Electronic Colloquium on Computational Complexity, Report No. 7 (2010)

Two Theorems in List Decoding*

Atri Rudra Steve Uurtamo

Department of Computer Science and Engineering,
University at Buffalo, The State University of New York,
Buffalo, NY, 14620.
{atri,uurtamo}@buffalo.edu

Abstract

We prove the following results concerning the list decoding of error-correcting codes:

1. We show that for *any* code with a relative distance of δ (over a large enough alphabet), the following result holds for *random errors*: With high probability, for a $\rho \leqslant \delta - \varepsilon$ fraction of random errors (for any $\varepsilon > 0$), the received word will have only the transmitted codeword in a Hamming ball of radius ρ around it. Thus, for random errors, one can correct *twice* the number of errors uniquely correctable from worst-case errors for any code. A variant of our result also gives a simple algorithm to decode Reed-Solomon codes from random errors that, to the best of our knowledge, runs faster than known algorithms for certain ranges of parameters.
2. We show that concatenated codes can achieve the list decoding capacity for *erasures*. A similar result for worst-case errors was proven by Guruswami and Rudra (SODA 08), although their result does not directly imply our result. Our results show that a subset of the random ensemble of codes considered by Guruswami and Rudra also achieve the list decoding capacity for erasures.

Our proofs employ simple counting and probabilistic arguments.

*Research supported by NSF CAREER Award CCF-0844796.

ISSN 1433-8092

1 Introduction

List decoding is a relaxation of the traditional unique decoding paradigm, where one is allowed to output a list of codewords that are close to the received word. This relaxation allows for designing list decoding algorithms that can recover from scenarios where almost all of the redundancy could have been corrupted [18, 8, 15, 6]. In particular, one can design binary codes from which one can recover from a $1/2 - \varepsilon$ fraction of errors. This fact has lead to many surprising applications in complexity theory– see e.g. the survey by Sudan [19] and Guruswami's thesis [4, Chap. 12].

The results mentioned above mostly deal with worst-case errors, where the channel is considered to be an adversary that can corrupt any arbitrary fraction of symbols (with an upper bound on the maximum fraction of such errors). In this work, we deal with random and erasure noise models, which are weaker than the worst-case errors model, and which also have interesting applications in complexity theory.

1.1 Random Errors

It is well-known that for worst-case errors, one cannot uniquely recover the transmitted codeword if the total number of errors exceeds half the distance. (We refer the reader to Section 2 for definitions related to codes.) List decoding circumvents this by allowing the decoder to output multiple nearby codewords. In situations where the decoder has access to some side information, one can prune the output list to obtain the transmitted codeword. In fact, most of the applications of list decoding in complexity theory crucially use side information. However, a natural question to ask is what one can do in situations where there is no side information (this is not an uncommon assumption in the traditional point-to-point communication model).

In such a scenario, it makes sense to look at a weaker random noise model and try to argue that the pathological cases that prevent us from decoding a code with relative distance δ from more than $\delta/2$ fraction of errors are rarely encountered.

Before we move on, we digress a bit to establish our notion of random errors. In our somewhat non-standard model, we assume that the adversary can pick the location of the ρ fraction of error *positions* but that the errors themselves are random. For the binary case, this model coincides with worst-case errors, so in this work, we consider alphabet size $q \geqslant 3$. We believe that this is a nice intermediary to the worst-case noise model and the more popular models of random noise, where errors are *independent* across different symbols. Indeed, a result with high probability in our random noise model (for roughly ρ errors) immediately implies a similar result for a more benign random noise model such as the q-ary symmetric noise channel with cross-over probability ρ.[1] For the rest of the paper, when we say random errors, we will be referring to the stronger random noise model above.

Related Work. The intuition that pathological worst-case errors are rare has been formalized for certain families of codes. For example, McEliece showed that for Reed-Solomon codes with distance δ, with high probability, for a fraction $\rho \leqslant \delta - \varepsilon$ of random errors, the output list size is one [14].[2] Further, for *most* codes of rate $1 - H_q(\rho) - \varepsilon$, with high probability, for a ρ fraction of random errors, the output list size is one. (This follows from Shannon's famous result on the capacity of the q-ary symmetric channel: for a proof, see e.g. [17].) It is also known that most

[1] In this model, every transmitted symbol remains untouched with probability $1 - \rho$ and is mapped to the other $q - 1$ possible symbols with probability $\rho/(q - 1)$. Finally, the noise acts independently on each symbol.

[2] The actual result is slightly weaker: see Section 3 for more details.

codes of rate $1 - H_q(\rho) - \varepsilon$ have relative distance at least ρ. Further, for $q \geqslant 2^{\Omega(1/\varepsilon)}$, it is known that such a code cannot have distance more than $\rho + \varepsilon$: this follows from the Singleton bound and the fact that for such an alphabet size, $1 - H_q(\rho) \geqslant 1 - \rho - \varepsilon$ (cf. [16, Sec 2.2.2]).

Our Results. In our first main result, we show that the phenomenon above is universal, that is, for *every* q-ary code, with $q \geqslant 2^{\Omega(1/\varepsilon)}$, the following property holds: if the code has relative distance δ, then for any $\rho \leqslant \delta - \varepsilon$ fraction of random errors, with high probability, the Hamming ball of fractional radius ρ around the received word will only have the transmitted codeword in it. We would like to point out three related points. First, our result implies that if we relax the worst-case error model to a random error model, then combinatorially one can always correct *twice* the number of errors. Second, one cannot hope to correct more than a δ fraction of random errors: it is easy to see that, for instance, for Reed-Solomon codes, *any* error pattern of relative Hamming weight $\rho > \delta$ will give rise to a list size greater than one. Finally, the proof of our result follows from a fairly straightforward counting argument.

A natural follow-up question to our result is whether the lower bound of $2^{\Omega(1/\varepsilon)}$ on q can be relaxed. We show that if q is $2^{o(1/\varepsilon)}$, then the result above is *not true*. This negative result follows from the following two observations/results. First, it is known that for *any* code with rate $1 - H_q(\rho) + \varepsilon$, the average list size, over all possible received words, is exponential. Second, it is known that Algebraic-Geometric (AG) codes over alphabets of size at least 49 can have relative distance *strictly* bigger than $1 - H_q(\rho)$ (cf. [10]). However, these two results do not immediately imply the negative result for the random error case. In particular, what we need to show is that there is at least one codeword $\mathbf{c}$ such that for most error patterns $\mathbf{e}$ of relative Hamming weight ρ, the received word $\mathbf{c} + \mathbf{e}$ has at least one codeword other than $\mathbf{c}$ within a relative Hamming distance of ρ from it. To show that this can indeed be true for AG codes, we use a generalization of an "Inverse Markov argument" from Dumer et al. [1].

A Cryptographic Application. In addition to being a natural noise model to study, list decoding in the random error model has applications in cryptography. In particular, Kiayias and Yung have proposed cryptosystems based on the hardness of decoding Reed-Solomon codes [11]. However, if for Reed-Solomon codes (of rate R), one can list decode ρ fraction of *random* errors then the cryptosystem from [11] can be broken for the corresponding parameter settings. Since Guruswami-Sudan can solve this problem for $\rho \leqslant 1 - \sqrt{R}$ for *worst-case* errors [8], Kiayias and Yung set the parameter $\rho > 1 - \sqrt{R}$. Beyond the $1 - \sqrt{R}$ bound, to the best of our knowledge, the only known algorithms to decode Reed-Solomon codes are the following trivial ones: (i) Go through all possible q^k codewords and output all the codewords with Hamming distance of ρ from the received word; and (ii) Go through all possible $\binom{n}{\rho n}$ error locations and output the codeword, if any, that agrees in the $(1 - \rho)n$ "non-error" locations.

