allows us to make them list-decodable in $O(n^2)$ time from a large (but constant) fraction of errors. This in particular enables us to uniquely decode these codes up to half of minimum distance, which solves the main open problem posed by [1]. By complicating the construction we can further reduce the time to $O(n^{1+\beta})$ for any constant $\beta > 0$.

The codes come in two versions: binary and over a large alphabet. The latter ones have relative distance $(1-\varepsilon)$, rate $\Omega(\varepsilon)$ and alphabet size $2^{O(1/\varepsilon)}$. The binary codes have relative distance $(1/2-\varepsilon)$ and rate $\Omega(\varepsilon^3)$.

We start the description from the codes over large alphabet and use it later to construct the binary codes. We need the following combinatorial objects:

1. An $[n,k]_q$ linear code C with minimum distance at least (say) $n/2$ such that C is list decodable from up to $(1-\alpha)$ fraction errors and has constant rate r_α. (Here α, q are *fixed* constants that do not depend on n or ε.)
2. A bipartite regular graph $G=(A,B,E)$ with left degree Δ, $|A|=n=|B|$, such that
   - (i) for $X\subset A$, $|X|=n/2$, we have $|\Gamma(X)|\geq(1-\varepsilon)|B|$ where $\Gamma(X)\subseteq B$ is the neighborhood of X (i.e., the graph has good "vertex-expansion")
   - (ii) for $X\subset A$, $|X|=n/2$, and $Y\subset B$, $|Y|=\delta|B|$, we have $\frac{|E(X:Y)|}{\Delta|X|}\geq\delta'$ for some δ' (here $E(X:Y)$ are the edges in the $X:Y$ cut).

We specify in turn how the above objects can be constructed. A code C satisfying above requirements exists and can be found in polynomial time, and can be list decoded from up to $(1-\alpha)$ fraction errors in $O(n^2)$ time. For example, one can use the constructions from [10] which concatenate an outer Reed-Solomon code with an appropriate inner code over a large, constant-sized alphabet. To speed up the decoding time further, one can use Reed-Solomon codes over exponentially large alphabets in a multilevel concatenated scheme (similar to those in Section 4.2) as the outer code and obtain near-linear time decoding. The result is formally stated below. We omit the proof, but we would like to mention that owing to the huge field sizes over which some Reed-Solomon codes in multilevel concatenation scheme are defined, it is crucial for us to use a *strongly polynomial time* list decoding algorithm for Reed-Solomon codes. Such an implementation of Sudan's list decoding algorithm [17] was given by Augot and Pecquet [2].

Claim 1 *For every $\alpha > 0$ there exists a prime power $q = q_\alpha$, which may be assumed to be a power of two, such that for all $\beta > 0$ the following holds. There is an* explicitly specified *code family with constant rate $r_{\alpha,\beta} > 0$ over an alphabet of size q with the property that a code of blocklength n in the family can be list decoded from up to $(1-\alpha)$ fraction errors in $O(n^{1+\beta})$ time, and can be encoded in $O(n \log^{O(1)} n)$ time.*

Finally, we mentioned that a graph with the properties as in (2) above exists and is explicitly constructible. In particular, we can use the Ramanujan graphs of [13], which are known to satisfy the above properties while having degree $\Delta = O(1/\varepsilon + 1/\delta) = O(1/\varepsilon)$.

5.1.1 Codes with rate $\Omega(\varepsilon)$, relative distance $(1-\varepsilon)$ and alphabet size $2^{O(1/\varepsilon)}$

Theorem 8 *For any $\beta, \delta > 0$ there is a constant $B > 1$ such that for all small enough $\varepsilon > 0$ there is an explicitly specified code family with the properties:*

(i) It has rate ε/B, relative distance at least $(1-\varepsilon)$ and alphabet size $2^{O(1/\varepsilon)}$.

(ii) A code of blocklength n in the family can be list decoded in $O(n^{1+\beta})$ time from up to a $(1-\delta)$ errors, and can be encoded in $O(n \log^{O(1)} n)$ time.

Proof: We will construct a $(n,k)_Q$ code C_1 with $Q = q^\Delta$ from the code C from (1) above by using the degree Δ expander G to redistribute symbols, similar to the construction in [1]. Formally, to encode a message x, we first encode x by by C, and then we push copies of each symbol of $C(x)$ along the edges of G and form $C_1(x)$. From the expansion property of G and that C has distance at least $n/2$, it is easy to see that C_1 has distance at least $(1-\varepsilon)n$. Since the rate of C is a constant (independent of ε) and $\Delta = O(1/\varepsilon)$, it is clear that the rate of C_1 is $\Omega(\varepsilon)$. This proves Part (i) of the theorem.

