Limits to List Decodability of Linear Codes*

Venkatesan Guruswami[†‡]

University of California at Berkeley
Computer Science Division
Berkeley, CA 94720.
Email: venka@cs.berkeley.edu

March, 2002

Abstract

We consider the problem of the best possible relation between the *list decodability* of a binary linear code and its *minimum distance*. We prove, under a widely-believed number-theoretic conjecture, that the classical "Johnson bound" gives, in general, the best possible relation between the list decoding radius of a code and its minimum distance. The analogous result is known to hold by a folklore random coding argument for the case of non-linear codes, but the linear case is more subtle and has remained open.

We prove our result by exhibiting an infinite family of binary linear codes of "large" minimum distance with a super-polynomial number (in block length) of codewords all within a Hamming ball of radius close to the Johnson bound. Even the existence of codes with a super-polynomial number of codewords in a ball of radius bounded away from the minimum distance (let alone radius close to the Johnson bound) was open prior to our work. We also *unconditionally* prove the "tightness" of the Johnson bound for decoding with list size that is an arbitrarily large constant.

Keywords: Error-correcting codes, linear codes, minimum distance, list decoding, Johnson bound.

Mathematics Subject Classification: 68R05, 94B65, 94B05.

*A preliminary version of this paper will appear in the Proceedings of the 34th ACM Symposium on Theory of Computing (STOC), Montréal, May 2002.

†Supported by a Miller Research Fellowship. Most of this work was done while the author was at MIT.

‡Address for Correspondence: University of California at Berkeley, Computer Science Division, 585 Soda Hall, Berkeley, CA 94720.

1 Introduction

Error-correcting codes are fundamental combinatorial constructs which are used for the purpose of reliable transmission of information over a noisy communication channel. Formally, an error-correcting code C over an alphabet Σ is an injective map $C : \Sigma^k \to \Sigma^n$ which "encodes" strings of length k (messages) into longer strings of length n — the image of C comprises the set of *codewords* of the code C. If Σ is a finite field and the map C is linear (over Σ), then the resulting code is called a *linear code*. A linear code has the advantage of succinct representation and efficient encoding, since encoding just involves a linear transformation by some $n \times k$ matrix (called a "generator matrix"). Most families of useful and widely studied codes such as Reed-Solomon, BCH, algebraic-geometric, low-density parity check codes are in fact linear codes.

An important parameter concerning a code is its *minimum distance* (or simply, *distance*). A code C is said to have distance d if every two distinct codewords differ in at least d positions. The *relative distance* of a code is the ratio of its distance to its block length, i.e. the quantity d/n. The minimum distance is a crucial parameter of a code, because large distance implies that distinct messages are mapped into codewords which are "far apart" from one another, so that even in the presence of (a few) errors, the transmitted codeword will not be confused for some other codeword, and can be "decoded" into the correct message that the sender intended to communicate. Quantitatively, this is brought out by the fact that "traditional" decoding algorithms can correct up to $(d-1)/2$ errors when using a code of distance d. This is because, when at most $(d-1)/2$ errors occurs, the received word has a *unique* codeword within distance $(d-1)/2$ from it, and this codeword must be the codeword that was transmitted. When $e \geq d/2$ errors occur, there could be more than one codeword within distance e of the received word, and thus traditional algorithms that are constrained to always output a unique codeword cannot decode beyond $d/2$ errors.

List decoding is a relaxation of unique decoding which allows the decoder to output a list of all codewords that differ from the received word in at most e positions. It thus permits recovery beyond $d/2$ errors. After a long hiatus since its introduction nearly four decades ago by Elias [2] and Wozencraft [21], list decoding has recently seen a spurt of activity, especially in the design of algorithms for list decoding important families of codes (see [6] for details and pointers about the recent work on list decoding). In order to efficiently list decode a certain code C up to, say, e errors, it is important to have an *a priori* guarantee that every Hamming ball of radius e will have only a "small" (say, polynomial in block length) number of codewords of C, as otherwise simply outputting the list of all codewords in the worst-case will take too much time. A fundamental and classical combinatorial result in coding theory, called the *Johnson bound*, provides such a guarantee and shows that in *any* code of distance d, list decoding with small lists is always possible well beyond $d/2$ errors. Much of the recent progress on algorithms for list decoding (especially their quantitative aspects) was fueled in part by this bound, since it provides a lower bound on the number of errors that one could hope to list decode and poses the challenge of matching this bound with efficient algorithms (the list decoding algorithm for Reed-Solomon codes due to Guruswami and Sudan [10] is a case in point).

The Johnson bound was originally proved by Johnson [15] in the context of constant-weight codes, and proofs of it in the context of list decoding appear, for example, in [3, 4, 12]. Let us focus on the case of binary codes. Here, the Johnson bound states that in any binary code of block length n and relative distance δ, every Hamming ball of radius $E = J(\delta)n$, where

$$J(\delta) \stackrel{\text{def}}{=} \frac{(1-\sqrt{1-2\delta})}{2} ,$$

has at most n codewords (for a slightly smaller radius, the bound on the number of codewords is at most a constant independent of n).

We call $J(\delta)$ the *Johnson bound on list decoding radius*, or simply *Johnson radius*, of a binary code. It is easily checked that $J(\delta) > \delta/2$ for $0 < \delta < 1/2$, implying that list decoding always permits recovery beyond the half-the-distance barrier. The Johnson bound plays an important role in the development of the subject of list decoding. By highlighting the fact that any code with large distance has large list decoding "potential", the Johnson bound brings to light the useful fact that one way to construct good list decodable codes is to construct codes with large minimum distance. In addition, it raises algorithmic questions concerning efficient list decoding of important families of codes beyond half the minimum distance. Such algorithms that decode up to the Johnson bound are now known for code families such as Reed-Solomon codes [10], algebraic-geometric codes [10, 17], Chinese Remainder codes [8], and Reed-Solomon concatenated with Hadamard codes [11] — the Johnson bound played a crucial motivational role in these results. Even outside contexts of list decoding, the Johnson bound is a central one in coding theory, and is for example implicitly used in the proof of the Elias-Bassalygo upper bound (the best "elementary" upper bound) on the rate of a code in terms of its relative distance.

It is therefore of considerable interest to understand if the Johnson bound is *tight*, namely whether one can improve the bound and show that Hamming balls of radius even larger than the Johnson radius always have a small number of codewords. In other words, is the Johnson radius the best possible bound on list decoding radius in terms of the minimum distance? For general non-linear codes, this was known to be the case — in fact a simple random coding argument shows that there exist codes of relative distance δ which have exponentially many codewords in a Hamming ball of (relative) radius ρ, for every $\rho > J(\delta)$ [4]. However, the situation for the case of linear codes has remained open. Since the most widely studied and used codes such as Reed-Solomon codes, algebraic-geometric codes and low-density parity check codes, are all linear, it is an important task to understand whether even with the restriction of linearity, one cannot prove a bound better than the Johnson radius for the list decoding radius.

The results of this paper demonstrate that the Johnson radius is indeed essentially tight as a general relation between list decodability and minimum distance, even for linear codes. Note that this does *not* say that for *every* linear code the Johnson bound on list decoding radius is the correct one — rather, it says there exist *some* codes for which this is the case. Indeed it is known that for *most* linear codes, the Johnson radius is *not* the best possible bound on the list decoding radius.[1]

Thus, purely as a function of the distance of the code, the Johnson radius gives (asymptotically) the best possible bound on list decoding radius. The basic strategy behind showing this is to construct a code family of certain relative minimum distance which has a large (super-constant or super-polynomial, depending upon the actual result) number of codewords within a Hamming ball of radius close to the Johnson radius. We stress that imposing the requirement of linearity makes this pursuit significantly harder, and the results presented in this paper represent the first asymptotic results that give linear code constructions with the property that there exist super-polynomial number of codewords in a ball of radius which is bounded away from the minimum distance.

1.1 Informal description of results

A formal description of our results in the form of theorem statements will be given after the relevant notation and formalism is developed in the next section. We state the results informally below. The results are stated for binary linear codes. Recall that the Johnson radius (normalized by block length) of a binary code of relative distance δ ($0 \le \delta \le 1/2$) equals $J(\delta) = \frac{1-\sqrt{1-2\delta}}{2}$. In this paper, we prove the following two results:

1. The list decoding radius for constant-sized lists approaches the Johnson radius of a code; in other words for every $0 < \delta < 1/2$, every integer L, and arbitrarily small $\varepsilon > 0$, there exist linear codes of relative distance δ with more than L codewords in a Hamming ball of relative radius $J(\delta) - \varepsilon$.
2. The list decoding radius for list size growing polynomially in the block length approaches the Johnson radius. In other words, for every δ, there exist linear codes of relative distance δ that have a super-polynomial number of codewords in some Hamming ball of relative radius close to $J(\delta)$.