It is interesting to note that each of the three algorithms mentioned above work in the stronger model of worst-case errors. However, since we only care about decoding from random errors, one might hope to design better algorithms that make use of the fact that the errors are random. In this paper, we show that (essentially) the proof of our first main result implies a related result that in turn implies a modest improvement in the running time of algorithms to decode Reed-Solomon codes from $\rho > 1 - \sqrt{R}$ fraction of random errors. The related result states the following: for any code with relative distance δ (over a large enough alphabet) with high probability, for a ρ fraction of random errors, Hamming balls of fractional radius $\delta - \varepsilon$ around the received word only have the transmitted codeword in them.[3] Note that unlike the statement of our result mentioned earlier, we are considering Hamming balls of radius *larger* than the fraction of errors. This allows us to improve the second trivial algorithm in the paragraph above so that one needs to verify fewer "error patterns." This leads to an asymptotic improvement in the running time over both of the trivial algorithms for certain setting of parameters, though the running time is still exponential and thus, too expensive to break the Kiayias-Yung cryptosystem.

1.2 Erasures

In the second part of the paper, we consider the erasure noise model, where the decoder knows the locations of the errors. (However, the error locations are still chosen by the adversary.) Intuitively, this noise model is weaker than the general worst-case noise model as the decoder knows for sure which locations are uncorrupted. This intuition can also be formalized. E.g., it is known that for a ρ fraction of worst-case errors, the list decoding capacity is $1 - H_q(\rho)$, whereas for a ρ fraction of erasures, the list decoding capacity is $1 - \rho$ (cf. [4, Chapter 10]). Note that the capacity for erasures is *independent* of the alphabet size. As another example, for a linear code, a combinatorial guarantee on list decodability from erasures gives a polynomial time list decoding algorithm. By contrast, such a result is not known for worst-case errors.

As is often the case, the capacity result is proven by random coding arguments. A natural quest then is to design explicit linear codes that achieve the list decoding capacity for erasures, and is an important milestone in the program of designing explicit codes that achieve list decoding capacity for worst-case errors. This goal is the primary motivation for our second main result.

Our Result and Related Work. For large enough alphabets, explicit linear codes that achieve list decoding capacity for erasures are not hard to find: e.g., Reed-Solomon codes achieve the capacity. For smaller alphabets, the situation is much different. For binary codes, Guruswami presented explicit linear codes that can handle $\rho = 1-\varepsilon$ fraction of erasures with rate $\Omega\left(\frac{\varepsilon^2}{\log(1/\varepsilon)}\right)$ [3]. For alphabets of size 2^t, $1 - \varepsilon$ fraction of erasures can be list decoded with explicit linear codes of rate $\Omega\left(\frac{\varepsilon^{1+1/t}}{t^2 \log(1/\varepsilon)}\right)$ [4, Chapter 10]. Thus, especially for binary codes, an explicit code with capacity of $1 - \rho$ is still a lofty goal. (In fact, breaking the ε^2 rate barrier for *polynomially* small ε would imply explicit construction of certain bipartite Ramsey graphs, solving an open question [3].)

To gain a better understanding about codes that achieve list decoding capacity for erasures, a natural question is to ask whether *concatenated codes* can achieve the list decoding capacity for erasures. Concatenated codes are the preeminent method to construct good list decodable codes over small alphabets. In fact, the best explicit list decodable binary codes (for both erasures [3] and worst-case errors [7]) are concatenated codes. Briefly, in code concatenation, an "outer" code over a large alphabet is first used to encode the message. Then "inner" codes over the smaller alphabet are used to encode each of the symbols in the outer codeword. These inner codes typically have a much smaller block length than the outer code, which allows one to use brute-force type algorithms to search for "good" inner codes. Also note that the rate of the concatenated code is the product of the rate of the outer and inner codes.

Given that concatenated codes have such a rigid structure, it seems plausible that such codes would not be able to achieve list decoding capacity. For the worst-case error model, Guruswami and Rudra showed that there do exist concatenated codes that achieve list decoding capacity [5].

[3] A similar result was shown for Reed-Solomon codes by McEliece [14].

However, for erasures there is an additional potential complication that does not arise for the worst-case error case. In particular, consider erasure patterns in which ρ fraction of the outer symbols are completely erased. It is clear by this example that the outer code needs to have rate very close to $1-\rho$. However, note that to approach list decoding capacity for erasures, the concatenated code needs to have rate $1-\rho-\varepsilon$. This means that the inner codes need to have rate very close to 1. By contrast, even though the result of [5] has some restrictions on the rate of the inner codes, it is not nearly as stringent as the requirement above. (The restriction in [5] seems to be an artifact of the proof, whereas for erasures, the restriction is unavoidable.) Further, this restriction on the inner rate is just by looking at a specific class of erasure patterns. It is reasonable to wonder if when taking into account all possible erasure patterns, we can rule out the possibility of concatenated codes achieving the list decoding capacity for erasures.

In our second main result, we show that concatenated codes *can* achieve the list decoding capacity for erasures. In fact, we show that choosing the outer code to be a Folded Reed-Solomon code ([6]) and picking the inner codes to be random independent linear codes with rate 1, will with high probability, result in a linear code that achieves the list decoding capacity for erasures. We show a similar result (but with better bounds on the list size) when the outer code is also chosen to be a random linear code. Both of these ensembles were shown to achieve the list decoding capacity for errors in [5], although, as mentioned earlier, the result for errors holds for a superset of concatenated codes (as the inner codes could have rates strictly less than 1). The proof of our result is similar to the proof structure in [5]. Because we are dealing with the more benign erasure noise model, some of the calculations in our proofs are much simpler than the corresponding ones in [5].

Approximating NP Witnesses. We conclude this section by pointing out that an application of binary codes that are list decodable from erasures is to the problem of approximating NP-witnesses [2, 12]. For any NP-language L, we have a polynomial-time decidable relation $R_L(\cdot,\cdot)$ such that $x \in L$ if and only if there exists a polynomially sized witness w such that $R_L(x,w)$ accepts. Thus, for an NP-complete language we do not expect to be able to compute the witness w in polynomial time given x. A natural notion of approximation is the following: given an ε fraction of the bits in a a correct witness w, can we verify if $x \in L$ in polynomial time? The results in [2, 12] show that such an approximation is not possible unless P=NP.

To be more precise, Gál et al. ([2]) consider the following problem: given a SAT formula ϕ over n variables can we, in polynomial time, compute another SAT formula ϕ' over $N = \text{poly}(n)$ variables such that given εN bits from a satisfying assignment to ϕ', we can compute a satisfying assignment to the original formula ϕ?

Kumar and Sivakumar's ([12]) reduction works for any NP-language L. However, their reduction computes a polynomial-time computable relation R'_L (with witness size $N = \text{poly}(n)$), which is different from the original predicate R_L such that the knowledge of εN many bits of some satisfying witness for R'_L can be used in polynomial time to compute a satisfying witness for R'_L. Both of these results are proven by picking a linear binary code C that can be list decoded from a $1-\varepsilon$ fraction of erasures and "encoding" $C(x)$ (where x is the input) into the definition of ϕ' (in the case of [2]) or R'_L (in the case of [12]). The intuition behind these reductions is that given sufficiently many bits of a satisfying witness, we can obtain a list of potentially satisfying witnesses by running the list decoding algorithm for C to recover from the erasures. (The connection to list decoding was implicit in [2]– it was made explicit in [12].)

Guruswami and Sudan ([9]) show that the reductions above can be made to work with $\varepsilon = N^{-1/2+\gamma}$ for the Kumar and Sivakumar problem and with $\varepsilon = N^{-1/4+\gamma}$ for the Gál et al. problem (for any constant $\gamma > 0$). An explicit linear code that meets the list decoding capacity for erasures will improve the value of ε above to $N^{-1+\gamma}$ and $N^{-1/2+\gamma}$, respectively.

Organization of the Paper. We begin with some preliminaries in Section 2. We present our first main result on random codes in Section 3 and our second main result on erasures in Section 4.

2 Preliminaries

For an integer $m \geqslant 1$, we will use $[m]$ to denote the set $\{1, \ldots, m\}$.

Basic Coding Definitions. A code C of *dimension* k and *block length* n over an alphabet Σ is a subset of Σ^n of size $|\Sigma|^k$. The *rate* of such a code equals k/n. Each n-tuple in C is called a codeword. Let $\mathbb{F}_q$ denote the field with q elements. A code C over $\mathbb{F}_q$ is called a linear code if C is a subspace of $\mathbb{F}_q^n$. In this case the dimension of the code coincides with the dimension of C as a vector space over $\mathbb{F}_q$. By abuse of notation we can also think of a linear code C as a map from an element in $\mathbb{F}_q^k$ to its corresponding codeword in $\mathbb{F}_q^n$, mapping a row vector $\mathbf{x} \in \mathbb{F}_q^k$ to a vector $\mathbf{xG} \in \mathbb{F}_q^n$ via a $k \times n$ matrix $\mathbf{G}$ over $\mathbb{F}_q$ which is referred to as the generator matrix.

