11 Linear-Time Codes for Unique Decoding

11.1 Context and Introduction

The goal of this chapter is also to construct codes which can be decoded from a large, and essentially up to a "maximum" possible, fraction of errors, with a near-optimal trade-off between rate and error-correction radius. The difference is that we are now interested in unique decoding as opposed to list decoding.

The biggest selling point of the codes in this chapter will be the linear-time encoding and decoding algorithms. Spielman [176] presented asymptotically good binary codes which can be encoded in linear time and also be (unique) decoded from a small (about 10^{-6}) fraction of errors in linear time. In this chapter, we will improve this error fraction dramatically, and present binary codes that can correct a fraction $(1/4 - \varepsilon)$ of errors, for arbitrary $\varepsilon > 0$, which is the maximum possible fraction of errors from which unique decoding is possible with positive rate. This is because for unique decoding, the maximum number of errors that can be corrected is limited by half the minimum distance of the code. Since binary code families of positive rate have relative distance less than $1/2$,[1] 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).

We will not only be able to construct asymptotically good codes over a large (resp. binary) alphabet that can be unique decoded in linear time from a fraction $(1/2 - \varepsilon)$ (resp. $(1/4 - \varepsilon)$) of errors, but also construct such codes of near-optimal rate (resp. rate matching those of the best polynomial-time constructions). Specifically, for every r, $0 < r < 1$, and $\varepsilon > 0$, we will construct codes over an alphabet of fixed size depending only on ε, which are linear-time encodable and linear-time decodable from a fraction $(1 - r - \varepsilon)/2$ of errors. Since relative distance of codes of rate r is at most $(1 - r)$, this trade-off is *optimal*, and we have linear-time algorithms to go along with it! Concatenation of these codes with binary inner codes that lie on the Gilbert-

[1] This is a well-known bound in coding theory that follows for example from the "Plotkin bound", cf. [193, Section 5.2].

V. Guruswami: List Decoding of Error-Correcting Codes, LNCS 3282, pp. 283-298, 2004.

Varshamov bound yields binary codes that can be encoded in linear time and decoded up to half the *Zyablov bound* in linear time. This essentially matches the performance achieved by polynomial time decodable constructions.

All our constructions share the common thread of using expander-like graphs as a component, and there is a strong overlap in techniques between this chapter and portions of Chapter 9 (specifically, Section 9.4). The expander graphs enable the design of efficient decoding algorithms through various forms of voting procedures. The presentation in this chapter should be reasonably self-contained and should allow the reader to read and appreciate the chapter on its own. Though the results of this chapter have no direct impact for list decoding, we point out that these expander-based techniques together with more sophisticated analysis methods have led subsequently to the construction of linear time encodable and *list* decodable codes as well [83].

Most of the material in this chapter appears in the papers [81, 82], the second of which was written only after the first version of this work was submitted. Therefore, the results reported in the thesis originally submitted to MIT are weaker than the ones stated in this chapter. But the proofs of the results in [82] are not any harder and yield near-optimal bounds, so we have chosen to follow the presentation of [82].

Organization: We present the necessary background on expanders first in Section 11.2. We then present a simpler construction of codes (with weaker guarantees) that is easier to describe and follow in Section 11.3. This enables unique decoding a fraction $(1/2-\varepsilon)$ of errors with rate $\Omega(\varepsilon^2)$ (which is worse than the optimal bound of $\Omega(\varepsilon)$), and by concatenation gives binary codes of rate $\Omega(\varepsilon^4)$ to correct a fraction $(1/4-\varepsilon)$ of errors. In Section 11.4 we present our linear-time "near-MDS" codes with near-optimal trade-offs (that match the Singleton bound). Finally, linear-time binary codes are obtained by concatenation of our near-MDS codes with suitable, constant-sized, inner codes in Section 11.5.