The second result above is much stronger than Result 1; however, we are able to prove this result only under a number-theoretic conjecture, namely that there are infinitely many primes p for which 2 is a primitive root mod p. This conjecture is a widely believed one and is a particular case of the "Artin conjecture", which states that every square free integer $a \neq \pm 1$ is a primitive root

[1]Our construction, therefore, really finds a needle in a haystack which explains one of the reasons why such a construction was not easy to come by.

mod p for infinitely many primes p. The Artin conjecture is known to hold under the Generalized Riemann Hypothesis (GRH) [14]. In fact, several unconditional results are also known concerning the Artin conjecture. For instance, the conjecture is known to be true for "most" values of a, and in particular for one of the three numbers $2, 3, 5$ [5, 13]; see the expository paper [19] for details. This implies that we can *unconditionally* prove that the tightness of the Johnson bound for linear codes over (at least) one of the three alphabets GF(2), GF(3) and GF(5)!

For ease of presentation we state and prove our results only for the binary case, but the reader should keep in mind that the proof works for linear codes over GF(q) for any prime q, and one can also state some unconditional results for one of a small number of alphabet sizes (however, since one does not know an explicit prime for which the Artin conjecture holds, for any *particular* alphabet size, the result will only be conditional).

1.2 Related Work

The study of constant-weight codes[2]has been the subject of a lot of research, and lower bounds on the rate of constant-weight codes imply certain limits on the list decodability of codes (by considering the case when the received word is $\mathbf{0}$). Indeed, Goldreich *et al* [4] show the tightness of the Johnson bound for *general non-linear* codes following this approach. Constant-weight codes are however not linear, and the corresponding question for linear codes is a lot harder and has received attention only recently. Justesen and Hϕholdt [16] showed that the Johnson bound is tight for list decoding with constant-sized lists for certain MDS codes.[3] In fact, our proof of the tightness of Johnson bound for decoding binary linear codes with constant-sized lists (Result 1 above) is based on a "concatenated" code construction using as outer code the MDS codes of [16], though one has to perform a somewhat careful analysis of their construction, and in particular, one needs an explicit upper bound on the alphabet size of the concerned MDS codes.

Obtaining a super-polynomial (or even a fixed polynomial) number of codewords in a small Hamming ball of a linear code has seen much less success. Prior to the results of [7], no binary linear code was known that had a fixed polynomial number of codewords within a ball of relative radius even slightly smaller than the relative distance. ([7] were able to prove this statement, stated formally as Theorem 3.4; our proof uses a similar high-level approach to that of [7].) The authors of [16] are able to obtain a linear number of codewords at the Johnson radius for certain Reed-Solomon codes, but these results hold only for a certain set of relative distances δ, and not for every value of δ in the range $0 < \delta < 1$. Actually, by using this construction instead of the MDS code construction (detailed in Lemma 4.1), we can simplify our proof of Theorem 3.2 and improve

[2]Constant-weight codes are codes all of whose codewords have an identical Hamming weight.

[3]MDS (short for Maximum Distance Separable) codes are linear codes which meet the so-called Singleton bound, i.e., codes whose dimension k, minimum distance d and block length n satisfy the optimal relation $k + d = n + 1$.

the bound on number of codewords we obtain in Theorem 5.3. However, as such a result will only hold for certain values of the relative distance, we omit the precise details of this statement.

1.3 Subsequent work

Very recently, the author and Igor Shparlinski [9] observed that one can get an *unconditional* upper bound on $L^{\mathrm{poly}}(\delta)$ which is only slightly larger than $J(\delta)$ by using techniques of this paper together with the fact that there exists a very large prime q for which 2 is a primitive root modulo q^k (i.e. 2 has order $\phi(q^k) = q^{k-1}(q-1)$ modulo q^k) for every $k \geq 1$. The larger the q for which this holds, the better the upper bound on $L^{\mathrm{poly}}(\delta)$, and since a straightforward computation exbihits such primes larger than 10^{20}, the bound they get is essentially just the Johnson bound.

2 Definitions and Preliminaries

We first develop the necessary notation and definitions in order to describe our results formally.

A linear code C over the finite field $\mathbb{F}_q$ is a subspace of $\mathbb{F}_q^n$ (the n-dimensional vector space over the field $\mathbb{F}_q$), where n is called the *block length* of C. The rate of C, denoted $R(C)$, equals k/n, where k is the dimension of C.

For strings $\mathbf{x}, \mathbf{y}$, $\Delta(\mathbf{x}, \mathbf{y})$ denotes the Hamming distance between them, i.e. the number of positions where they differ. The Hamming ball of radius e around a string $\mathbf{x} \in \mathbb{F}_q^n$ is defined by: $B(\mathbf{x}, e) = \{\mathbf{y} \in \mathbb{F}_q^n : \Delta(\mathbf{x}, \mathbf{y}) \leq e\}$.

The *minimum distance*, or simply *distance*, of a code C, denoted $\mathrm{dist}(C)$, is the minimum Hamming distance between two distinct codewords of C. A q-ary code of block length n, dimension k and minimum distance d is denoted as an $[n, k, d]_q$ code. In this paper, we will state our results only for binary linear codes; we next define the basic notions concerning list decodability of binary codes.

Definition 2.1 *For positive integers e, L, a binary code $C \subseteq \mathbb{F}_2^n$ is said to be* (e, L)-list decodable *if* every *Hamming ball of radius e has at most L codewords of C, i.e. $\forall\, \mathbf{x} \in \mathbb{F}_2^n$, $|B(\mathbf{x}, e) \cap C| \leq L$. For a code C of block length n and an integer $L \geq 1$, the* list-of-L decoding radius *of C, denoted* $\mathsf{radius}(C, L)$*, is defined to be the* maximum *value of e for which C is (e, L)-list decodable.*

Definition 2.2 (List decoding radius) *For an infinite family of codes $\mathcal{C} = \{C_i\}_{i \geq 1}$ where C_i has block length n_i, and a function $\ell : \mathbb{Z}^+ \to \mathbb{Z}^+$, define the* list-of-$\ell$ decoding radius *of $\mathcal{C}$, denoted* $\mathsf{LDR}_\ell(\mathcal{C})$*, to be*

$$\mathsf{LDR}_\ell(\mathcal{C}) = \liminf_i \left\{ \frac{\mathsf{radius}(C_i, \ell(n_i))}{n_i} \right\}.$$

Definition 2.3 (Lower bound on list decoding radius) *For a distance parameter* δ*,* $0 \le \delta \le 1/2$*, and list size* $\ell : \mathbb{Z}^+ \to \mathbb{Z}^+$*, the* lower bound on list-of-ℓ decoding radius *for* binary linear *codes of relative distance* δ*, denoted* $L_\ell(\delta)$*, is defined to be*

$$L_\ell(\delta) = \inf_{\mathcal{C} \ \mid \ \delta(\mathcal{C}) \ge \delta} \mathsf{LDR}_\ell(\mathcal{C}) \ ,$$

where the infimum is taken over all binary linear *code families of relative distance at least* δ*.*

One could also define the above function $L_\ell(\delta)$ for relative distance δ which is a function of the block length. In this case, the infimum above would be taken over codes families $\mathcal{C} = \{C_i\}_{i \ge 1}$ that satisfy $\mathrm{dist}(C_i) \ge \delta(n_i) \cdot n_i$ where n_i is the block length of C_i for $i \ge 1$.

Note that the terminology "lower bound on list decoding radius" comes from the fact that $L_\ell(\delta)$ is the *maximum* fractional radius r for which *every* code of block length n and relative distance δ is guaranteed to have at most $\ell(n)$ codewords in any Hamming ball of radius rn. In other words, every code family of relative distance δ has list-of-ℓ decoding radius *at least* $L_\ell(\delta)$.

We will be interested in studying the above function for the case of list size being a constant or growing as a polynomial in the block length. To do so, we introduce further definitions.