The Hamming distance between two vectors in $\mathbf{x}, \mathbf{y} \in \Sigma^n$, denoted by $\Delta(\mathbf{x}, \mathbf{y})$, is the number of places they differ in. The (minimum) distance of a code C is the minimum Hamming distance between any two distinct codewords from C. The relative distance is the ratio of the distance to the block length.

We will need the following notions of the weight of a vector. Given a vector $\mathbf{v} \in \{0, 1, \ldots, q-1\}^n$, its Hamming weight, which is the number of non-zero entries in the vector, is denoted by $\textsc{wt}(\mathbf{v})$. Given a vector $\mathbf{y} = (y_1, \ldots, y_n) \in \{0, \ldots, q-1\}^n$ and a subset $S \subseteq [n]$, $\mathbf{y}_S$ will denote the subvector $(y_i)_{i \in S}$, and $\textsc{wt}_S(\mathbf{y})$ will denote the Hamming weight of $\mathbf{y}_S$.

Code Concatenation. Concatenated codes are constructed from two different types of codes that are defined over alphabets of different sizes. If we are interested in a concatenated code over $\mathbb{F}_q$, then the *outer code* C_{out} is defined over $\mathbb{F}_Q$, where $Q = q^k$ for some positive integer k, and has block length N. The second type of codes, called the *inner codes*, and which are denoted by $C_{\text{in}}^1, \ldots, C_{\text{in}}^N$, are defined over $\mathbb{F}_q$ and are each of dimension k (note that the message space of C_{in}^i for all i and the alphabet of C_{out} have the same size). The concatenated code, denoted by $C = C_{\text{out}} \circ (C_{\text{in}}^1, \ldots, C_{\text{in}}^N)$, is defined as follows: Let the rate of C_{out} be R and let the block lengths of C_{in}^i be n (for $1 \leqslant i \leqslant N$). Define $K = RN$ and $r = k/n$. The input to C is a vector $\mathbf{m} = \langle m_1, \ldots, m_K \rangle \in (\mathbb{F}_q^k)^K$. Let $C_{\text{out}}(\mathbf{m}) = \langle x_1, \ldots, x_N \rangle$. The codeword in C corresponding to $\mathbf{m}$ is defined as follows

$$C(\mathbf{m}) = \langle C_{\text{in}}^1(x_1), C_{\text{in}}^2(x_2), \ldots, C_{\text{in}}^N(x_N) \rangle.$$

The outer code C_{out} in this paper will either be a random linear code over $\mathbb{F}_Q$ or the folded Reed-Solomon code from [6]. In the case when C_{out} is random linear, we will pick C_{out} by selecting $K = RN$ vectors uniformly at random from $\mathbb{F}_Q^N$ to form the rows of the generator matrix. For every position $1 \leqslant i \leqslant N$, we will choose an inner code C_{in}^i to be a random linear code over $\mathbb{F}_q$ of block length n and rate $r = k/n$. In particular, we will work with the corresponding generator matrices $\mathbf{G}_i$, where every $\mathbf{G}_i$ is a random $k \times n$ matrix over $\mathbb{F}_q$. All the generator matrices $\mathbf{G}_i$ (as well as the generator matrix for C_{out}, when we choose a random C_{out}) are chosen independently. This fact will be used crucially in our proofs.

List Decoding. We define some terms related to list decoding.

Definition 1 (List decodable code for errors). *For $0 < \rho < 1$ and an integer $L \geqslant 1$, a code $C \subseteq \Sigma^n$ is said to be (ρ, L)-list decodable if for every $\mathbf{y} \in \Sigma^n$, the number of codewords in C that are within Hamming distance ρn from y is at most L.*

Given a vector $\mathbf{c} = (c_1, \ldots, c_n) \in \Sigma^n$ and an erased received word $\mathbf{y} = (y_1, \ldots, y_n) \in (\Sigma \cup \{?\})^n$,[4] we will use $\mathbf{c} \simeq \mathbf{y}$ to denote the fact that for every $i \in [n]$ such that $y_i \neq ?$, $c_i = y_i$. With this definition, we are ready to define the notion of list decodability for erasures. Further, for an erased received word, we will use $\text{WT}(\mathbf{y})$ to denote the number of erased positions.

Definition 2 (List decodable code for erasures). *For $0 < \rho < 1$ and an integer $L \geqslant 1$, a code $C \subseteq \Sigma^n$ is said to be $(\rho, L)_{led}$-list decodable if for every $\mathbf{y} \in (\Sigma \cup \{?\})^n$ with $\text{WT}(\mathbf{y}) \leqslant \rho n$, the number of codewords $\mathbf{c} \in C$ such that $\mathbf{c} \simeq \mathbf{y}$ is at most L.*

Reed-Solomon and Related Codes. The classical family of Reed-Solomon (RS) codes over a field $\mathbb{F}$ are defined to be the evaluations of low-degree polynomials at a sequence of distinct points of $\mathbb{F}$. Folded Reed-Solomon codes are obtained by viewing the RS code as a code over a larger alphabet $\mathbb{F}^s$ by bundling together s consecutive symbols for some folding parameter s. We will not need any specifics of folded RS codes (in fact, even their definition) beyond certain properties that we recall in Section 4.

3 Random Errors

In this section we consider the random noise model mentioned in the introduction: the error locations are adversarial but the errors themselves are random. Our main result is the following.

Theorem 1. *Let $0 < \varepsilon, \delta < 1$ be reals and let q and $n \geqslant \Omega(1/\varepsilon)$ be positive integers. Let $\Sigma = \{0, 1, \ldots, q-1\}$.[5] Let $0 < \rho \leqslant \delta - \varepsilon$ be a real. Let C be a code over Σ of block length n and relative distance δ. Let $S \subseteq [n]$ with $|S| = (1-\rho)n$. Then the following hold:*

(a) If $q \geqslant 2^{\Omega(1/\varepsilon)}$, then for every codeword $\mathbf{c}$ and all but a $q^{-\Omega(\varepsilon n)}$ fraction of error patterns $\mathbf{e} \in \Sigma^n$ with $\text{WT}(\mathbf{e}) = \rho n$ and $\text{WT}_S(\mathbf{e}) = 0$, the only codeword within the Hamming ball of radius ρn around the received word $\mathbf{c} + \mathbf{e}$ is $\mathbf{c}$.

(b) Let $\gamma > 0$. If $q > \max\left(n, \left(\frac{e}{1-\delta+\varepsilon}\right)^{\left\lceil \frac{1}{\gamma} \right\rceil}\right)$, then for every codeword $\mathbf{c}$ and all but a $(q-1)^{-((1-\gamma)\varepsilon/2-(1-\delta)\gamma)n}$ fraction of error patterns $\mathbf{e} \in \Sigma^n$ with $\text{WT}(\mathbf{e}) = \rho n$ and $\text{WT}_S(\mathbf{e}) = 0$, the only codeword within the Hamming ball of radius $(\delta - \varepsilon)n$ around the received word $\mathbf{c} + \mathbf{e}$ is $\mathbf{c}$.

A weaker version of Theorem 1 was previously known for RS codes [14]. (Though the bounds for part (b) are better in [14].) In particular, McEliece showed Theorem 1 for RS codes but over *all* error patterns of Hamming weight ρn. In other words, Theorem 1 implies the result in [14] if we average our result over all subsets $S \subseteq [n]$ with $|S| = \rho n$.

[4] ? denotes an erasure.

[5] We will assume that Σ is equipped with a monoid structure, i.e. for any $a, b \in \Sigma$, $a + b \in \Sigma$ and 0 is the identity element.

Part (a) of Theorem 1 implies that for $e \leqslant (\delta - \varepsilon)n$ random errors, with high probability, the Hamming ball of radius e has one codeword in it. Note that this is *twice* the number of errors for which an analogous result can be shown for worst-case errors. Part (b) of Theorem 1 implies the following property of Reed-Solomon codes (where we pick $\varepsilon = 4R$ and $\gamma = 1/2$).