The decoding algorithm that corrects $(1-\delta)n$ errors works as follows. Let z be some message in $[Q]^n$ and x be any message such that $\mathsf{dist}(C_1(x), z) \leq (1-\delta)n$. Let Y be the set positions on which z agrees with some $C_1(x)$; therefore $|Y| \geq \delta n$. The algorithm tries to recover the codeword (from C) corresponding to z. This is done by constructing, for each $i = 1 \ldots n$ (i.e., on the left side of the graph G), a list $L(i)$ of t most popular symbols among a's stored in z on positions $j : (i,j) \in E$. The crucial point is that if $C(x)_{|i} \notin L(i)$, then the fraction of edges $(i,j), j \in Y$ is at most $1/(1+t)$. Let X be the set of i's such that $C(x)_{|i} \notin L(i)$. It follows that $\frac{|E(X:Y)|}{\Delta|X|} \leq 1/(1+t)$. Thus if we pick t large enough so that $1/(1+t) < \delta'$, then by Property 2(ii) of G, we must have $|X| < n/2$. So, we can pick one symbol from each $L(i)$ at random and we get a string s with expected fractional distance from $C(x)$ at most $1/2 + 1/2 \cdot (1-1/t) = 1 - 1/2t$. So if we pick $t > 1/2\alpha$, then $\mathsf{dist}(C(x), s) \leq (1-\alpha)n$, and we can decode x successfully using the list decoding algorithm for C. The running time of this procedure is $O(n/\varepsilon)$ plus the time to decode C, which is $O(n^{1+\beta})$, for a total of $O(n^{1+\beta})$ time. □

Corollary 9 *For every $\varepsilon > 0$, there exists a rate $\Omega(\varepsilon)$ code family over alphabet of size $2^{O(1/\varepsilon)}$ with relative distance at least $(1-\varepsilon)$ that can be uniquely decoded from $(1-\varepsilon)/2$ fraction errors in near-linear time.*

Note that the novel aspect in this construction is that even to perform unique decoding of the final code C_1 we need list-decodability of C.

5.1.2 Binary codes of rate $\Omega(\varepsilon^3)$, relative distance $1/2-\varepsilon$

We now concatenate the code C_1 from Theorem 8 with a binary code to get binary codes of rate $\Omega(\varepsilon^3)$ and relative distance at least $(1/2-\varepsilon)$ and that have efficient algorithms to decode up to half the minimum distance. In fact, we will be able to list decode the codes up to an η fraction of errors for any fixed $\eta < 1/2$. The details of the proof are omitted.

Theorem 10 *For any $\beta > 0$ and any η, $0 < \eta < 1/2$, there is a constant $B > 1$ such that for all small enough $\varepsilon > 0$ there is a binary code family with rate ε^3/B and relative distance at least $(1/2-\varepsilon)$, such that a code of blocklength N from the family can be list decoded from up to an η fraction of errors in $O(N^{1+\beta} + 2^{O(1/\varepsilon^3)})$ time, and encoded in $O(N \log^{O(1)} N)$ time. The code can be constructed in probabilistic $O(1/\varepsilon^3)$ or deterministic $2^{O(1/\varepsilon^3)}$ time.*

5.2 Linear-time codes for unique decoding

In this section we present another version of the above constructions, which will enable us to achieve linear encoding and decoding times.

5.2.1 Codes with rate $\Omega(\varepsilon^2)$ uniquely decodable up to $(1/2-\varepsilon)$ errors in linear time

Theorem 11 *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* $(1/2-\varepsilon)$ *fraction 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 decodable from $\gamma > 0$ fraction of errors in linear time (here γ is an absolute constant). An explicit construction of such a code is known. [19].
2. A graph $G = (A, B, E)$ with $|A| = n$, $|B| = \gamma n$, and left degree Δ, such that:
   (a) for every set $X \subset A$ with $|X| \geq \gamma n$, if Y is the set of neighbors of X in G, then $|Y| \geq (1-\varepsilon)|B|$.
   (b) for every set $Y \subset B$ with $|Y| \geq (1/2+\varepsilon)|B|$, the set $X \subset A$, having at least as many neighbors in $B - Y$ as in Y, has size at most γn.

