



Tight Bounds on List-Decodable and List-Recoverable Zero-Rate Codes

Nicolas Resch   

Informatics' Institute, University of Amsterdam, The Netherlands

Chen Yuan  

School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China

Yihan Zhang   

Institute of Science and Technology Austria, Klosterneuburg, Austria

Abstract

In this work, we consider the list-decodability and list-recoverability of codes in the *zero-rate regime*. Briefly, a code $\mathcal{C} \subseteq [q]^n$ is (p, ℓ, L) -list-recoverable if for all tuples of input lists (Y_1, \dots, Y_n) with each $Y_i \subseteq [q]$ and $|Y_i| = \ell$, the number of codewords $c \in \mathcal{C}$ such that $c_i \notin Y_i$ for at most pn choices of $i \in [n]$ is less than L ; list-decoding is the special case of $\ell = 1$. In recent work by Resch, Yuan and Zhang (ICALP 2023) the zero-rate threshold for list-recovery was determined for all parameters: that is, the work explicitly computes $p_* := p_*(q, \ell, L)$ with the property that for all $\varepsilon > 0$ (a) there exist positive-rate $(p_* - \varepsilon, \ell, L)$ -list-recoverable codes, and (b) any $(p_* + \varepsilon, \ell, L)$ -list-recoverable code has rate 0. In fact, in the latter case the code has *constant* size, independent on n . However, the constant size in their work is quite large in $1/\varepsilon$, at least $|\mathcal{C}| \geq (\frac{1}{\varepsilon})^{O(q^L)}$.

Our contribution in this work is to show that for all choices of q, ℓ and L with $q \geq 3$, any $(p_* + \varepsilon, \ell, L)$ -list-recoverable code must have size $O_{q, \ell, L}(1/\varepsilon)$, and furthermore this upper bound is complemented by a matching lower bound $\Omega_{q, \ell, L}(1/\varepsilon)$. This greatly generalizes work by Alon, Bukh and Polyanskiy (IEEE Trans. Inf. Theory 2018) which focused only on the case of binary alphabet (and thus necessarily only list-decoding). We remark that we can in fact recover the same result for $q = 2$ and even L , as obtained by Alon, Bukh and Polyanskiy: we thus strictly generalize their work.

Our main technical contribution is to (a) properly define a linear programming relaxation of the list-recovery condition over large alphabets; and (b) to demonstrate that a certain function defined on a q -ary probability simplex is maximized by the uniform distribution. This represents the core challenge in generalizing to larger q (as a binary simplex can be naturally identified with a one-dimensional interval). We can subsequently re-utilize certain Schur convexity and convexity properties established for a related function by Resch, Yuan and Zhang along with ideas of Alon, Bukh and Polyanskiy.

2012 ACM Subject Classification Mathematics of computing \rightarrow Coding theory

Keywords and phrases List Decoding, List Recovery, Zero Rate

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.82

Related Version *Full Version*: <https://arxiv.org/abs/2309.01800>

Funding The research of C. Yuan was supported in part by the National Key R&D Program of China under Grant 2023YFE0123900 and Natural Science Foundation of Shanghai under the 2024 Shanghai Action Plan for Science, Technology and Innovation Grant 24BC3200700. The research of N. Resch is supported in part by an NWO (Dutch Research Council) grant with number C.2324.0590, and this work was done in part while he was visiting the Simons Institute for the Theory of Computing, supported by DOE grant #DE-SC0024124.

1 Introduction

Given an error-correcting code $\mathcal{C} \subseteq [q]^n$, a fundamental requirement is that the codewords are sufficiently well-spread in order to guarantee some non-trivial correctability properties. This is typically enforced by requiring that the minimum distance of the code $d = \min\{d_H(c, c') :$



© Nicolas Resch, Chen Yuan, and Yihan Zhang;

licensed under Creative Commons License CC-BY 4.0

16th Innovations in Theoretical Computer Science Conference (ITCS 2025).

Editor: Raghu Meka; Article No. 82; pp. 82:1–82:21



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$\mathbf{c} \neq \mathbf{c}' \in \mathcal{C}$ }, where $d_{\text{H}}(\cdot, \cdot)$ denotes the Hamming distance (i.e. the number of coordinates on which two strings differ). Note that minimum distance d is equivalent to the following “packing” property: if we put a ball of radius $r := \lfloor d/2 \rfloor$ around any point $\mathbf{z} \in [q]^n$ – i.e. we consider the Hamming ball $\mathcal{B}_{\text{H}}(\mathbf{y}, r) := \{\mathbf{x} \in [q]^n : d_{\text{H}}(\mathbf{x}, \mathbf{y}) \leq r\}$ – then all these balls contain at most 1 codeword from \mathcal{C} .