Definition 2.4 *For real distance* $0 \le \delta < 1/2$ *and an integer constant* $c \ge 1$*:*

(i) The quantity $L_c^{\mathrm{const}}(\delta)$ *is defined to be* $L_\ell(\delta)$ *where* $\ell(n) = c$ *is a constant function.*

(ii) The quantity $L_c^{\mathrm{poly}}(\delta)$ *is defined to be* $\limsup_{c_1 \to \infty} L_{\ell_{c_1}}(\delta)$ *where* $\ell_{c_1}(n) = c_1 n^c$*.*

(iii) The quantity $L^{\mathrm{const}}(\delta)$ *is defined as* $\limsup_{c \to \infty} \{L_c^{\mathrm{const}}(\delta)\}$*.*

(iv) The quantity $L^{\mathrm{poly}}(\delta)$ *is defined as* $\limsup_{c \to \infty} \{L_c^{\mathrm{poly}}(\delta)\}$*.*

Some coding constructs

We now quickly review some of the classical codes that we will use in our constructions. Over a field $\mathbb{F}_q$ with at least n elements, a *Reed-Solomon code* of block length n is defined by the evaluations of polynomials of degree at most k at n distinct elements of the field. This gives an $[n, k+1, n-k]_q$ linear code over $\mathbb{F}_q$. The *binary Hadamard code* of dimension m maps a binary string x of length m into a string of length 2^m consisting of the dot products (over $\mathbb{F}_2$) of x with every m-bit string; it is an $[2^m, m, 2^{m-1}]_2$ linear code.

A simple yet powerful tool that we make good use of in our constructions is that of *code concatenation*. Given an $[N, K, D]_{2^m}$ linear code C_1 over $\mathrm{GF}(2^m)$ and an $[n, m, d]$ binary linear code C_2, their *concatenation* $\tilde{C} = C_1 \oplus C_2$ is a code which first encodes the message according to C_1 and then encodes each of symbols of the codeword of C_1, viewed as an m-bit string, using C_2. The concatenated code $\tilde{C}$ is then an $[Nn, Kk, \ge Dd]$ binary linear code. The codes C_1 and C_2 as above are respectively referred to as the *outer* and *inner* codes of the concatenation.

3 Formal statement of results

In the above notation, the "classical" Johnson bound on list-of-c decoding radius, see for example [12] or [6, Chapter 3], implies the following:

Proposition 3.1 *For every δ, $0 \le \delta \le 1/2$ and every integer $c \ge 1$, $L_c^{\rm const}(\delta) \ge \frac{1}{2}(1-\sqrt{1-2\delta+\frac{2\delta}{c}})$. As a corollary, we have $L^{\rm const}(\delta) \ge J(\delta) = (1-\sqrt{1-2\delta})/2$.*

Thus the above gives a lower bound on $L^{\rm const}(\delta)$. The first result is that this lower bound is tight:

Theorem 3.2 *For every δ, $0 \le \delta \le 1/2$, $L^{\rm const}(\delta) = J(\delta) = (1-\sqrt{1-2\delta})/2$.*

Given the precise understanding of the $L^{\rm const}$ function from the above theorem, we are next interested in understanding the functions $L^{\rm poly}$. Here we make the following conjecture that the Johnson bound is in fact tight even for list decoding with polynomial-sized lists.

Conjecture 1 *For every $0 < \delta < 1/2$, $L^{\rm poly}(\delta) = \frac{1}{2}\cdot(1-\sqrt{1-2\delta})$.*

We are as yet unable to settle the conjecture. But we are able to prove it assuming a widely accepted number-theoretic conjecture which is in turn implied by the Generalized Riemann Hypothesis. Hence, there is strong evidence for the truth of the conjecture. The specific result we prove is stated below:

Theorem 3.3 (Main Result) *Assume that there exist infinitely many primes p such that 2 is a primitive root mod p, i.e., 2 is a generator of the cyclic multiplicative group $\mathbb{F}_p^*$ of $\mathbb{F}_p$. Then $L^{\rm poly}(\delta) = J(\delta)$.*

The above strengthens (albeit based on a number-theoretic assumption) the following result which was proved in [7].

Theorem 3.4 ([7]) *For every integer $c \ge 1$ and every δ, $0 < \delta < 1/2$, we have $L_c^{\rm poly}(\delta) < \delta$.*

Note that $J(\delta) < \delta$ in the entire range $0 < \delta < 1/2$, and therefore, the (tight) result $L^{\rm poly}(\delta) = J(\delta)$ certainly implies the result of Theorem 3.4 above. Nevertheless, the result of Theorem 3.4 is not quite subsumed by our main result, since it is unconditional and does not rely on any unproven number-theoretic conjecture. Recall from the discussion in the introduction, however, that for one of the three alphabet sizes $2, 3, 5$, the result of Theorem 3.3 is *unconditional.*

4 Proof for constant-sized lists

In this section, we will prove Theorem 3.2. The results of this section are based on the ideas of Justesen and Hϕholdt [16], though we fill in several details in their proofs and also need some new ideas to prove our claim. In particular, Justesen and Hϕholdt state their results only for large alphabets while we are interested in results for binary codes. In fact their results hold for Maximum Distance Separable (MDS) codes which are $[N, K, D]$ linear codes whose dimension and minimum distance satisfy the optimal trade-off $K + D = N + 1$ (i.e., they match the *Singleton Bound*). Such a code is characterized by an $N \times K$ generator matrix which has the property that every $K \times K$ submatrix has rank K (cf. [18, Chap. 11]).

4.1 The Basic Construction

The following lemma is at the core of the results of this section. It is proved using the basic construction scheme in [16]. However, we need an explicit upper bound on the size of the field over which the code is defined, while [16] were content in getting MDS codes over some large enough field. Consequently, we have to be more careful in our proof, specifically when proving a certain linear algebraic claim from [16], because we need an explicit upper bound on the field size. The linear-algebraic fact needed in the proof of the following lemma is stated and proved as a separate technical lemma in Appendix A.

Lemma 4.1 *For all large enough integers m, s with $2 < s < m + 1$, and for all sufficiently large f, there exists a linear MDS code $\mathbf{C}$ over $\mathrm{GF}(2^f)$ of block length $N = \binom{m+1}{s}$, relative distance $\delta = 1 - \frac{s(s-1)}{m(m+1)}$, and dimension $(1 - \delta)N + 1$, with the property that there exists a Hamming ball of radius τN, where $\tau = 1 - s/(m + 1)$, containing at least $(m + 1)$ codewords of $\mathbf{C}$. In other words, $\mathbf{C}$ is* not *$(\tau N, m)$-list decodable. Moreover, it suffices if $f \geq \Omega(N)$ for the above claim, so that an MDS code with the stated properties exists over a field of size at most $2^{O(N)}$.*

Proof: The code construction and the configuration of $(m + 1)$ codewords that lie in a small Hamming ball will be based on a certain "combinatorial design". Since the design used is a trivial one, we simply present the actual construction without developing or using any design-theoretic notation or terminology.

Let $2 < s < m + 1$ and $N = \binom{m+1}{s}$. Let $B_1, B_2, \ldots, B_N$ be the set of N s-elements subsets of $\{0, 1, 2, \ldots, m\}$, ordered so that the $r = \binom{m}{s-1}$ sets containing 0 are the first r sets $B_1, B_2, \ldots, B_r$. For $j = 0, 1, \ldots, m$, define the N-dimensional vector $\mathbf{c_j}$ as follows: $\mathbf{c_0} = \mathbf{0}$ and for $1 \leq j \leq m$ and $1 \leq l \leq N$,

$$\mathbf{c_j}(l) = \begin{cases} 0 & \text{if } j \in B_l \text{ and } 0 \in B_l, \\ 1 & \text{if } j \in B_l \text{ and } 0 \notin B_l, \end{cases} \tag{1}$$

c_1	**0**	**0**	**0**				**1**	**1**	**1**	
c_2	**0**			**0**	**0**		**1**	**1**		**1**
c_3		**0**		**0**		**0**	**1**		**1**	**1**
c_4			**0**		**0**	**0**		**1**	**1**	**1**

Figure 1: The construction for $m = 4$, $s = 3$ and $N = 10$. We assume a lexicographic ordering of the 3-element subsets of $\{0, 1, 2, 3, 4\}$, so that $B_1 = \{0, 1, 2\}$, $B_2 = \{0, 1, 3\}$, $B_3 = \{0, 1, 4, \}$, $B_4 = \{0, 2, 3\}$, and so on.

The values of the $\mathbf{c_j}$'s in the remaining positions (not fixed by (1) above) will be suitably picked elements from $\mathrm{GF}(2^f) \setminus \{0, 1\}$, no element being used more than once. Here f is any sufficiently large integer. The exact choice of f that will suffice and the choice of elements at the remaining positions will be described shortly.