11.2 Background on Expanders

There are several ways in which expander graphs are defined in the literature. For our application here we will also need some "isoperimetric" or "pseudorandom" 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 $\frac{1}{\Delta} \cdot \mathbf{A}$.

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 [10, Chap. 9]).

Lemma 11.1. *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*

$$t \leq \frac{\lambda^2(1-b)n}{b} \ . \tag{11.1}$$

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 11.2. *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.*

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 [10, Chap. 9]).

Lemma 11.3. *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*

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

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 $\{G_i\}_{i \geq 1}$ 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 [131] and [136].

11.3 Linear-Time Encodable and Decodable Codes: Construction I

In this section, we present a quite simple (given as starting point the Spielman code) construction of linear-time codes that enables correction up to the

maximum possible error fractions (which is $(1/2-\varepsilon)$ for codes over a large alphabet and $(1/4-\varepsilon)$ for binary codes). In the next section we will improve this construction, but the codes of this section are easier to describe and elucidate the main idea behind our approach. We would therefore recommend reading this section before reading the improved constructions in Section 11.4.

11.3.1 Codes with Rate $\Omega(\varepsilon^2)$ Decodable Up to a Fraction $(1/2-\varepsilon)$ of Errors

Theorem 11.4. *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 [176, 175].
(2) A Δ-regular bipartite graph $G = (A, B, E)$ with $|A| = |B| = n$, such that:
   (a) for every set $X \subset A$ with $|X| \geq \gamma n$, 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 $|Y| \geq (1/2+\varepsilon)n$, the set $X' \subseteq A$ defined by

$$X' = \{x \in A : \ x \text{ has as many neighbors in } B \setminus Y \text{ as in } Y\} \qquad (11.2)$$

   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 11.2 with the choice $b = \gamma$. This gives that for all $X \subseteq A$ with $|X| \geq \gamma n$, the set $Y \subseteq B$ of all nodes with neighbors in X satisfies

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

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 (11.2). 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 11.3, we know that

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

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

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

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. This is the same as the construction illustrated in Figure 9.2, with the left code being fixed to the linear-time codes due to Spielman [176].

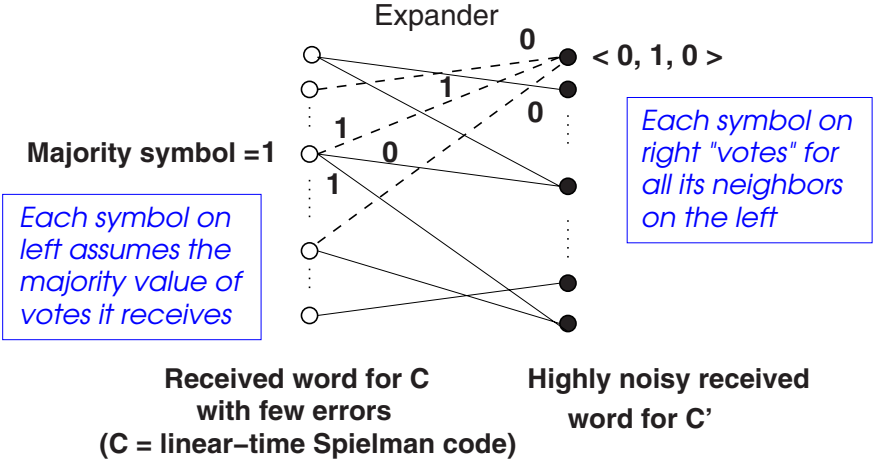

Fig. 11.1. The majority voting based decoding algorithm

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 11.1: *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. □

11.3.2 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 11.5. *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 Sketch: The code is constructed by concatenating the code from Theorem 11.4 with a suitable binary code. Let C' be the code from the Theorem 11.4.[2] 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, cf. Proposition 8.10. 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 $\delta \stackrel{\text{def}}{=} (1-\varepsilon)(1/2-\varepsilon) =$

[2] Actually, we will need to make slight changes in the assumptions about the components used in the construction of C' in Theorem 11.4, 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 11.4, 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.