Corollary 2. *Let $k \leqslant n < q$ be integers such that $q > \left(\frac{n}{k}\right)^2$. Then the following property holds for Reed-Solomon codes of dimension k and block length n over $\mathbb{F}_q$. For at least $1 - q^{-\Omega(k)}$ fraction of error patterns $\mathbf{e}$ of Hamming weight* at most $n - 4k$ *and any codeword $\mathbf{c}$, the only codeword that agrees in at least $4k$ positions with $\mathbf{c} + \mathbf{e}$ is $\mathbf{c}$.*

We would like to point out that in Corollary 2, the radius of the Hamming ball can be *larger* than the number of errors. This can be used to slightly improve upon the best known algorithms to decode RS codes from random errors beyond the Johnson bound for super-polynomially large q. See Section 3.1 for more details.

A natural question is whether the lower bound of $q \geqslant 2^{\Omega(1/\varepsilon)}$ in part (a) of Theorem 1 can be improved. In Section 3.2 we show that this is not possible.

Proof of Theorem 1. Let $\mathbf{c} \in C$ be the transmitted codeword. For an $\alpha \geqslant 1 - \delta + \varepsilon$, we call an error pattern $\mathbf{e}$ (with $\mathrm{WT}(\mathbf{e}) = \rho n$ and $\mathrm{WT}_S(\mathbf{e}) = 0$) *$\alpha$-bad* if there exits a codeword $\mathbf{c}' \neq \mathbf{c} \in C$ such that $\Delta(\mathbf{c}+\mathbf{e}, \mathbf{c}') = (1-\alpha)n$ (and every other codeword has a larger Hamming distance from $\mathbf{c}+\mathbf{e}$). We will show that the number of α-bad error patterns (over all $\alpha \geqslant 1 - \delta + \varepsilon$) is an exponentially small fraction of error patterns $\mathbf{e}$ with $\mathrm{WT}(\mathbf{e}) = \rho n$ and $\mathrm{WT}_S(\mathbf{e}) = 0$, which will prove the theorem.

Fix $\alpha \geqslant 1-\delta+\varepsilon$. Associate every α-bad error pattern $\mathbf{e}$ with the lexicographically first codeword $\mathbf{c}' \neq \mathbf{c} \in C$ such that $\Delta(\mathbf{c} + \mathbf{e}, \mathbf{c}') = (1 - \alpha)n$. Let $A \subseteq [n]$ be the set of positions where $\mathbf{c}'$ and $\mathbf{c} + \mathbf{e}$ agree. Further, define $S_0 = S \cap A$, $S_1 = A \cap ([n] \setminus S)$ and $\beta = |S_0|/n$. Thus, for every α-bad error pattern $\mathbf{e}$, we can associate such a pair of subsets $(S_0, S_1) \subseteq S \times ([n] \setminus S)$. Hence, to count the number of α-bad error patterns it suffices to count for each possible pair (S_0, S_1), with $|S_0| = \beta n$ and $|S_1| = (\alpha - \beta)n$ for some $\alpha - \rho \leqslant \beta \leqslant \alpha$, the number of α-bad patterns that can be associated with it. (The lower and upper bounds on β follow from the fact that $S_1 \subseteq [n] \setminus S$ and $S_0 \subseteq A$, respectively.)

Fix sets $S_0 \subseteq S$ and $S_1 \subseteq [n] \setminus S$ with $|S_0| = \beta n$ and $|S_1| = (\alpha-\beta)n$ for some $\alpha-\rho \leqslant \beta \leqslant \alpha$. To upper bound the number of α-bad error patterns that are associated with (S_0, S_1), first note that such error patterns take all the $(q-1)^{(\rho-\alpha+\beta)n}$ possible values at the positions in $[n]\setminus(S \cup S_1)$. Fix a vector $\mathbf{x}$ of length $n-|S|-|S_1|$ and consider all the α-bad error patterns $\mathbf{e}$ such that $\mathbf{e}_{[n]\setminus(S\cup S_1)} = \mathbf{x}$. Recall that each error pattern is associated with a codeword $\mathbf{c}' \neq \mathbf{c}$ such that $\mathbf{c}'$ and $\mathbf{c} + \mathbf{e}$ agree exactly in the positions $S_0 \cup S_1$. Further, such a codeword $\mathbf{c}'$ is associated with exactly one α-bad error pattern $\mathbf{e}$, where $\mathbf{e}_{[n]\setminus(S\cup S_1)} = \mathbf{x}$. (This is because fixing $\mathbf{c}'$ fixes $\mathbf{e}_{S_1}$ and $\mathbf{e}_S$ is already fixed by the definition of S.) Thus, to upper bound the number of α-bad error patterns associated with (S_0, S_1), where $\mathbf{e}_{[n]\setminus(S\cup S_1)} = \mathbf{x}$ (call this number $N_{\alpha,S_0,S_1,\mathbf{x}}$), we will upper bound the number of such codewords $\mathbf{c}'$. Note that as C has relative distance δn, once any $(1-\delta)n+1$ positions are fixed, there is at most one codeword that agrees with the fixed positions (if there is no such codeword then the corresponding "error pattern" does not exist). Thus, there is at most one possible $\mathbf{c}'$ once we fix (say) the "first" $(1 - \delta)n + 1 - |S_0|$ values of $\mathbf{e}_{S_1}$ (recall that $\mathbf{c}'_{S_0} = \mathbf{c}_{S_0}$). This implies that

$$N_{\alpha,S_0,S_1,\mathbf{x}} \leqslant (q - 1)^{(1-\delta-\beta)n+1}.$$

Let M_α be the number of choices for (S_0, S_1), which is just the number of choices for A. As the number of choices for $\mathbf{x}$ is $(q - 1)^{(\rho-\alpha+\beta)n}$, the number of α-bad error patterns is at most

$$M_\alpha \cdot (q - 1)^{(\rho-\alpha+\beta)n} \cdot (q - 1)^{(1-\delta-\beta)n+1} = M_\alpha \cdot (q - 1)^{(1-\delta-\alpha)n+1} \cdot (q - 1)^{\rho n}. \quad (1)$$

Proof of part(a). Note that the number of α-bad patterns for any $\alpha \geqslant 1-\delta+\varepsilon$ is upper bounded by

$$M_\alpha \cdot (q-1)^{-\varepsilon n+1} \cdot (q-1)^{\rho n}.$$

We trivially upper bound M_α by 2^n. Recalling that there are $(q-1)^{\rho n}$ error patterns $\mathbf{e}$ with $\mathrm{WT}(\mathbf{e}) = \rho n$ and $\mathrm{WT}_S(\mathbf{e}) = 0$ and that α can take at most n values, the fraction of α-bad patterns (over all $\alpha \geqslant 1-\rho \geqslant 1-\delta+\varepsilon$) is at most

$$n2^n(q-1)^{-\varepsilon n+1} \leqslant (q-1)^{\left(-\varepsilon+\frac{2}{\log(q-1)}+\frac{1}{n}\right)n} \leqslant (q-1)^{-\varepsilon n/3} \leqslant q^{-\varepsilon n/6},$$

where the first inequality follows from the fact that $n \leqslant 2^n$, the second inequality is true for $n \geqslant 3/\varepsilon$ and $q \geqslant 2^{6/\varepsilon}$ and the last inequality follows from the inequality $(q-1) \geqslant \sqrt{q}$ (which in turn is true for $q \geqslant 3$).