Figure 1 illustrates the construction of the $\mathbf{c_j}$'s for the case $m = 4$ and $s = 3$; in this case $N = 10$ so that each vector has 10 coordinates out of which $\binom{m}{s-1} = 6$ are filled with 0's or 1's.

Define $\delta = (1 - \frac{s(s-1)}{m(m+1)})$. Now, for any pair of elements out of $\{0, 1, \ldots, m\}$, the number of sets B_i which contain both of them is precisely $\binom{m-1}{s-2}$. Using this fact, it is follows that the Hamming weight of each $\mathbf{c_j}$, $1 \le j \le m$, as well as the Hamming distance $\Delta(\mathbf{c_i}, \mathbf{c_j})$ between $\mathbf{c_i}$ and $\mathbf{c_j}$ for $1 \le i < j \le m$, all equal

$$D \stackrel{\text{def}}{=} \binom{m+1}{s} - \binom{m-1}{s-2} = N\Big(1 - \frac{s(s-1)}{m(m+1)}\Big) = \delta N \ . \tag{2}$$

Our goal will be to realize an $[N, K, D]$ MDS code with $K = N - D + 1$ that contains all the $\mathbf{c_j}$'s, $0 \le j \le m + 1$, as codewords. Furthermore all these $(m + 1)$ codewords will lie in a Hamming ball, say Γ, of radius $E = \tau N$ where $\tau = 1 - \frac{s}{m+1}$. Together, these will imply the statement of the lemma. The latter of the above goals is easier to meet, while the former requires some work. Therefore, we first specify the center $\mathbf{x} = \langle x_1, x_2, \ldots, x_N \rangle$ of the Hamming ball Γ. This is defined as:

$$x_l = \begin{cases} 0 & \text{for } 1 \le l \le r \\ 1 & \text{for } r < l \le N. \end{cases}$$

(Recall that $r = \binom{m}{s-1} = \frac{sN}{m+1}$ is the number of B_i's that contain 0.) It is clear that Hamming distance between $\mathbf{x}$ and $\mathbf{c_0} = \mathbf{0}$ is

$$\Delta(\mathbf{x}, \mathbf{c_0}) = N - r = N\Big(1 - \frac{s}{m+1}\Big) = \tau N \ .$$

For $j \geq 1$, $\mathbf{c_j}$ has a 0-entry in $\binom{m-1}{s-2}$ of the first r entries (corresponding to s-sets that contain both 0 and j) and a 1-entry in $\binom{m-1}{s-1}$ of the last $(N-r)$ entries (corresponding to s-sets that contain j but not 0). Hence we have

$$\Delta(\mathbf{x}, \mathbf{c_j}) = N - \left(\binom{m-1}{s-2} + \binom{m-1}{s-1} \right) = \tau N \ .$$

Thus, we have all the $m+1$ "codewords" $\mathbf{c_j}$, $0 \leq j \leq m$, in a Hamming ball of radius τN.

It remains to realize the strings $\mathbf{c_1}, \ldots, \mathbf{c_m}$ as codewords in an $[N, K = N-D+1, D]$ MDS code over a sufficiently large field $\mathbb{F} = \mathrm{GF}(2^f)$. Note that $\mathbf{c_0} = \mathbf{0}$ is always a member of any linear code and hence we only have to worry about the $\mathbf{c_j}$'s, for $1 \leq j \leq m$. We have quite a bit of flexibility in filling out the entries of the $\mathbf{c_j}$'s in positions other than those already filled with 0s and 1s. We would like to show that this allows us to fill in the remaining positions with distinct entries so that the $\mathbf{c_j}$'s can be realized as codewords in an $[N, K, D]$ MDS code.

To construct this MDS code, we will construct an $N \times K$ matrix G over a large enough finite field $\mathbb{F} = \mathrm{GF}(2^f)$ of such that every $K \times K$ submatrix of G has full rank. The linear code defined by such a generator matrix G is MDS, and this will be our promised MDS code.In addition we will prove that there exists a way to fill in the entries other than $0, 1$ in $\mathbf{c_j}$, $1 \leq j \leq m$, so that the first m columns of G are $\mathbf{c_1}, \ldots, \mathbf{c_m}$. We will then have the desired MDS code which contains the $\mathbf{c_j}$'s as codewords (since the columns of G are certainly codewords of the code defined with generator matrix G).

To do this we take a two-step approach. In the first step we prove that for all large enough fields $\mathbb{F}$ ($|\mathbb{F}| = 2^{\Omega(N)}$ suffices), it is possible to fill in the missing entries of the $\mathbf{c_j}$'s using *distinct* elements from $\mathbb{F} \setminus \{0, 1\}$, such that the $N \times m$ matrix H with m columns $\mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c_m}$ has the property that every $K \times m$ submatrix has rank m. The details of this step are somewhat tedious and are explained in a separate technical "linear- algebraic" lemma (Lemma A.1), whose statement and proof we defer to Appendix A. In the second step, we show that provided the field $\mathbb{F}$ is large enough (again $|\mathbb{F}| = 2^{\Omega(N)}$ suffices), one can add a further $(K-m)$ columns to H to get a matrix G, so that every $K \times K$ submatrix of G has rank K. This matrix G will be the final matrix which is the generator matrix of the desired MDS code. The second step is easy to establish. In fact, filling the remaining $(K-m)$ columns with *random* entries from $\mathbb{F}$ works with high probability. Indeed, the number of $K \times K$ submatrices to consider is $\binom{N}{K} \leq 2^N$. Fix one such submatrix, say M. We know that its first m columns are linearly independent (this follows from the property of the matrix H guaranteed by the first step). Suppose they span a space W of dimension m. The probability that for a random choice of the remaining entries, the $(m+1)$'th column lies in W is at most $q^{-(K-m)}$ where $q = |\mathbb{F}|$. In general the probability that the i'th column does not increase the dimension of the column span is at most $q^{-(K-i+1)}$, for $m < i \leq K$. By a union bound, the probability that M has rank less than K is at most

$$\sum_{i=m+1}^{K} q^{-(K-i+1)} \leq 2/q \ ,$$

for $q \geq 2$. By a union bound over all $K \times K$ submatrices, the probability that G has some $K \times K$ submatrix of rank less than K is at most $2^N \cdot (2/q) < 1$ provided $q = |\mathbb{F}| = 2^{\Omega(N)}$. In particular, this implies that there exists an $N \times K$ matrix with entries in $\mathbb{F}$ that defines an MDS code. The proof is thus complete modulo Lemma A.1 which we state and prove in Appendix A. $\square$

Lemma 4.2 *For all $\varepsilon > 0$ and $\forall\ \delta$, $0 < \delta < 1$ there exist infinitely many N for which the following holds. There exists an $[N, K, D]_Q$ linear code C_N with $D \geq \delta N$ and $Q = 2^{O(N)}$ with the property that there exists a Hamming ball in $[Q]^N$ of radius $E = (1 - \sqrt{1-\delta-\varepsilon})N$ that has at least $\lg N$ codewords of C_N.*

Proof: Fix $\delta > 0$ and $\varepsilon > 0$. For any m such that $m > 2/\varepsilon$, it is easy to see that there exists an s, $2 < s < (m+1)$, for which

$$\delta \leq 1 - \frac{s(s-1)}{m(m+1)} \leq \delta + \varepsilon \ . \tag{3}$$

Consider the code C for this choice of m, s guaranteed by Lemma 4.1. The block length of C equals $N = \binom{m+1}{s}$. Now, since $s \leq m+1$, we have $\frac{s}{m+1} \geq \sqrt{\frac{s(s-1)}{m(m+1)}}$, and hence

$$\tau = 1 - \frac{s}{m+1} \leq 1 - \sqrt{\frac{s(s-1)}{m(m+1)}} \leq 1 - \sqrt{1-\delta-\varepsilon}$$

(the last step above follows using Equation (3)). Thus there are at least $m+1$ codewords of C in a Hamming ball of radius $(1-\sqrt{1-\delta-\varepsilon})N$. Since $N \leq 2^{m+1}$, we get at least $\lg N$ codewords in a ball of radius $(1-\sqrt{1-\delta-\varepsilon})N$, as desired. $\square$

With Lemma 4.2 in place, proving the main result of this section, namely that $L^{\mathrm{const}}(\delta) = (1 - \sqrt{1-2\delta})/2$, is fairly straightforward.

Proof of Theorem 3.2: Concatenate the $[N, K, D]_Q$ code C_N from Lemma 4.2 with the binary Hadamard code of dimension $\lg Q$ and block length Q to obtain the concatenated code $\tilde{C}$. The block length of $\tilde{C}$ is then $n = NQ = N \cdot 2^{O(N)}$. The relative distance of $\tilde{C}$ is at least $\delta' = \delta/2$.