This latter viewpoint can easily be generalized to obtain *list-decodability*, where we now require that such Hamming balls do not capture “too many” codewords. That is, for $p \in [0, 1]$ and $L \in \mathbb{N}$ a code is called (p, L) -*list-decodable* if every Hamming ball of radius pn contains less than L codewords from \mathcal{C} . In notation: for all $\mathbf{y} \in [q]^n$, $|\mathcal{B}_{\text{H}}(\mathbf{y}, pn) \cap \mathcal{C}| \leq L - 1$.¹ This notion was already introduced in the 50’s by Elias and Wozencraft [10, 30, 11] but has in the past 20 years seen quite a bit of attention due to its connections to other parts of theoretical computer science [13, 2, 21, 19, 18, 26].

One can push this generalization further to obtain *list-recoverability*. Here, we consider a tuple of input lists $\mathbf{Y} = (Y_1, \dots, Y_n)$, where each $Y_i \subseteq [q]$ has size at most ℓ (for some $\ell \in \mathbb{N}$). The requirement is that the number of codewords that “disagree” with \mathbf{Y} in at most pn coordinates is at most $L - 1$. More formally, if for all $\mathbf{Y} = (Y_1, \dots, Y_n)$ the number of codewords $\mathbf{c} \in \mathcal{C}$ such that $|\{i \in [n] : c_i \notin Y_i\}| \leq pn$ is at most $L - 1$, the code is called (p, ℓ, L) -*list-recoverable*. Note that (p, L) -list-decodability is nothing other than $(p, 1, L)$ -list-recoverability. Initially, list-recoverability was abstracted as a useful stepping stone towards list-decoding concatenated codes. However, in recent years this notion has found many connections to other parts of computer science, e.g. in cryptography [15, 16], randomness extraction [14], hardness amplification [8], group testing [17, 23], streaming algorithms [9], and beyond.

Rate versus noise-resilience. Having fixed a desired “error-tolerance” as determined by the parameters p, ℓ and L we would also like the code \mathcal{C} to be as large as possible: intuitively, this implies that the code contains the minimal amount of redundancy possible. A fundamental question in coding theory is to understand the achievable tradeoffs between the rate $R := \frac{\log_q |\mathcal{C}|}{n}$ and some “error-resilience” property of the code, e.g., minimum distance, list-decodability, or list-recoverability.

This question in full generality is wide open. Even for the special case of $q = 2$ and $L = 2$ (i.e. determining the optimal tradeoff between rate and distance for binary codes) is unclear: on the possibility side we have the Gilbert-Varshamov bound [12, 28] showing $R \geq 1 - H_2(p/2)$ is achievable (here, $H_2(x) = -x \log_2 x - (1 - x) \log_2 (1 - x)$ is the binary entropy function), while bounds of Elias and Bassalygo [3] and the linear programming bound [29, 22, 7] give incomparable and non-tight upper bounds. None of these bounds have been substantially improved in at least 40 years. The situation is even more complicated for larger q : for $q = 49$ (and larger prime powers) the celebrated algebraic geometry codes of Tsfasman, Vlăduț and Zink [27] provide explicit codes of higher rate in certain regimes than those promised by the Gilbert-Varshamov bound.

When one relaxes the question to allow an asymptotically growing list size L then we do have a satisfactory answer: the answer is provided by the list-decoding/-recovery theorem, which states that for all $\varepsilon > 0$ there exist $(p, \ell, O(1/\varepsilon))$ -list-recoverable codes of rate $1 - H_{q, \ell}(p)$ where

¹ Typically the upper bound is L , rather than $L - 1$. However, for “impossibility” arguments this parametrization is more common, as it leads to less cumbersome computations.

$$H_{q,\ell}(x) := p \log_q \left(\frac{q-\ell}{p} \right) + (1-p) \log_q \left(\frac{\ell}{1-p} \right)$$

is (q, ℓ) -ary entropy function [24].² On the other hand, any code of rate $R \geq 1 - h_{q,\ell}(p)$ fails to be (p, ℓ, L) -list-recoverable unless $L \geq q^{\Omega(\varepsilon n)}$. However, this does not provide very meaningful bounds if one is interested in, say, $(p, 2, 5)$ -list-recoverable codes.

Positive versus zero-rate regimes. Thus far, we have implicitly been discussing the *positive-rate regime*. However, one can also ask questions about the behaviour of codes in the *zero-rate regime*. For context, recent work by Resch, Yuan and Zhang [25] computed the zero-rate threshold for list-recovery: that is, for all alphabet sizes $q \geq 2$, input list sizes ℓ and output list size L , they determine the value $p_*(q, \ell, L)$ such that (a) for all $p < p_*(q, \ell, L)$ there exist infinite families of positive rate (p, ℓ, L) -list-recoverable codes over the alphabet $[q]$, and (b) for all $p > p_*(q, \ell, L)$ there does not exist such an infinite family.

Having now delineated the “positive rate” and the “zero-rate” regimes depending on how p compares to $p_*(q, \ell, L)$, in this work we study the zero-rate regime for list-recoverable codes for all alphabet sizes q . In [25], it is shown that (p, ℓ, L) -list-recoverable codes $\mathcal{C} \subseteq [q]^n$ with $p = p_*(q, \ell, L) + \varepsilon$ have constant size (that is, independent of the block length n); however, this constant is massive in the parameters due to the use of a Ramsey-theoretic bound. In particular, the dependence on ε is at least $(1/\varepsilon)^{2q^L}$, and this is additionally multiplied by a tower of 2’s of height roughly L .