Proof of part (b). Note that $M_\alpha = \binom{n}{\alpha n} \leqslant (e/\alpha)^{\alpha n}$. Thus, the number of α-bad error patterns is upper bounded by

$$(q-1)^{\left(1-\delta-\alpha+\alpha\cdot\frac{\log(e/\alpha)}{\log(q-1)}\right)n+1}\cdot(q-1)^{\rho n} \leqslant (q-1)^{(1-\delta-\alpha(1-\gamma))n+1}\cdot(q-1)^{\rho n} \leqslant (q-1)^{(-(1-\gamma)\varepsilon+\gamma(1-\delta))n+1}\cdot(q-1)^{\rho n},$$

where the inequalities follow from the facts that $q > \left(\frac{e}{1-\delta+\varepsilon}\right)^{1/\gamma}$ and $\alpha \geqslant 1-\delta+\varepsilon$. Recalling that there are $(q-1)^{\rho n}$ error patterns $\mathbf{e}$ with $\mathrm{WT}(\mathbf{e}) = \rho n$ and $\mathrm{WT}_S(\mathbf{e}) = 0$ and that α can take at most n values, the fraction of α-bad patterns (over all $\alpha \geqslant 1-\delta+\varepsilon$) is at most

$$n(q-1)^{(-(1-\gamma)\varepsilon+(1-\delta)\gamma)n+1} \leqslant (q-1)^{\left(-(1-\gamma)\varepsilon+\gamma(1-\delta)+\frac{2}{n}\right)n} \leqslant (q-1)^{\left(-\frac{(1-\gamma)\varepsilon}{2}+\gamma(1-\delta)\right)n},$$

where the first inequality follows from the fact that $q > n$ and the second inequality is true for $n \geqslant 4/((1-\gamma)\varepsilon)$. ■

3.1 An Implication of Corollary 2

To the best of our knowledge, for $e > n-\sqrt{kn}$, the only known algorithms to decode Reed-Solomon (RS) codes from e random errors are the trivial ones: (i) Go through all possible codewords and output the closest codeword– this takes $2^{O(k\log q)}\cdot n$ time and (ii) Go through all possible $\binom{n}{e}$ error locations and check that the received word outside the purported error locations is indeed a RS codeword– this takes $2^{O((n-e)\log(n/(n-e)))}\cdot O(n^2)$ time.

If $e \leqslant n-4k$, then by Corollary 2, we can go through all the $\binom{n}{4k}$ choices of subsets of size $4k$ and check if the received word projected down to the subset lies in the corresponding projected down RS code. This algorithm takes $2^{O(k\log(n/k))}\cdot O(n^2)$ time, which is better than the trivial algorithm (ii) mentioned above for e in $n-\omega(k)$. Further, this algorithm is better than the trivial algorithm (i) when q is super-polynomially large in n.

3.2 On the Alphabet Size in Theorem 1

It is well-known that *any* code that is (ρ, L)-list decodable that also has rate at least $1-H_q(\rho)+\varepsilon$ needs to satisfy $L = q^{\Omega(\varepsilon n)}$ (cf. [4]). A natural way to try to show that part (a) of Theorem 1 is false for $q \leqslant 2^{o(1/\varepsilon)}$ is to look at codes whose relative distance is strictly larger than $1-H_q(\rho)$. Algebraic-geometric (AG) codes are a natural candidate since they can beat the Gilbert-Varshamov bound for an alphabet size of at least 49 (cf. [10]). The only catch is that the lower bound on L

follows from an average case argument and we need to show that over *most* error patterns, the list size is more than one. For this we need an "Inverse Markov argument," like one in [1].

(The argument above was suggested to us by Venkat Guruswami.)

We begin with the more general statement of the "Inverse Markov argument" from [1]. (We thank Madhu Sudan for the statement and its proof.)

Lemma 3. *Let $G = (L, R, E)$ be a bipartite graph with $|L| = n_L$ and $|R| = n_R$. Let the average left degree of G be denoted by $\bar{d_L}$. Note that the average right degree is $\bar{d_R} = \frac{n_L \cdot d_L}{n_R}$. Then the following statements are true:*

(i) If we pick an edge $e = (u, v)$ uniformly at random from E, then the probability that[6] *$d(v) \leqslant \varepsilon \bar{d_R}$ is at most ε.*

(ii) If G is d-left regular then consider the following process: Uniformly at random pick a vertex $u \in L$. Then uniformly at random pick a vertex $v \in R$ in u's neighborhood. Then the probability that $d(v) \leqslant \varepsilon \frac{dn_L}{n_R}$ is at most ε.

***Proof*.** We first note that (ii) follows from (i) as the random process in (ii) ends up picking edges uniformly at random from E.

To conclude, we prove part (i). Consider the set $R' \subseteq R$ such that $v \in R'$ satisfies $d(v) \leqslant \varepsilon \bar{d_R}$. Note that that the maximum number of edges that have an end-point in R' is at most $\varepsilon \bar{d_R} \cdot n_R = \varepsilon |E|$. Thus, the probability that a uniformly random edge in E has an end point in R' is upper bounded by $\varepsilon |E|/|E| = \varepsilon$, as desired. ■

The following is an easy consequence of Lemma 3 and the standard probabilistic method used to prove the lower bound for list decoding capacity.

Lemma 4. *Let $q \geqslant 2$ and $0 \leqslant \rho < 1 - 1/q$. Then the following holds for large enough n. Let $C \subseteq \{0, \ldots, q-1\}^n$ be a code with rate $1 - H_q(\rho) + \gamma$. Then there exists a codeword $\mathbf{c} \in C$ such that for at least a $1 - q^{-\Omega(\gamma n)}$ fraction of error patterns $\mathbf{e}$ of Hamming weight at most ρn, it is true that the Hamming ball of radius ρn around $\mathbf{c} + \mathbf{e}$ has at least two codewords from C in it.*

***Proof*.** Define the bipartite graph $G_{C,\rho} = (C, \{0, \ldots, q-1\}^n, E)$ as follows. For every $\mathbf{c} \in C$, add $(\mathbf{c}, \mathbf{y}) \in E$ such that $\Delta(\mathbf{c}, \mathbf{y}) \leqslant \rho n$. Note that $G_{C,\rho}$ is a $\mathrm{Vol}_q(\rho n)$-left regular bipartite graph, where $\mathrm{Vol}_q(r)$ is the volume of the q-ary Hamming ball with radius r. Note that the graph has an average right degree of

$$\bar{d_R} = \frac{\mathrm{Vol}_q(\rho n) \cdot q^{(1 - H_q(\rho) + \gamma)n}}{q^n} \geqslant q^{\gamma n - o(n)},$$

where in the above we have used the following well known inequality (cf. [13]):

$$\mathrm{Vol}_q(\rho n) \geqslant q^{H_q(\rho)n - o(n)}.$$

Thus, by part (b) of Lemma 3 (with $\varepsilon = (\bar{d_R})^{-1} \leqslant q^{-\gamma n + o(n)}$), we have

$$\Pr_{\mathbf{c} \in C} \Pr_{\substack{\mathbf{e} \in \{0,\ldots,q-1\}^n \\ \mathrm{WT}(\mathbf{e}) \leqslant \rho n}} [\mathbf{c} + \mathbf{e} \text{ has at most one codeword within Hamming distance } \rho n] \leqslant q^{-\gamma n + o(n)}.$$

Thus, there must exist at least one codeword $\mathbf{c} \in C$ with the required property. ■

[6]For any vertex v, we denote its degree by $d(v)$.

Thus, given Lemma 4, we can prove that part (a) of Theorem 1 is not true for a certain value of q if there exists a code $C \subseteq \{0, \ldots, q-1\}^n$ with relative distance δ such that it has rate at least $1 - H_q(\delta - \varepsilon) + \gamma$ for some $\gamma > 0$. Now it is known that for fixed $\alpha > 0$, $H_q(\alpha) \geqslant \alpha + \Omega\left(\frac{1}{\log q}\right)$ (cf. [20, Lecture 7]). Thus, we would be done if we could find a code with relative distance δ and rate at least

$$1 - \delta + \varepsilon + \gamma - O(1/\log q).$$

For $q \leqslant 2^{o(1/\varepsilon)}$, the bound above for small enough ε is upper bounded by $1-\delta-\varepsilon-\frac{1}{\sqrt{q}-1}$ (assuming that $\gamma = \Theta(\varepsilon)$). It is known that AG codes over alphabets of size $\geqslant 49$ with relative distance δ exist that achieve a rate of $1 - \delta - \frac{1}{\sqrt{q}-1}$. Thus, for $49 \leqslant q \leqslant 2^{o(1/\varepsilon)}$, AG codes over alphabets of size q are the required codes.

4 Concatenated Codes

This section first shows that with folded Reed-Solomon codes and independently chosen small random linear inner codes, the resulting concatenated code can achieve erasure capacity in a list decoding setting. A similar result holds when the outer code is a random linear code, and this result is presented second.