$(1/2 - O(\varepsilon))$.[3] 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.

It remains to 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. This is accomplished by a general technique to decode concatenated codes called Generalized Minimum Distance (GMD) decoding due to Forney [60]. This requires a decoding algorithm for the outer code C' that can correct any combination of a fraction s of erasures and e of errors as long as $2e + s \leq (1 - \varepsilon)$. It is possible to extend the algorithm from Theorem 11.4 to have this property. We omit the details of this as well as the workings of GMD decoding now, since we will anyway discuss these in the next two sections where we give linear-time codes of near-optimal rate (specifically Theorems 11.8 and 11.10). □

11.4 Linear-Time Codes with Near-Optimal Rate

In this section, we will describe our construction of linear-time encodable/decodable codes over large alphabets which are near-MDS and match the Singleton bound. We first describe the construction over large alphabets, and will then describe how we can get binary codes by using concatenation plus GMD decoding.

11.4.1 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 11.2 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 [6, 7], and specifically the construction of near-MDS erasure codes in [7].

[3] The code C^* will be linear since C_3 is linear and it is easy to check that the construction from Theorem 11.4 gives an additive code C'.

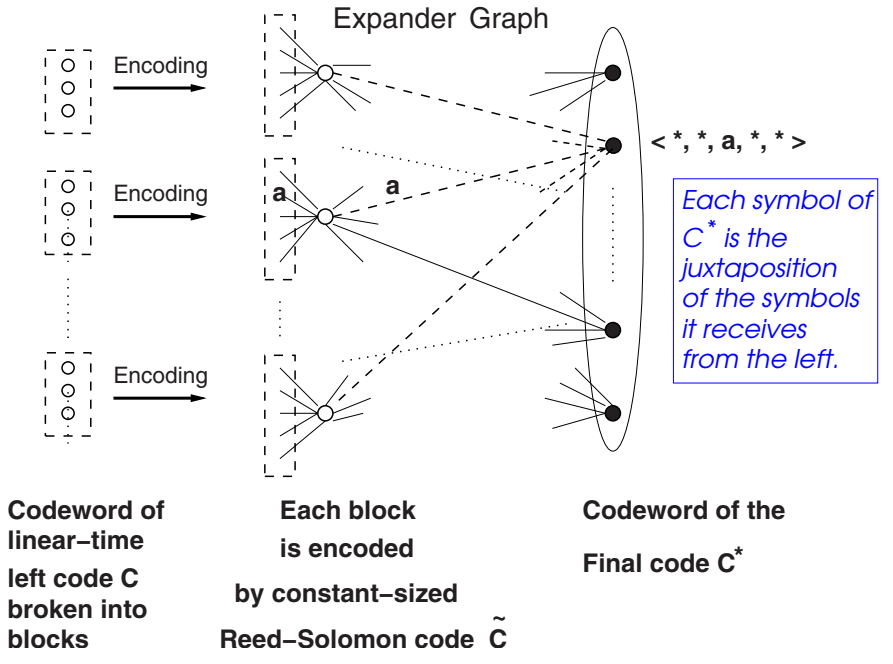

Fig. 11.2. Basic structure of the construction of near-MDS linear time codes. The "left" code is first broken into blocks and each block encoded by a constant-sized Reed-Solomon code $\tilde{C}$. Note that the second symbol **a** of the encoded block is sent to the second neighbor of the corresponding 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.

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 [176, 175] 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 [175] 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 [7] and [201]. 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 [176]); this translates into a slightly better alphabet size for our overall construction. We discuss this construction next, before moving on to the construction of the final near-MDS code.

11.4.2 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 11.2. 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 11.4.3.