Let $\mathbf{x} = \langle x_1, x_2, \ldots, x_N \rangle \in [Q]^N$ be such that $|B(\mathbf{x}, E) \cap C_N| \geq \lg N$, where $E = (1 - \sqrt{1-\delta-\varepsilon})N$. Define $\mathbf{y} = \mathrm{Had}(x_1) \circ \mathrm{Had}(x_2) \cdots \mathrm{Had}(x_N)$. Since every two distinct codewords of the Hadamard code differ in a fraction 1/2 of positions, it is easy to see that

$$|B(\mathbf{y}, EQ/2) \cap \tilde{C}| \geq \lg N = \Omega(\lg \lg n) \ .$$

Thus there are super-constant number of codewords in a ball of radius $EQ/2$. Now

$$\frac{EQ/2}{NQ} = \frac{E}{2N} = \frac{(1-\sqrt{1-\delta-\varepsilon})}{2} = \frac{(1-\sqrt{1-2\delta'-\varepsilon})}{2} \ ,$$

and since $\delta(\tilde{C}) \geq \delta'$, we get that there exists a code family $\mathcal{C}$ with a super-constant number of codewords in a ball of relative radius $(1 - \sqrt{1 - 2\delta(\mathcal{C}) - \varepsilon})/2$. Since $\varepsilon > 0$ was arbitrary, we get that $L^{\mathrm{const}}(\delta) = (1 - \sqrt{1 - 2\delta})/2$, as desired. $\square$

The lemma below follows from the above proof. Its rather special form might be mysterious now, but it will only be used in the code constructions of Section 5, and is not necessary for the results of this section. The main features of it that will be useful in Section 5 are that its dimension can be a multiple of any desired integer k, and that it can have several codewords within a ball of radius close to the Johnson bound on list decoding radius. The code will be used in Section 5 as an inner code in a concatenation scheme with outer Reed-Solomon code to obtain a code with super-polynomial number of codewords at close to the Johnson radius, modulo some technical conditions. The block length n in the result below can be improved by using a different code as the inner code in place of the Hadamard code; however, the bottleneck in the rate in the construction of Section 5 is elsewhere and improving the rate of the code below will not impact our final result. Hence, for ease of presentation, we choose to use the Hadamard code in the construction below.

Lemma 4.3 *For all $\varepsilon > 0$, the following holds. For all large enough integers k and every δ, $0 < \delta < 1/2$, there exist an integer $b \leq 2^{O(2^k)}$, and a binary linear code $C^{\mathrm{bin}}_{k,\delta}$ that has the following properties:*

(i) It has dimension kb, block length $n \leq 2^{O(k\sqrt{b})}$, and relative distance δ.

(ii) There exists a Hamming ball of radius $\frac{n}{2}(1-\sqrt{1-2\delta-\varepsilon})$ which contains at least 2^k codewords of $C^{\mathrm{bin}}_{k,\delta}$.

Proof Sketch: Let us use Lemma 4.1 with the choice $m = 2^k - 1$. This gives us an MDS code of block length $n_0 \leq 2^{2^k}$, relative distance 2δ and dimension $b_0 = (1 - 2\delta)n_0 + 1$ over a field $\mathrm{GF}(2^{kn_0})$ (since k is a sufficiently large integer, such a code exists over $\mathrm{GF}(2^{kn_0})$). Now, arguing as in Lemma 4.2, the code will contain at least 2^k codewords in a Hamming ball of radius $(1-\sqrt{1-2\delta-\varepsilon})n_0$. We then concatenate this code with a binary Hadamard code of dimension kn_0. The relative distance of the binary concatenated code will be δ, its dimension kb where $b \stackrel{\mathrm{def}}{=} b_0 n_0$, and its block length will be $n = n_0 2^{kn_0} \leq 2^{O(k\sqrt{b})}$ (since $b = b_0 n_0$ and $b_0 = \Theta(n_0)$). Note that $b \leq n_0^2 \leq 2^{O(2^k)}$. Furthermore, as argued in the proof of Theorem 3.2, due to the corresponding property of the MDS code, the concatenated code will contain at least 2^k codewords in a ball of radius $\frac{n}{2}(1 - \sqrt{1 - 2\delta - \varepsilon})$, as claimed. $\square$

5 Proof for super-polynomial list sizes

We now set out to prove Theorem 3.3, which states that, assuming a certain number-theoretic conjecture, the Johnson radius *is* the true bound on list decoding radius as a function of the distance of the code alone.

5.1 Proof Idea

The basic idea is to use an outer Reed-Solomon code over a finite field $\mathbb{F} = \mathrm{GF}(2^l)$ and concatenate it with an inner code that (roughly stated) maps all elements of a multiplicative subgroup S of $\mathbb{F}^*$, consisting of r'th powers for some suitable r, into some small Hamming ball. Hence all codewords of the concatenated code that correspond to evaluations of polynomials that are perfect r'th powers get mapped into a small Hamming ball. If parameters could be picked so that there are a super-polynomial number of such r'th powers, we will have the desired construction. We note that this is same high-level idea as that used in [7] (a similar construction also appears in [20] in a different setting). However, we want a super-polynomial number of codewords at exactly the Johnson radius, while [7] were content with a fixed polynomial number of codewords within some ball of radius less than the minimum distance. Therefore, we need some new ideas, and specifically a different inner code (instead of the Hadamard code used by [7]), in order to obtain our construction.

The inner code we code C_{in} will be the one guaranteed by Lemma 4.3, with a suitable choice of k in relation to the size of the subgroup S. But now this inner code is only guaranteed to map *some* 2^k messages into a small Hamming ball, and these 2^k messages need not have anything to do with the elements of the subgroup under consideration. However, here comes the crucial idea. These 2^k messages (viewed as vectors over $\mathbb{F}_2$ of suitable length) must contain a linearly independent subset T of size k. Now, if the elements of $S \subseteq \mathrm{GF}(2^l)$ are linearly independent over $\mathrm{GF}(2)$, then there must exist a invertible $\mathrm{GF}(2)$-linear map F which maps elements of S into those of T (in some arbitrary ordering of the elements of S, T).[4] Define C'_{in} to be $C_{\text{in}} \circ F$ (in other words, the encoding under C'_{in} first encodes the message by the invertible map F, and then encodes the resulting string using C_{in}). Note that since F, C_{in} are linear codes, so is C'_{in}. Now C'_{in} will map all elements of S into a small Hamming ball, since C_{in} does so for elements of T. Using C'_{in} as the inner code with outer Reed-Solomon code, we can get, after a suitable choice of parameters, a super-polynomial number of codewords of the concatenated code all of which lie in a Hamming ball of radius close to the Johnson bound.

The astute reader might be puzzled about the need for the seemingly odd number-theoretic conjecture assumed in the hypothesis of Theorem 3.3. This arises because of the need to have a multiplicative subgroup whose elements are linearly independent over $\mathrm{GF}(2)$. We do not know how to prove that such subgroups exist for infinitely many sizes without making any number-theoretic assumption. However, this fact is implied by the conjecture that there exist infinitely many primes p for which 2 is a generator of the cyclic group $\mathbb{F}_p^*$.

[4]The fact that elements of S are linearly independent over $\mathrm{GF}(2)$ is independent of the specific representation of $\mathrm{GF}(2^l)$ over $\mathrm{GF}(2)$, since this fact only depends on the additive properties of the subgroup S.

5.2 The Technical Proof

We first record the number-theoretic fact stated at the end of the previous section. The proof is based on standard algebra and we omit the details.

Lemma 5.1 *Let p be an odd prime and let 2 be a generator of the multiplicative group $\mathbb{F}_p^*$ of the finite field $\mathbb{F}_p$. Let α be the primitive p'th root of unity in the field* $\mathrm{GF}(2^{p-1})$ *(this must exist since $2^{p-1} \equiv 1 \bmod p$ by Fermat's little theorem). Consider the subgroup $(1, \alpha, \ldots, \alpha^{p-1})$ of the multiplicative group of* $\mathrm{GF}(2^{p-1})$. *Then the only non-trivial linear dependence among these p elements is: $1 + \alpha + \cdots + \alpha^{p-1} = 0$. In particular, the elements $\alpha, \alpha^2, \ldots, \alpha^{p-1}$ are linearly independent over* $\mathrm{GF}(2)$.