4.1 Folded Reed-Solomon Outer Code

Theorem 5. *Let q be a prime power and let $0 < R \leqslant 1$ be an arbitrary rational number. Let $n, K, N \geqslant 1$ be large enough integers such that $K = RN$. Let C_{out} be a folded Reed-Solomon code over $\mathbb{F}_{q^n}$ of block length N and rate R. Let $C_{\text{in}}^1, \ldots, C_{\text{in}}^N$ be random linear codes over $\mathbb{F}_q$, where C_{in}^i is generated by a random $n \times n$ matrix $\mathbf{G}_i$ over $\mathbb{F}_q$ and the random choices for $\mathbf{G}_1, \ldots, \mathbf{G}_N$ are all independent.*[7] *Then the concatenated code $C^* = C_{\text{out}} \circ (C_{\text{in}}^1, \ldots, C_{\text{in}}^N)$ is a $\left(1 - R - \varepsilon, \left(\frac{N}{\varepsilon^2}\right)^{O\left(\varepsilon^{-2}\log(1/R)\right)}\right)_{led}$-list decodable code with probability at least $1 - q^{-\Omega(nN)}$ over the choices of $\mathbf{G}_1, \ldots, \mathbf{G}_N$. Further, C^* has rate R w.h.p.*

To set up the proof of the theorem above, we begin by collecting certain definitions and results from [5]. The following notion of independence will be crucial.

Definition 3 (Independent tuples)**.** *Let C be a code of block length N and rate R defined over $\mathbb{F}_{q^k}$. Let $J \geqslant 1$ and $0 \leqslant d_1, \ldots, d_J \leqslant N$ be integers. Let $\mathbf{d} = \langle d_1, \ldots, d_J \rangle$. An ordered tuple of codewords $(\mathbf{c}^1, \ldots, \mathbf{c}^J)$, $\mathbf{c}^j \in C$ is said to be $(\mathbf{d}, \mathbb{F}_q)$-independent if the following holds. $d_1 = \mathrm{WT}(\mathbf{c}^1)$ and for every $1 < j \leqslant J$, d_j is the number of positions i such that c_i^j is $\mathbb{F}_q$-independent of the vectors $\{c_i^1, \ldots, c_i^{j-1}\}$, where $\mathbf{c}^\ell = (c_1^\ell, \ldots, c_N^\ell)$.*

Note that for any tuple of codewords $(\mathbf{c}^1, \ldots, \mathbf{c}^J)$ there exists a unique $\mathbf{d}$ such that it is $(\mathbf{d}, \mathbb{F}_q)$-independent. The next two results will be crucial in the proof of our second main result.

Lemma 6 ([5])**.** *Let $\varepsilon > 0$ and let C be a folded Reed-Solomon code of block length N and rate $0 < R < 1$ that is defined over $\mathbb{F}_Q$, where $Q = q^k$. For any L-tuple of codewords from C, where $L \geqslant J \cdot (N/\varepsilon^2)^{O\left(\varepsilon^{-1} J \log(q/R)\right)}$, there exists a sub-tuple of J codewords such that the J-tuple is $(\mathbf{d}, \mathbb{F}_q)$-independent, where $\mathbf{d} = \langle d_1, \ldots, d_J \rangle$ with $d_j \geqslant (1 - R - \varepsilon)N$, for every $1 \leqslant j \leqslant J$.*

[7] We stress that we do *not* require that the $\mathbf{G}_i$'s have rank n.

Lemma 7 ([5])**.** *Let C be a folded Reed-Solomon code of block length N and rate $0 < R < 1$ that is defined over $\mathbb{F}_Q$, where $Q = q^k$. Let $J \geqslant 1$ and $0 \leqslant d_1, \ldots, d_J \leqslant N$ be integers and define $\mathbf{d} = \langle d_1, \ldots, d_J \rangle$. Then the number of $(\mathbf{d}, \mathbb{F}_q)$-independent tuples in C is at most*

$$q^{NJ(J+1)} \prod_{j=1}^{J} Q^{\max(d_j - N(1-R)+1, 0)} .$$

Given the outer code C_{out} and the inner codes C_{in}^i, recall that for every codeword $\mathbf{u} = (\mathbf{u}_1, \ldots, \mathbf{u}_N) \in C_{\text{out}}$, the codeword $\mathbf{u}\mathbf{G} \stackrel{def}{=} (\mathbf{u}_1\mathbf{G}_1, \mathbf{u}_2\mathbf{G}_2, \ldots, \mathbf{u}_N\mathbf{G}_N)$ is in $C^* = C_{\text{out}} \circ (C_{\text{in}}^1, \ldots, C_{\text{in}}^N)$, where the operations are over $\mathbb{F}_q$.

We now begin with the proof. The fact that C^* has rate R w.h.p. follows the argument used in [5] and is omitted.

Define $Q = q^k$. Let L be the worst-case list size that we are aiming for (we will fix its value at the end). By Lemma 6, any $L+1$-tuple of C_{out} codewords $(\mathbf{u}^0, \ldots, \mathbf{u}^L) \in (C_{\text{out}})^{L+1}$ contains at least $J = \left\lfloor (L+1)/(N/\gamma^2)^{O\left(\gamma^{-1} J \log(q/R)\right)} \right\rfloor$ codewords that form a $(\mathbf{d}, \mathbb{F}_q)$-independent tuple, for some $\mathbf{d} = \langle d_1, \ldots, d_J \rangle$, with $d_j \geqslant (1 - R - \gamma)N$ for all $1 \leqslant j \leqslant J$ (we will specify γ, $0 < \gamma < 1 - R$, later). Thus, to prove the theorem it suffices to show that with high probability, there is no received word $\mathbf{y} \in (\mathbb{F}_q \cup \{?\})^{nN}$ with $\textsc{wt}(\mathbf{y}) \leqslant (1 - R - \varepsilon)nN$ and J-tuple of codewords $(\mathbf{u}^1\mathbf{G}, \ldots, \mathbf{u}^J\mathbf{G})$, where $(\mathbf{u}^1, \ldots, \mathbf{u}^J)$ is a J-tuple of folded Reed-Solomon codewords that is $(\mathbf{d}, \mathbb{F}_q)$-independent, such that $\mathbf{u}^i\mathbf{G} \simeq \mathbf{y}$ for every $1 \leqslant i \leqslant J$. For the rest of the proof, we will call a J-tuple of C_{out} codewords $(\mathbf{u}^1, \ldots, \mathbf{u}^J)$ a *good* tuple if it is $(\mathbf{d}, \mathbb{F}_q)$-independent for some $\mathbf{d} = \langle d_1, \ldots, d_J \rangle$, where $d_j \geqslant (1 - R - \gamma)N$ for every $1 \leqslant j \leqslant J$.

Define $\rho = 1 - R - \varepsilon$. Note that by the union bound, we need to show that

$$\sum_{\substack{\mathbf{y} \in (\mathbb{F}_q \cup \{?\})^{nN} \\ \textsc{wt}(\mathbf{y}) \leqslant \rho nN}} P_{\mathbf{y}} \leqslant q^{-\Omega(nN)}, \tag{2}$$

where

$$P_{\mathbf{y}} = \sum_{\text{good } (\mathbf{u}^1, \ldots, \mathbf{u}^J) \in (C_{\text{out}})^J} \mathbf{Pr}\left[\bigwedge_{i=1}^{J} \mathbf{u}^i\mathbf{G} \simeq \mathbf{y}\right].$$

For now fix a good tuple $(\mathbf{u}^1, \ldots, \mathbf{u}^J)$ that is $(\mathbf{d} = \langle d_1, \ldots, d_J \rangle, \mathbb{F}_q)$-independent. Define sets $S_i \subseteq [N]$ ($|S_i| = d_i$) to be the positions that are "witnesses" to the fact that $(\mathbf{u}^1, \ldots, \mathbf{u}^J)$ is $(\mathbf{d}, \mathbb{F}_q)$-independent.

Then the probability that a particular codeword matches the unerased positions of the received word is:

$$\mathbf{Pr}[\mathbf{u}^i\mathbf{G} \simeq \mathbf{y}] \leqslant \mathbf{Pr}[(\mathbf{u}^i\mathbf{G})_{S_i} \simeq \mathbf{y}_{S_i}]. \tag{3}$$

Further, the latter probability in inequality (3) is independent of the probability for any $j \neq i$. To see this, let E_i be the event that $(\mathbf{u}^i\mathbf{G})_{S_i} \simeq \mathbf{y}_{S_i}$.