The latter graph with $\Delta = O(\frac{1}{\gamma\varepsilon^2})$ can be obtained from an Ramanujan graph (i.e. an expander with second largest eigenvalue $\lambda = O(1/\sqrt{\Delta})$). It is folklore that such a graph has the following "mixing" property: For every pair of subsets $X \subseteq A$ and $Y \subseteq B$, we have

$$|\frac{E(X:Y)}{\Delta|X|} - \frac{|Y|}{|B|}| \leq \lambda\sqrt{\frac{|Y|}{|X|}} . \quad (1)$$

It is easy to check that with $\lambda = O(1/\sqrt{\Delta}) = O(\sqrt{\gamma}\varepsilon)$, the above property implies both (a) and (b) above.

Given the code C and graph G, our code (call it C') is constructed in the same way as the code from the previous subsection. Since C has constant rate, clearly C' has $\Omega(\varepsilon^2)$ rate. As before the expansion property of G implies that C' has relative distance at least $(1-\varepsilon)$.

Clearly, the encoding time for C' is the same as for C, (i.e., linear) plus $O(n/\varepsilon^2)$. In order to decode a received word z which differs from a codeword $C'(x)$ in at most a $(1/2-\varepsilon)$ fraction of positions, 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 $C(x)$. 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 $O(n)$ for the decoding of C, for a total of $O(n/\varepsilon^2)$ time. □

5.2.2 Linear-time binary codes with rate $\Omega(\varepsilon^4)$ and decodable up to $(1/4-\varepsilon)$ 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 $(1/4-\varepsilon)$ fraction errors.

Theorem 12 *For any* $\varepsilon > 0$ *there is a binary code family with rate* $\Omega(\varepsilon^4)$ *and relative distance at least* $(1/2-\varepsilon)$*, such that a code of blocklength* N *from the family can be uniquely decoded from up to* $(1/2-\varepsilon)/2$ *fraction 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 11 with a suitable binary code. Details follow. Let C' be the code from the previous section, and assume its alphabet size is $Q = 2^{O(1/\varepsilon^2)}$. Let C_3 be any $[O(\log Q/\varepsilon^2), \log Q]_2$ linear code with minimum distance $1/2-\varepsilon$. (Such a code can be constructed by a picking random linear code in probabilistic $O(1/\varepsilon^4)$ time or by brute-force search in $2^{O(1/\varepsilon^4)}$ time.) We concatenate C' with C_3 obtaining a code (say C'') with rate $\Omega(\varepsilon^4)$ and relative distance $\rho = (1-\varepsilon)(1/2-\varepsilon)$. Since C' can be encoded in $O(n/\varepsilon^2)$ time, so can C'' (since the encoding by C_3 can be done in $1/\varepsilon^2$ time using a look-up table building which takes a one-time cost of $2^{O(1/\varepsilon^2)}$ time and space). The blocklength of C'' equals $N = O(n/\varepsilon^4)$, and hence the claimed encoding time holds.

It remains to show how to decode C'' from $\rho/2$ fraction of errors in linear-time. Since $\varepsilon > 0$ is arbitrary, this will imply the claimed result. To this end, we use the GMD algorithm of [6]. The algorithm decodes from $\rho/2$ errors under the following assumptions:

- the inner code can be decoded from $(1/2-\varepsilon)/2$ fraction of errors
- the outer code can be decoded from s fraction of erasures and e fraction of errors, when $s + 2e \leq (1-\varepsilon)$

The running time of the whole procedure is equal to $O(MT_{\text{in}} + mT_{\text{out}})$, where M is the length of the outer code, T_{in} is the cost of the inner decoder, $m = O(1/\varepsilon^4)$ is the length of the inner code and T_{out} is the cost of the outer decoder. In the following we show that $T_{out} = O(n/\varepsilon^2)$. Each inner decoding can be done in $O(1/\varepsilon^4)$ time using table look-up and building the table takes $2^{O(1/\varepsilon^4)}$ time. Therefore the overall decoding time equals $O(n/\varepsilon^6 + 2^{O(1/\varepsilon^4)}) = O(N/\varepsilon^2 + 2^{O(1/\varepsilon^4)})$.

It remains to show that the outer code can be decoded from s fraction of erasures and e fraction of errors ($2e + s \leq (1-\varepsilon)$) in linear time. We assume that the degree Δ of the Ramanujan graph $G = (A, B, E)$ equals $\frac{c}{\gamma\varepsilon^2}$ for a large enough constant c so that Equation (1) implies that for any $X \subseteq A$ with $|X| \geq \gamma n/2$ and any $Y \subseteq B$, the following holds:

$$|\frac{E(X:Y)}{\Delta|X|} - \frac{|Y|}{|B|}| \leq \varepsilon/3 \ . \qquad (2)$$

Let S be the set of erasures in the received message, and let E be the set of errors. The decoding algorithm proceeds by the majority voting as before, except that the "votes" coming from the set S are not counted. In order to prove the correctness, we would like to prove all but γ fraction 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 can correct up to a γ fraction of errors).