Corollary 5.2 *Assume that there exist infinitely many primes p for which 2 is a generator of $\mathbb{F}_p^*$. Then for each one of those infinitely many primes the following holds: For every $b \geq 1$, there exists a multiplicative subgroup S' of* $\mathrm{GF}(2^{(p-1)b}) \setminus \{0\}$ *of size p such that the only* $\mathrm{GF}(2)$*-linear dependence among elements of S' is that their sum is zero. In particular, any set of $(p-1)$ elements of the subgroup are linearly independent over* $\mathrm{GF}(2)$.

We now state and prove the main theorem of this section. The claim of Theorem 3.3 clearly follows from the result below.

Theorem 5.3 *Assume that there exist infinitely many primes p for which 2 is a generator of $\mathbb{F}_p^*$. Then, for every δ, $0 < \delta < 1/2$, and every $\varepsilon > 0$, there exist infinitely many N for which there exists a binary linear code of block length N and minimum distance at least δN that has at least $N^{\Omega(\varepsilon \log\log\log N)}$ codewords in a Hamming ball of radius $\frac{N}{2}(1 - \sqrt{1 - 2\delta - \varepsilon})$.*

Proof: The proof follows the idea outlined in Section 5.1. We just need to pick the various parameters appropriately and then carefully bound the number of codewords within a certain Hamming ball of the desired radius.

Let p be a prime such that 2 is a generator of $\mathbb{F}_p^*$. Apply Lemma 4.3 with the choice of $k = (p-1)$ and $\delta = \delta/(1-\varepsilon)$ to get a binary linear code $C' = C^{\mathrm{bin}}_{(p-1),\delta/(1-\varepsilon)}$ of relative distance at least $\delta' \stackrel{\mathrm{def}}{=} \delta/(1-\varepsilon)$. Let the dimension of C' be $(p-1)b$ and let its block length be n'; we have $b \leq 2^{O(2^p)}$ and $n' \leq 2^{O(p\sqrt{b})}$.

Let $S' = (1, \alpha, \alpha^2, \ldots, \alpha^{p-1})$ be a multiplicative subgroup of $\mathrm{GF}(2^{(p-1)b})$ consisting of the r'th powers in $\mathrm{GF}(2^{(p-1)b}) \setminus \{0\}$ where $r = \frac{2^{(p-1)b}-1}{p}$. By Corollary 5.2, the elements in $S = (\alpha, \alpha^2, \ldots, \alpha^{p-1})$ are linearly independent. By Lemma 4.3, there exist a set of at least 2^{p-1} codewords of C' in a Hamming ball B of radius $\frac{n'}{2}(1 - \sqrt{1 - 2\delta' - \varepsilon})$, centered at some $\mathbf{v} \in \mathbb{F}_2^{n'}$. Among these 2^{p-1} codewords there must exist at least $(p-1)$ non-zero codewords corresponding to a set T of $(p-1)$ linearly independent messages (in $\mathbb{F}_2^{(p-1)b}$). Owing to the linear independence of elements of both S and T, there must be an invertible linear transformation F that maps the elements of S into those of T (for some fixed ordering of the elements of S and T). Consider the linear code $C'' = C' \circ F$, i.e., encoding by F followed by the encoding C'. Since F is an invertible linear transformation, C'' is a binary linear code that has the same dimension (namely $(p-1)b$) and minimum distance as C'. Also, clearly the encodings of the elements of S as per C'' all lie in the ball B, which is of radius $\frac{n'}{2}\left(1 - \sqrt{1 - \frac{2\delta}{1-\varepsilon} - \varepsilon}\right)$.

The final code C^* we use to prove the claim of the theorem will be a Reed-Solomon code over $\mathrm{GF}(2^{(p-1)b})$ of rate ε and block length $n = 2^{(p-1)b}$, concatenated with C'' as the inner code. We first quantify the main parameters of C^*. Its block length is

$$N = n \cdot n' = 2^{(p-1)b} n' \le 2^{pb} 2^{O(p\sqrt{b})} = 2^{p \cdot 2^{O(2^p)}}$$

(since $b \le 2^{O(2^p)}$). The relative distance of C^* is at least $(1-\varepsilon) \cdot \delta' = \delta$. Since there are infinitely many choices of p by the hypothesis, we can construct such a code C^* for infinitely many block lengths N. The messages of C^* are in one-one correspondence with degree εn polynomials over $\mathrm{GF}(n)$. Also note that the block length n of the outer Reed-Solomon code satisfies $n = N^{\Omega(1)}$.

We will now establish a lower bound on the number of polynomials whose evaluations at the field elements of $\mathrm{GF}(2^{(p-1)b})$ belong to the set S for "most" of the n field elements. We will do this by considering polynomials of the form $P(x) = R(x)^r$ where R is a random degree $\varepsilon n/r$ polynomial, and estimating the probability that $P(\gamma_i) \in S$ for a large fraction of i's (here we assume $\gamma_1, \dots, \gamma_n$ are the elements of $\mathrm{GF}(2^{(p-1)b})$). This will imply that for $\mathbf{r} = \mathbf{v}^n$, i.e. the center $\mathbf{v}$ of the ball B repeated n times, once for each Reed-Solomon codeword position, a ball centered at $\mathbf{r}$ of radius "about" $\frac{N}{2}(1 - \sqrt{1-2\delta})$ will have several (about $n^{\varepsilon n/r}$) codewords, and for our choice of parameters this will yield a super-polynomial number of codewords within a ball of radius close to the Johnson radius.

Let us now perform the above-mentioned probability estimation. The r'th power of any elements of $\mathrm{GF}(n)$ always lies in $S \cup \{0,1\}$. For the choice of $P(x) = R(x)^r$ for a random R, for a fixed γ_j we have

(a) $\mathbf{Pr}[P(\gamma_j) = 0] = 1/n$;

(b) $\mathbf{Pr}[P(\gamma_j) = 1] = r/n$;

(c) $\mathbf{Pr}[P(\gamma_j) \in S] = (n-r-1)/n$.

For $1 \le j \le n$, define the indicator random variable I_j as follows: $I_j = 1$ if $P(\gamma_j) \in S$ and $I_j = 0$ otherwise. Then the random variable $Z = \sum_{j=1}^n I_j$ measures the number of positions j, $1 \le j \le n$, for which $P(\gamma_j) \in S$. By (c), we have, for P chosen randomly as above, $\mathbf{E}[Z] = n - r - 1$.

Also the I_j's are pairwise independent 0/1 random variables and hence

$$\mathrm{Var}(Z) = \sum_{j=1}^n \mathrm{Var}(I_j) = \sum_{j=1}^n \mathbf{E}[I_j](1 - \mathbf{E}[I_j]) = \frac{(n-r-1)(r+1)}{n} . \tag{4}$$

By a straightforward application of Chebyshev's inequality, it follows that

$$\mathbf{Pr}[Z \le n - r - 1 - \sqrt{n}] \le \frac{(n-r-1)(r+1)}{n^2} \le \frac{2}{p} \le \frac{1}{2} \tag{5}$$

for $p \ge 4$, where the second last step follows since $r + 1 = (n + p - 1)/p \le 2n/p$. Hence at least a fraction 1/2 of the polynomials P of the form $P(x) = R(x)^r$ satisfy $P(\gamma_j) \in S$ for at least $(n - r - \sqrt{n})$ values of j. For these polynomials, their encoding by C^* will differ from $\mathbf{r}$ in at most

$$e = (n - r - \sqrt{n})\frac{n'}{2}\Big(1 - \sqrt{1 - \frac{2\delta}{1-\varepsilon} - \varepsilon}\Big) + (r + \sqrt{n})n'$$

places. Since $n \le 2^{p \cdot 2^{O(2^p)}}$ and $r = (n-1)/p$, we have $r = o(n)$ and hence $(r + \sqrt{n})n' = o(nn') = o(N)$. Thus

$$e \le \frac{N}{2}\Big(1 - \sqrt{1 - \frac{2\delta}{1-\varepsilon} - \varepsilon}\Big) + o(N) \ . \tag{6}$$

The number of such polynomials whose encodings differ from $\mathbf{r}$ in at most e positions is at least $\frac{1}{2} \cdot n^{\varepsilon n/r} \ge \frac{1}{2} \cdot n^{\varepsilon p}$. Since $n = N^{\Omega(1)}$ and $N \le 2^{p \cdot 2^{O(2^p)}}$, we have $p = \Omega(\log\log\log N)$. The number of codewords of C^* within a Hamming distance of e from $\mathbf{r}$ is therefore $N^{\Omega(\varepsilon \log\log\log N)}$.