Then note that:

$$\mathbf{Pr}\left[\bigwedge_{i=1}^{J} E_i\right] = \mathbf{Pr}\left[\bigwedge_{i=2}^{J} E_i \mid E_1\right] \cdot \mathbf{Pr}[E_1].$$

As $(\mathbf{u}^1, \ldots, \mathbf{u}^J)$ is a good tuple, this is simply:

$$= \mathbf{Pr}\left[\bigwedge_{i=2}^{J} E_i\right] \cdot \mathbf{Pr}[E_1].$$

Using induction, we get that the probability that all messages in the list match is just the product of the individual probabilities. Thus, we have:

$$\mathbf{Pr}\left[\bigwedge_{i=1}^{J} \mathbf{u}^i\mathbf{G} \simeq \mathbf{y}\right] \leqslant \mathbf{Pr}\left[\bigwedge_{i=1}^{J} (\mathbf{u}^i\mathbf{G})_{S_i} \simeq \mathbf{y}_{S_i}\right] = \prod_{i=1}^{J} \mathbf{Pr}[(\mathbf{u}^i\mathbf{G})_{S_i} \simeq \mathbf{y}_{S_i}].$$

If we let u_i be the number of unerased q-ary symbols in $\mathbf{y}_{S_i}$, then since all the $\mathbf{G}_i$ are independent random matrices:

$$\mathbf{Pr}[(\mathbf{u}^i\mathbf{G})_{S_i} \simeq \mathbf{y}_{S_i}] = q^{-u_i} \leqslant q^{-d_i n + \rho n N}.$$

Note that the reason that $(-d_i n + \rho n N) \geqslant -u_i$ is because in the worst case, all erasures occur in S_i.

We take a union bound over the number of different ways that the d_i can occur:

$$P_{\mathbf{y}} \leqslant \sum_{(1-R-\gamma)N \leqslant d_1, d_2, \cdots, d_J \leqslant N} \left(q^{NJ(J+1)} \prod_{i=1}^{J} Q^{\ \max(0, d_i - N(1-R))} \right) \prod_{i=1}^{J} q^{-d_i n + \rho n N}. \tag{4}$$

The bound in parenthesis in inequality (4) comes from Lemma 7.

Now since

$$\max(0, d_i - (1-R)N) \leqslant d_i - (1-R-\gamma)N,$$

we can rewrite this, collapsing the two products into one, as:

$$P_{\mathbf{y}} = \sum_{(1-R-\gamma)N \leqslant d_1, d_2, \cdots, d_J \leqslant N} \left(q^{NJ(J+1)} \prod_{i=1}^{J} q^{n(d_i - (1-R-\gamma)N) - d_i n + \rho n N} \right). \tag{5}$$

But since:

$$nd_i - nd_i = 0,$$

we can rewrite this again, replacing the sum with an upper bound, as

$$P_{\mathbf{y}} \leqslant N^J q^{NJ(J+1)} q^{JnN(\rho - 1 + R + \gamma)}.$$

Note that:

$$q^{NJ(J+1)} = q^{NJn\left(\frac{J+1}{n}\right)}.$$

So for $n \geqslant (J+1)/\gamma$:

$$q^{NJ(J+1)} \leqslant q^{JnN\gamma}.$$

Note also that the total number of possible received words can be bounded as follows:

$$\binom{nN}{\rho n N} \cdot q^{(1-\rho)nN} \leqslant q^{2nN}, \tag{6}$$

where the first term in the product on the left-hand side of inequality (6) is the number of ways to choose erasure locations, and the second term is the number of ways to choose symbols in the unerased positions.

Also,

$$N^J \leqslant q^{J \log N} \leqslant q^{JnN\gamma}$$

for large enough N.

After applying these bounds, we get that:

$$\mathbf{Pr}[C^* \text{ is not } (\rho, L)_{led}] \leqslant q^{2nN} q^{JnN(\rho-1+R+3\gamma)}. \tag{7}$$

Recall that we have $R = 1 - \rho - \varepsilon$ and can choose J and γ freely.

Setting

$$J \geqslant 1/\gamma$$

will make

$$q^{nN} \leqslant q^{JnN\gamma},$$

and in particular,

$$q^{2nN} \leqslant q^{JnN(2\gamma)}.$$

If we pick $\gamma = \varepsilon/10$, then our final error probability in inequality (7) will be:

$$\mathbf{Pr}[C^* \text{ is not } (\rho, L)_{led}] \leqslant q^{-\frac{\varepsilon}{2} JnN},$$

establishing the desired error bound.

Remark 1. It is easy to see that the rate of the inner codes have to be very close to 1. To see this consider the erasure pattern where ρ fraction of the outer codeword symbols are completely erased. To recover from such a situation, we need R to be close to $1 - \rho$. One could re-visit the proof above for general r and try to figure out how far away from 1 r can be. If we had $r < 1$ then in (5), the exponent within the product should read $rn(d_i - (1 - R - \gamma)) - d_i n + \rho nN$. We ultimately need $R^* = Rr = 1 - \rho - \varepsilon$. Using this and some manipulations, the exponent becomes $(1-r)(1-d_i/N) - \varepsilon + r\gamma$. The only thing that we can guarantee about d_i is that $d_i \geqslant (1-R-\gamma)N$. If we desire the ultimate error probability to be $q^{-\Omega(\varepsilon nNJ)}$, then the proof goes through only if $rR \geqslant R - O(\varepsilon)$.

4.2 Random Linear Outer Code

Theorem 8. *Let q be a prime power and let $0 < R \leqslant 1$ be an arbitrary rational. Let $n, K, N \geqslant 1$ be large enough integers such that $K = RN$. Let C_{out} be a random linear code over $\mathbb{F}_{q^n}$ that is generated by a random $K \times N$ matrix over $\mathbb{F}_{q^n}$. Let $C_{\text{in}}^1, \ldots, C_{\text{in}}^N$ be random linear codes over $\mathbb{F}_q$, where C_{in}^i is generated by a random $n \times n$ matrix $\mathbf{G}_i$ and the random choices for $C_{\text{out}}, \mathbf{G}_1, \ldots, \mathbf{G}_N$ are all independent. Then the concatenated code $C^* = C_{\text{out}} \circ (C_{\text{in}}^1, \ldots, C_{\text{in}}^N)$ is a $\left(1 - R - \varepsilon, q^{O(1/\varepsilon^2)}\right)_{led}$-list decodable code with probability at least $1 - q^{-\Omega(nN)}$ over the choices of $C_{\text{out}}, \mathbf{G}_1, \ldots, \mathbf{G}_N$. Further, with high probability, C^* has rate R.*

***Proof*.** Let $q \geqslant 2$ and $R^* = R$ be the rate of the outer code (the inner codes are chosen so that their dimension $k = n$, and therefore have rate 1).

We define a segment of a codeword in C^* as a sequence of consecutive q-ary symbols generated by one particular inner code. An assumption that we will make for the ease of analysis (and which we will remove later) is that erasures, which occur with relative rate ρ, will be equally distributed among the concatenated codeword segments. This means that in our received word $\mathbf{y}$, the result of each of the N inner code encodings will contain at most ρn erasures.

We will show that there exists some integer L such that any subset of $L+1$ distinct encoded messages has the property that they all match the non-erased segments of the received word with low probability. Then we'll apply the union bound to show that with high probability, the code meets the list decoding capacity for erasures.

Define $Q = q^k$ and $\rho = 1 - R^* - \varepsilon$. Let $J = \lfloor \log_Q(L+1) \rfloor$. Then there exists a subset of size at least J of our list (which is of size $L+1$) such that the set of messages $\{\mathbf{m}_1, \mathbf{m}_2, ...\mathbf{m}_J\}$ will be linearly independent over $\mathbb{F}_Q$. This is because there are only Q^J unique ways to form linear sums of these messages over $\mathbb{F}_Q$.

Because of this fact and because C_{out} is a random linear code, the set $\{C_{out}(\mathbf{m}_1), C_{out}(\mathbf{m}_2), \cdots, C_{out}(\mathbf{m}_J)\}$ can be treated as a set of independently chosen random vectors in $\mathbb{F}_{q^k}^N$.