To prove this, define $X \subset A$ to be the set of nodes which have at most a fraction of $(1 - s - \varepsilon/3)$ neighbors in the set $B \setminus S$. Also, define $X' \subset A$ to be the set of nodes which have at least a fraction of $(e + \varepsilon/3)$ neighbors in E. By the Property (2) of G it follows that $|X|, |X'| \leq \gamma n/2$. Consider any node from $A \setminus (X \cup X')$. It has less than $(e + \varepsilon/3)$ fraction of neighbors in E. Moreover, it has at least $(1 - s - \varepsilon/3) - (e + \varepsilon/3) = (1 - s - e - 2\varepsilon/3)$ fraction of neighbors in $B \setminus (S \cup E)$ (i.e., in the set of correct positions). Since $2e + s \leq (1 - \varepsilon)$, it follows that $(e + \varepsilon/3) \leq (1 - s - e - 2\varepsilon/3)$. Therefore, the voting procedure works correctly for all except γ fraction of codeword positions of C, as we desired to show. $\square$

Acknowledgments

We thank Madhu Sudan for his encouragement and useful comments on the paper, and for the pointer to the strongly polynomial time Reed-Solomon list decoding algorithm in [2].

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] D. Augot and L. Pecquet. A Hensel lifting to replace factorization in list decoding of algebraic-geometric and Reed-Solomon codes. *IEEE Transactions on Information Theory*, 46:2605–2613, November 2000.

[3] A. Barg and G. Zémor. Linear-time decodable, capacity achieving binary codes with exponentially falling error probability. *IEEE Transactions on Information Theory*, to appear.

[4] P. Elias. List decoding for noisy channels. *Wescon Convention Record*, Part 2, Institute of Radio Engineers (now IEEE), pp. 94-104, 1957.

[5] G. D. Forney. *Concatenated Codes*. MIT Press, Cambridge, MA, 1966.

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

[7] A. Garcia and H. Stichtenoth. A tower of Artin-Schreier extensions of function fields attaining the Drinfeld-Vladut bound. *Inventiones Mathematicae*, 121 (1995), pp. 211-222.

[8] V. Guruswami, J. Håstad, M. Sudan and D. Zuckerman. Combinatorial bounds for list decoding. *Proceedings of the Annual Allerton Conference on Communication, Control and Computing*, Monticello, Illinois, October 2000, pp. 603-612.

[9] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and Algebraic-geometric codes. *IEEE Trans. on Information Theory*, 45 (1999), pp. 1757-1767. Preliminary version appeared in *Proc. of FOCS'98*.

[10] V. Guruswami and M. Sudan. List decoding algorithms for certain concatenated codes. *Proceedings of the 32nd ACM Symposium on the Theory of Computing (STOC)*, Portland, OR, May 2000, pp. 181-190.

[11] V. Guruswami and M. Sudan. On representations of algebraic-geometric codes. *IEEE Transactions on Information Theory*, 2001, to appear.

[12] K. Shum, I. Aleshnikov, P. V. Kumar, H. Stichtenoth and V. Deolalikar. A low-complexity algorithm for the construction of algebraic-geometric codes better than the Gilbert-Varshamov bound. *Preprint*, January 2001.

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

[14] R. Roth and G. Ruckenstein. Efficient decoding of Reed-Solomon codes beyond half the minimum distance. *IEEE Transactions on Information Theory*, Vol 46, No 1, pp. 246-257, January 2000.

[15] 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.

[16] M. A. Shokrollahi and H. Wasserman. List decoding of algebraic-geometric codes. *IEEE Trans. on Information Theory*, Vol. 45, No. 2, pp. 432-437, March 1999.

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

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

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

[20] M. A. Tsfasman, S. G. Vlădut and T. Zink. Modular curves, Shimura curves, and codes better than the Varshamov-Gilbert bound. *Math. Nachrichten*, 109:21-28, 1982.

[21] J. M. Wozencraft. List Decoding. *Quarterly Progress Report*, Research Laboratory of Electronics, MIT, Vol. 48 (1958), pp. 90-95.

[22] G. Zémor. On Expander Codes. *IEEE Transactions on Information Theory*, to appear.

[23] V. V. Zyablov and M. S. Pinsker. List cascaded decoding. *Problemy Peredachi Informatsii*, Vol 17, No. 4, pp. 29-33, 1981.