Since $\varepsilon > 0$ was arbitrary, from Equation (6) the result claimed in the theorem follows. □

Letting $\varepsilon \to 0$, the bound in Theorem 5.3 approaches the Johnson radius $J(\delta) = (1 - \sqrt{1-2\delta})/2$. As a corollary, therefore, we get Theorem 3.3, our main result of this section.

6 Concluding Remarks

Constructions of *linear* codes with a large number of codewords in a Hamming ball of radius even slightly less than the minimum distance has remained an elusive and difficult task. The recent work of [7] made good progress towards this question by bounding the list decoding radius for decoding with a fixed polynomial-sized list away from the minimum distance of the code. In this paper, we used some of the elements of the construction of [7] together with some new ideas to exhibit codes with a super-polynomial number of codewords at a radius close to the Johnson bound on list decoding radius. This gives a precise understanding of the best possible bound on list decoding radius of linear codes purely as a function of the distance of the code. There are some obvious open questions concerning results in this paper:

1. Prove the result of Theorem 3.3 (i.e. $L^{\text{poly}}(\delta) = (1 - \sqrt{1-2\delta})/2$) *without making any number-theoretic assumption.* A good first step would be to unconditionally prove that $L^{\text{poly}}(\delta) < \delta$ for every δ, $0 < \delta < 1/2$ (recall that it is known from [7] that $L_c^{\text{poly}}(\delta) < \delta$ for each fixed constant c).

2. Our construction only manages to get a mildly super-polynomial number of codewords in a Hamming ball of radius close to the Johnson radius; one ought to be able to improve upon this bound. Specifically, for $0 < \delta < 1/2$, does there exist a linear code of relative distance δ with an *exponential* (and not just super-polynomial) number of codewords within a Hamming ball of relative radius close to the Johnson radius $J(\delta)$? As a first step, can one have exponentially many codewords within a ball of relative radius strictly less than δ?

Acknowledgments

Many thanks to Madhu Sudan for useful discussions during the course of this work and for suggesting the use of a "matching-based" approach in the proof of the linear-algebraic Lemma A.1. I would like to thank Johan Håstad for useful discussions, and Hendrik Lenstra and Igor Shparlinski for discussions and pointers about the Artin conjecture.

References

[1] Emil Artin. *Collected Papers.* ed. S. Lang and J. T. Tate, Springer-Verlag, 1965. pp. viii-ix.

[2] Peter Elias. List decoding for noisy channels. *Technical Report 335, Research Laboratory of Electronics, MIT*, 1957.

[3] Peter Elias. Error-correcting codes for list decoding. *IEEE Transactions on Information Theory*, 37:5–12, 1991.

[4] Oded Goldreich, Ronitt Rubinfeld, and Madhu Sudan. Learning polynomials with queries: The highly noisy case. *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science*, pages 294–303, 1995.

[5] R. Gupta and M. Rum Murty. A remark on Artin's conjecture. *Invent. Math.*, **78** (1984), pp. 127–130.

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

[7] Venkatesan Guruswami, Johan Håstad, Madhu Sudan, and David Zuckerman. Combinatorial bounds for list decoding. *Proceedings of the 38th Annual Allerton Conference on Communication, Control and Computing*, pages 603–612, October 2000.

[8] Venkatesan Guruswami, Amit Sahai, and Madhu Sudan. Soft-decision decoding of Chinese Remainder codes. *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science*, pages 159–168, 2000.

[9] Venkatesan Guruswami and Igor Shparlinski. Untitled manuscript, March 2002.

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

[11] Venkatesan Guruswami and Madhu Sudan. List decoding algorithms for certain concatenated codes. *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 181–190, 2000.

[12] Venkatesan Guruswami and Madhu Sudan. Extensions to the Johnson bound. *Manuscript*, February 2001. The results also appear in [6, Chapter 3].

[13] D. R. Heath-Brown. Artin's Conjecture for Primitive Roots. *Quart. J. Math.*, **37** (1986), pp. 27-38.

[14] Christopher Hooley. On Artin's conjecture. *J. Reine Angew. Math.*, 225:209–220, 1967.

[15] Selmer M. Johnson. A new upper bound for error-correcting codes. *IEEE Transactions on Information Theory*, 8:203–207, 1962.

[16] Jϕrn Justesen and Tom Hϕholdt. Bounds on list decoding of MDS codes. *IEEE Transactions on Information Theory*, 47(4):1604–1609, May 2001.

[17] Ralf Koetter and Alexander Vardy. Algebraic soft-decision decoding of Reed-Solomon codes. *Proceedings of the 38th Annual Allerton Conference on Communication, Control and Computing*, pages 625–635, October 2000.

[18] F. J. MacWilliams and Neil J. A. Sloane. *The Theory of Error-Correcting Codes.* Elsevier/North-Holland, Amsterdam, 1981.

[19] M. Ram Murty. Artin's conjecture for primitive roots. *Math. Intelligencer*, 10(4):59-67, 1988.

[20] Amnon Ta-Shma and David Zuckerman. Extractor Codes. *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 193–199, July 2001.

[21] John M. Wozencraft. List Decoding. *Quarterly Progress Report, Research Laboratory of Electronics, MIT*, 48:90–95, 1958.

A The technical linear algebraic lemma

We now state and prove the technical linear-algebraic result used in the proof of Lemma 4.1. The proof is not difficult, but is somewhat long. Actually, we believe that the lemma should have a significantly simpler and shorter proof, and we encourage the reader to try and find an alternate proof.

Lemma A.1 *For the vectors* $\mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c_m}$ *partially defined as in Equation (1), there is a way to fill in the remaining entries using distinct field elements from* $\mathbb{F} \setminus \{0, 1\}$*, for any field* $\mathbb{F}$ *that satisfies* $|\mathbb{F}| \geq 2^{\Omega(N)}$*, such that the following holds. The* $N \times m$ *matrix* H *with columns* $\mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c_m}$ *has the property that every* $K \times m$ *submatrix of* H *has rank* m *over the field* $\mathbb{F}$.

Proof: The particular proof method used here was suggested to us by Madhu Sudan. Consider the matrix H with columns $\mathbf{c_1}, \ldots, \mathbf{c_m}$ and consider filling each of the as yet unfilled entries of the $\mathbf{c_j}$'s with a different "indeterminate". Let us label these indeterminates $x_1, x_2, \ldots, x_T$, where $T \leq mN$. We will then prove that there is a setting of the indeterminates for which each of the $K \times m$ submatrices has rank m.

Let M be any $K \times m$ submatrix of H. We will prove that there exists an $m \times m$ submatrix M' of M whose determinant is non-zero as a polynomial in the x_i's. Note that each such $m \times m$ determinant is a *multilinear* polynomial in the x_i's. Let us give this determinant polynomial a label, say $D(M)$. Once we establish this fact, we can simply consider the polynomial, say P, which is the product all such determinant polynomials $D(M)$ (over all $K \times m$ submatrices M). Any setting of distinct values to the x_i's which makes P evaluate to a non-zero element would then imply a setting of the x_i's for which every $K \times K$ submatrix has some $m \times m$ submatrix with non-zero determinant, or in other words has rank m, as desired. Now P being a product of non-zero polynomials, is itself non-zero. Furthermore, P is a product of $\binom{N}{K}$ multilinear polynomials, and hence the degree of P in each variable is at most $\binom{N}{K} \leq 2^N$. Thus provided $|\mathbb{F}| \geq 2^N + T$, one can set arbitrary distinct values to all but one indeterminate involved in P, say to all but x_1, to give a univariate polynomial $\tilde{P}$ of degree at most 2^N in x_1. $\tilde{P}$ has at most 2^N roots over $\mathbb{F}$, and since $|\mathbb{F}| \geq 2^N + T$, there must exist a value in $\mathbb{F}$ outside the roots of $\tilde{P}$ and the at most $(T-1)$ values given to the indeterminates other than x_1. Giving x_1 such a value implies that P is non-zero for such a setting of the x_i's. Thus, if $|\mathbb{F}| \geq 2^{\Omega(N)}$, there exists a setting of distinct values to the x_i's which makes P evaluate to a non-zero element.

It remains to prove that every $K \times m$ submatrix of H has a $m \times m$ submatrix whose determinant is a *non-zero* multilinear polynomial in the x_i's. Since H already has 0's and 1's filled in at some positions, this fact is somewhat tricky to prove. It would be good to recollect the details of the design construction from Lemma 4.1 and specifically the pattern of 0's and 1's that are already filled in the matrix H. In particular the following easily verified facts will be handy later on in the proof:

Fact 1 Every column of H has exactly $\binom{m-1}{s-2} = (K-1)$ zeroes, and $\binom{m-1}{s-1}$ ones.