Lemma 11.6. *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 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 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 [7], there is an explicit such code C_2 over 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 [201] of the analysis of Sipser and Spielman [171]. 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 [20, 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 11.7 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 $|A^{(i)}| \leq \rho n\left(\frac{\gamma'}{4} - \frac{2}{\sqrt{d}}\right)$ 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 $\beta m = \beta n d$ 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. □ *(Lemma 11.6)*

Lemma 11.7 ([201]). *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, $|T| \leq \frac{1}{2-\rho}|S|$.

11.4.3 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 11.8. *For every r, $0 < r < 1$, and all sufficiently small $\varepsilon > 0$, there exists an* explicitly *specified family of* GF(2)*-linear (also called additive)*[4] *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 11.4.1 with left code being the code from Lemma 11.6 for the choice $\gamma = \varepsilon/4$. Let x be a message of length m over 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 GF(q). We assume, by Lemma 11.6, 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 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 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 $\left|\frac{|E(X:Y)|}{|X|\Delta} - \frac{|Y|}{|B|}\right| \leq \varepsilon/4$.

[4] Recall that a code C over a field of characteristic 2 is said to 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.

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. Explicit constructions of Ramanujan graphs are known [131] and since $C, \tilde{C}$ are explicitly specified as well, our overall construction is explicit.[5] 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 GF(q)-symbols obtained from its Δ neighbors on the left, which, equivalently, can be viewed as a single symbol over GF(q^Δ). The string (of length n) consisting of these symbols forms the encoding of x by our overall code over GF(q^Δ), call it C^*. (Taking another quick look at Figure 11.2 before reading on might be useful to the reader.)

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 $\zeta \in$ GF(q^Δ), 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.

[5] Here we assume that parameters have been so picked that there is an explicit Ramanujan graph, eg. the construction of [131], 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.

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

$$e_i < e + \varepsilon/4 \quad \text{for every } i \in A \setminus (X' \cup X'') \; . \tag{11.3}$$

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

$$s_i < s + \varepsilon/4 \quad \text{for every } i \in A \setminus (X' \cup X'') \; . \tag{11.4}$$

Since $2e + s \leq (1 - r - \varepsilon)$ by hypothesis, we have, combining (11.3) and (11.4) 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. □

11.5 Linear-Time Encodable Binary 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 11.8 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 [60], 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 11.8, the overall decoding time is linear. The statement we need about GMD decoding is formally stated below — a proof appears in Appendix A.

Proposition 11.9. *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 $\tilde{D} \leq \text{dist}(C_{\text{out}})$. 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.*

Using the above result for concatenated codes with outer codes from Theorem 11.8 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.

Theorem 11.10. *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*

$$e \geq \max_{R<r<1} \frac{(1 - r - \varepsilon)H^{-1}(1 - R/r)}{2} \tag{11.5}$$

($H^{-1}(y)$ 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 (11.5) is half the Zyablov bound [202], 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 [164]. 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.

11.6 Bibliographic Notes

The simple scheme of using expanders to increase the distance of codes we used in Section 11.3 first appeared in [6]. The majority voting based decoding algorithm for such codes was given in our joint work with Indyk [81]. The basic scheme that was described in Section 11.4.1 first appeared in [7] where they used it to construct linear-time codes for recovery from erasures. The results of Theorem 11.8 and Theorem 11.10 first appeared in our joint work with Indyk [82]. This paper [82] also contained some results on list decoding that were described in Chapters 9 and 10 — the results on unique decoding alone, together with improvements that attain the Blokh-Zyablov bound as well as the Forney exponent for decoding under the binary symmetric channel, appear in a journal paper [85].

We saw in this chapter an instance of how techniques developed for list decoding are useful also for new, powerful results on unique decoding. Another instance of this is the work of Guruswami and Indyk [84] on a probabilistic construction of efficiently decodable binary linear codes that meet the Gilbert-Varshamov bound — specifically, they used a concatenation scheme with an outer list-decodable code to get binary codes on the GV bound for low rates together with a polynomial time algorithm to perform decoding up to half the distance.