Fix an s so that $1 \leqslant s \leqslant N$ and let y_s represent a particular segment of our received word. (There are N such segments over $\mathbb{F}_Q$). In our list of J outer encoded messages, we denote by i the size of the subset of these where for each outer encoded message $C_{out}(\mathbf{m}_t)$, restricted to the segment s, $C_{out}(\mathbf{m}_t)$ is the zero vector. $J - i$ is then the number of messages such that $C_{out}(\mathbf{m}_t)$ is not the zero vector when restricted to the segment s.

We can bound the probability that each of these messages match the received word at this segment, in the unerased positions, as follows:

$$\mathbf{Pr}[(C^*(\mathbf{m}_t))_s \simeq y_s] \leqslant \left(\frac{1}{q^n}\right)^i \left(1 - \frac{1}{q^n}\right)^{J-i} \cdot q^{-(1-\rho)n(J-i)}. \tag{8}$$

In the above, the relationship $(C^*(\mathbf{m}_t))_s \simeq y_s$ means that the concatenated code, on message $\mathbf{m}_t$, restricted to segment s, matches the received word y on segment s at all unerased positions.

If $(C_{out}(\mathbf{m}_t))_s = 0$, then we just assume that $(C^*(\mathbf{m}_t))_s \simeq y_s$, so this is an upper bound, and not an equality.

The first term in the RHS of (8) is the probability that i messages at this segment map to the zero vector, and the second term is the probability that $J - i$ messages map to something other than the zero vector.

The third term is the probability that those nonzero $J - i$ messages match the received word in every unerased position.

Now since

$$\left(1 - \frac{1}{q^n}\right)^{J-i} \leqslant 1,$$

we have that

$$\mathbf{Pr}[(C^*(\mathbf{m}_t))_s \simeq y_s] \leqslant q^{-(1-\rho)nJ} \cdot q^{i(1-\rho)n - in}.$$

Also, because $(1-\rho)$ is always less than 1,

$$q^{i(1-\rho)n-in} \leqslant 1.$$

Therefore

$$\mathbf{Pr}[(C^*(\mathbf{m}_t))_s \simeq y_s] \leqslant q^{-(1-\rho)nJ}.$$

The probability, then, that every message in the list matches the received word in the unerased positions for a single segment, taken over all possible choices of locations and sizes of i is then (by the union bound over such locations and sizes, noting that there are at most q^J ways to make these choices):

$$\mathbf{Pr}\left[\bigwedge_{t=1}^{J}(C^*(\mathbf{m}_t))_s \simeq y_s\right] \leqslant q^J \cdot q^{-(1-\rho)nJ}. \tag{9}$$

Recalling that each inner code is chosen independently, the probability that this is true for all segments is then

$$\mathbf{Pr}\left[\bigwedge_{t=1}^{J} C^*(\mathbf{m}_t) \simeq \mathbf{y}\right] \leqslant q^{JN} \cdot q^{-(1-\rho)nJN}.$$

Taking the union bound over all possible received words and lists of size J:

$$\mathbf{Pr}[C^* \text{ is not } (\rho, L)_{led}] \leqslant q^{nN} \cdot q^{(1-\rho)nN} \cdot q^{kKJ} \cdot q^{JN} \cdot q^{-(1-\rho)nJN}. \tag{10}$$

The first term in RHS of (10) is an upper bound on the number of possibilities for the erasure positions. The second term is the number of ways to specify the unerased positions, the third term is the number of possible lists of size J, and the fourth and fifth terms come from the previous inequality.

Since $kK = R^*nN$, and $2 > 1 + (1-\rho)$, this can be rewritten and simplified as:

$$\mathbf{Pr}[C^* \text{ is not } (\rho, L)_{led}] \leqslant q^{-nNJ\left(\frac{-2}{J} - R^* - \frac{1}{n} + (1-\rho)\right)}.$$

If we can choose n, R^*, and J appropriately so that:

$$\frac{-2}{J} - R^* - \frac{1}{n} + (1-\rho) \geqslant \varepsilon/2,$$

then this probability will be exponentially small.

Setting $n \geqslant J, J = \lceil \frac{6}{\varepsilon} \rceil$ works.

We still need to fix the assumption that the ρ fraction of erasures are all distributed equally among the N encoded segments.

Note that if we describe the fraction of erasures in each segment by ρ_s, then

$$\sum_{s=1}^{N} \rho_s n = \rho n N.$$

The per-segment probability then becomes

$$\mathbf{Pr}[(C^*(\mathbf{m}_t))_s \simeq y_s] \leqslant q^J \cdot q^{-(1-\rho_s)nJ}$$

and the probability for the entire received word becomes

$$\mathbf{Pr}\left[\bigwedge_{t=1}^{J} C^*(\mathbf{m}_t) \simeq y\right] \leqslant \prod_{s=1}^{N} q^J \cdot q^{-(1-\rho_s)nJ}.$$

Note further that the ρ_s terms can be collected in the exponent and simplified to inequality (9).

Finally, the claim that C^* has rate R follows from a similar argument to that from [5] and is omitted. ■

Acknowledgments

We thank Venkat Guruswami and Parikshit Gopalan for helpful discussions. Thanks again to Madhu Sudan for kindly allowing us to include Lemma 3 in this paper.

References

[1] I. Dumer, D. Micciancio, and M. Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Transactions on Information Theory*, 49(1):22–37, 2003.

[2] A. Gál, S. Halevi, R. J. Lipton, and E. Petrank. Computing from partial solutions. *Proceedings of the 14th Annual IEEE Conference on Computation Complexity*, pages 34–45, 1999.

[3] V. Guruswami. List decoding from erasures: Bounds and code constructions. *IEEE Transactions on Information Theory*, 49(11):2826–2833, 2003.

[4] V. Guruswami. *List decoding of error-correcting codes.* Number 3282 in Lecture Notes in Computer Science. Springer, 2004.

[5] V. Guruswami and A. Rudra. Concatenated codes can achieve list-decoding capacity. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 258–267, 2008.

[6] V. Guruswami and A. Rudra. Explicit codes achieving list decoding capacity: Error-correction up to the Singleton bound. *IEEE Transactions on Information Theory*, 54(1):135–150, January 2008. Preliminary version appeared as "Explicit capacity-achieving list-decodable codes" in *Proceedings of STOC 06.*

[7] V. Guruswami and A. Rudra. Better binary list decodable codes via multilevel concatenation. *IEEE Transactions on Information Theory*, 55(1):19–26, 2009.

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

[9] V. Guruswami and M. Sudan. List decoding algorithms for certain concatenated codes. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 181–190, 2000.

[10] T. Høholdt, J. H. van Lint, and R. Pellikaan. *Algebraic Geometry Codes.* Handbook of Coding Theory, (V.S. Pless, W.C. Huffamn and R.A. Brualdi Eds.), Elsevier, 1998.

[11] A. Kiayias and M. Yung. Cryptographic hardness based on the decoding of reed-solomon codes. *IEEE Transactions on Information Theory*, 54(6):2752–2769, 2008.

[12] S. R. Kumar and D. Sivakumar. Proofs, codes, and polynomial-time reducibilities. *Proceedings of the 14th Annual IEEE Conference on Computation Complexity*, 1999.

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

[14] R. J. McEliece. On the average list size for the Guruswami-Sudan decoder. In *7th International Symposium on Communications Theory and Applications (ISCTA)*, July 2003.

[15] F. Parvaresh and A. Vardy. Correcting errors beyond the Guruswami-Sudan radius in polynomial time. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 285–294, 2005.

[16] A. Rudra. *List Decoding and Property Testing of Error Correcting Codes*. PhD thesis, University of Washington, 2007.

[17] A. Rudra. Limits to list decoding random codes. In *Proceedings of the 15th Annual International Conference on Computing and Combinatorics (COCOON)*, pages 27–36, 2009.

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

[19] M. Sudan. List decoding: Algorithms and applications. *SIGACT News*, 31:16–27, 2000.

[20] M. Sudan. Algorithmic introduction to coding theory, 2001. Lecture Notes available at `http://people.csail.mit.edu/madhu/FT01/`.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de