Fact 2 No column has 0s or 1s filled in more than $\binom{m-2}{s-3}$ rows out of the rows that have 0s filled in some other column (this is because the sets corresponding to all such rows must contain three fixed elements out of $\{0, 1, \ldots, m\}$, and there are clearly $\binom{m-2}{s-3}$ such s-element subsets of $\{0, 1, \ldots, m\}$).

Fact 3 For every two columns, the number of rows in which they both have entries filled in is at most $\binom{m-1}{s-2} = K - 1$.

Now, consider a fixed $K \times m$ submatrix, say M, of H, indexed by a subset $R \subseteq [N]$ of K rows of H. We want to prove that M has a $m \times m$ submatrix M' whose determinant is a non-zero polynomial in the x_i's. Our "rough" goal is to prove that one can identify a subset $S \subset R$ of size m such that for each column j, $1 \leq j \leq m$, there is a *different* row $s_j \in S$ such that entry in the s_j'th row and j'th column is an indeterminate, say x_{i_j}. Loosely, we call such an S a "matching set". (The reason for this terminology will be clear shortly.) The determinant of the $m \times m$ submatrix defined by the rows in such a subset S is then clearly non-zero, as it contains the term $x_{i_1} x_{i_2} \cdots x_{i_m}$, which cannot be canceled by any other term.

We do not always succeed in finding such a matching set, but we *almost* do, and in any case will always be able to conclude that the concerned determinant is a non-zero polynomial. The term "matching set" should be suggestive, as we will find a good set S of rows by finding a matching between the m columns and a subset of the K rows of M in a suitably defined graph. We will prove such a matching exists by appealing to *Hall's theorem*.[5] The relevant bipartite graph, call it B, is defined as follows. It has one side associated with the m columns of H and the other side with set of rows R. There is an edge between $j \in [m]$ and $\ell \in R$, if the entry in row number ℓ and column number j is an indeterminate. We first claim the following:

Claim A. *For any subset $T \subseteq [m]$ with $|T| = p \geq 3$, the neighborhood of T in the above bipartite graph B, denoted $N_B(T)$, contains at least p elements.*

Indeed, if a row $\ell \in R$ is not adjacent to any element of T, it means that the s-element subset of $\{0, 1, \ldots, m\}$ that corresponds to row ℓ contains T as a subset. The number of such s-element subsets is at most $\binom{m+1-p}{s-p}$. Hence if $p > s$, the neighborhood of T includes all rows in R, so assume $p \leq s$. Now

$$\begin{aligned} K &= \binom{m-1}{s-2} + 1 \\ &> \frac{(m-1)(m-2)\cdots(m-p+2)}{(s-2)(s-3)\cdots(s-p+1)} \binom{m+1-p}{s-p} . \end{aligned}$$

[5] Hall's theorem is a classic graph-theoretic result that gives a necessary and sufficient condition for a bipartite graph $B = (X, Y, E)$ to have a matching that matches every vertex in X to a distinct vertex in Y (we are assuming $|X| \leq |Y|$, of course). The condition is that the neighborhood of every subset $T \subseteq X$, $N_B(T) \subseteq Y$, satisfies $|N_B(T)| \geq |T|$. This is clearly a necessary condition, and Hall's theorem states that it also sufficient. We will refer to the condition "$|N_B(T)| \geq |T|$ for all $T \subseteq X$" as *Hall's condition*.

If m is large enough, and $p \geq 3$, it is now easily verified that the number of rows in R adjacent to some element in T is at least p, as claimed. $\square$ *(Claim A)*

The above claim states that Hall's condition is satisfied for subsets of $[m]$ of size 3 or more. Hence in the case when it is also satisfied for subsets of $[m]$ of size at most 2, we will have a desired matching of size m in the graph B. The $m \times m$ submatrix of M indexed by the rows used in the matching will then have non-zero determinant.

It remains to deal with the case when Hall's condition is violated for some subset of $[m]$ of size at most 2. In such a situation we will not be able to find a matching for all m columns, but will find a matching at least $(m-2)$ columns and then carefully pick two more rows so that we can still argue that the concerned determinant is non-zero. This involves a somewhat tedious case analysis.

If Hall's condition is violated for some subset of size at most 2, then there must exist some $a \in [m]$ such that $|N_B(a)| \leq 1$. This means that in the matrix H, the column number a has either 0s or 1s already filled in at $(K-1)$ or more of the K rows of R. Assume for definiteness that it has 0s or 1s filled in the first $(K-1)$ rows of R. We distinguish two cases:

Case 1: Column a has only 0s in the first $(K-1)$ rows of R.

In this case, a must have either a 1 or an indeterminate in the K'th row. This follows from Fact 1 that we recalled about the pattern of 0s and 1s in H. Furthermore, by Fact 2, all columns other than a can have at most $\binom{m-2}{s-3}$ entries filled with a 0 or 1, among the first $(K-1)$ rows of R. For large enough m, we have

$$(K-1) - \binom{m-2}{s-3} = \binom{m-1}{s-2} - \binom{m-2}{s-3} \geq 2\ ,$$

and therefore every column in $[m] \setminus \{a\}$ has at least 2 neighbors among the first $(K-1)$ rows of R. Together with Claim A, this implies that Hall's condition holds in the subgraph of B obtained by deleting the last row of R and the column a. Therefore, there must exist a matching of the $(m-1)$ columns other than a into a set R' of $m-1$ distinct rows (among the first $K-1$ rows of R). Let $x_{i_1}, \ldots, x_{i_{m-1}}$ be the indeterminates at the $m-1$ positions of H defined by this matching. Then, since column a must have either a 1 or an indeterminate in the K'th row, the rows in R' together with the K'th row of R define an $m \times m$ submatrix whose determinant is non-zero.

Case 2: Column a has 0s and 1s filled in the first $(K-1)$ rows of R, and not all these entries are 0s.
By permuting rows if necessary, assume that a has either a 1 or an indeterminate in the K'th row of R. In this situation, we further distinguish two subcases:

Case 2.1: Each $b \in [m] \setminus \{a\}$ has at least two neighbors in R, i.e., has 0s and 1s filled in at most $(K-2)$ of the rows in R.

Once again, we claim that if this were the case, the subgraph of B obtained by deleting the column a and the K'th row of R, obeys Hall's condition. Indeed the hypothesis implies that the condition holds for sets of size one, since each $b \in [m] \setminus \{a\}$ has at least one neighbor among the first $K-1$ rows of R. For subsets T of size $p \geq 2$, we apply Claim A to $T \cup \{a\}$ to conclude that $T \cup \{a\}$ has at least $p+1$ neighbors in R, and hence T at least p neighbors among the first $(K-1)$ rows of R (since a is at best adjacent to the last row of K).

Now as in Case 1, the rows corresponding to the matching of size $(m-1)$ from this subgraph together with K'th row of R give us the necessary $m \times m$ submatrix with non-zero determinant.

Case 2.2: There exists some other element $b \in [m]$, $b \neq a$, such that column number b has 0s and 1s filled in the first $(K-1)$ rows of R.

We can assume that b also does not have all 0's in the first $(K-1)$ rows of R, as otherwise we could apply the argument of Case 1 with b instead of a.

Now, at least one of a, b must have an indeterminate in the K'th row of R. This follows from Fact 3. Assume, without loss of generality, that a has an indeterminate, say x_{a^*}, in the K'th row of R. Further, by permuting the first $(K-1)$ rows of R if necessary, we may without loss of generality that the $(K-1)$'th row of R has entry 1 in column b (this is possible since column b does not have all 0's filled in these rows).

Consider the subgraph of B obtained by deleting the last two rows of R and the columns a, b. Using Claim A, it is easily checked that this subgraph satisfies Hall's condition. Therefore it has a matching which matches each of the $(m-2)$ columns other than a, b to distinct rows where they have an indeterminate. Let these indeterminates be $x_{r_1}, x_{r_2}, \ldots, x_{r_{m-2}}$. Now consider the $m \times m$ submatrix M' indexed by the $(m-2)$ rows from the matching together with the last two rows of R. Since a has an indeterminate x_{i^*} in the last row of R, and b has a 1 in the second last row of R, it follows that the determinant of M' has a term $x_{a^*} x_{r_1} \cdots x_{r_{m-2}}$ which cannot be canceled by any other term in the expansion of the determinant of M'. Therefore the determinant of M', as a multilinear polynomial, is non-zero, as we desired to show. This completes the proof in Case 2.2 as well. $\square$