To the best of our knowledge, prior work on this question focuses exclusively on the $q = 2$ case. For example, in the case of $L = 2$ (i.e., unique-decoding) we have $p_*(2, 1, 2) = 1/4$, and work by Levenshtein [20] shows that a code construction based on Hadamard matrices corrects a $1/4 + \varepsilon$ fraction of errors and has size $1/(4\varepsilon) + O(1)$. A particularly relevant prior work is due to Alon, Bukh and Polyanskiy [1]. Herein the authors consider this question for the special case of $q = 2$ (and thus, necessarily, only for list-decoding). In particular, they show that when L is even if $p = p_*(2, 1, L) + \varepsilon$ then such a (p, L) -list-decodable code $\mathcal{C} \subseteq [2]^n$ has size at most $O_L(1/\varepsilon)$, and moreover provide a construction of such a code with size $\Omega_L(1/\varepsilon)$.³ They observe some interesting behaviour in the case of odd L ; in particular, the maximum size of a $(p_*(2, 1, 3) + \varepsilon, 3)$ -list-decodable code is $\Theta(1/\varepsilon^{3/2})$.⁴

Our motivations for this investigation are three-fold. Firstly, the zero-rate regime offers combinatorial challenges and interesting behaviours that we uncover in this work. Secondly, many codes that find applications in other areas of theoretical computer in fact have subconstant rate. Lastly, the zero-rate regime appears much more tractable than the positive rate regime – indeed, we can obtain tight upper and lower bounds on the size of a code, as we will soon see. It would be interesting to determine to what extent such techniques could be useful for understanding the positive rate regime as well.

² Note that setting $\ell = 1$ recovers the standard q -ary entropy function, which itself reduces to the binary entropy function upon setting $q = 2$.

³ Note that for the special case of $q = 2$, the zero-rate threshold for list-decoding had already been established by Blinovskiy [4].

⁴ This argument in fact shows a flaw in an earlier claimed proof of Blinovskiy that claimed such codes have size $O_L(1/\varepsilon)$ for all $L \in \mathbb{N}$.

1.1 Our results

Our main result in this work is a tight bound on the size of a (p, ℓ, L) -list-recoverable code over an alphabet of size $q \geq 3$ when $p > p_*(q, \ell, L)$. The main technical challenge is to compute the following upper bound on the size of such a code.

► **Theorem 1.** *Let $q, \ell, L \in \mathbb{N}$ with $q \geq 3$. $\ell < q$ and $L > \ell$ be fixed constants. Let $\varepsilon > 0$ and put $p = p_*(q, \ell, L) + \varepsilon$. Suppose $\mathcal{C} \subseteq [q]^n$ is (p, ℓ, L) -list-recoverable. Then $|\mathcal{C}| \leq O_{q, \ell, L}(1/\varepsilon)$.*

We complement the above negative result with the following code construction, showing the upper bound is tight.

► **Theorem 2.** *Let $q, \ell, L \in \mathbb{N}$ with $q \geq 3$ and $\ell < q$ be fixed constants. Let $\varepsilon > 0$ and put $p = p_*(q, \ell, L) + \varepsilon$. There exists a (p, ℓ, L) -list-recoverable code $\mathcal{C} \subseteq [q]^n$ such that $|\mathcal{C}| \geq \Omega_{q, \ell, L}(1/\varepsilon)$.*

We emphasize that in the above theorems the implied constants may depend on q, ℓ and L .

Note that our results explicitly *exclude* the case of $q = 2$. As [1] prove, the binary alphabet behaves in subtle ways: the bound on the code size depends on the parity of L . Intriguingly, our work demonstrates that such behaviour does not arise over larger alphabets.

1.2 Technical Overview

The double-counting argument. Since our focus is on zero-rate list-decodable/-recoverable codes, it helps to first review the proof of the zero-rate threshold $p_*(q, \ell, L)$. A lower bound can be easily obtained by a random construction that attains a positive rate for any $p \leq p_*(q, \ell, L) - \varepsilon$. For the upper bound, let us first consider the list-decoding case, i.e., $\ell = 1$. The proof in [5, 6, 25], at a high-level, proceeds via a double-counting argument.⁵ For any (p, ℓ, L) -list-decodable code $\mathcal{C} \subset [q]^n$, the proof aims to upper and lower bound the *radius* of a list averaged over the choice of the list from \mathcal{C} :

$$\frac{1}{M^L} \sum_{(\mathbf{c}_1, \dots, \mathbf{c}_L) \in \mathcal{C}^L} \text{rad}_H(\mathbf{c}_1, \dots, \mathbf{c}_L). \quad (1)$$

Comparing the bounds produces an upper bound on $|\mathcal{C}|$. Here $\text{rad}_H(\cdot)$, known as the Chebyshev radius of a list, is the relative radius of the smallest Hamming ball containing all codewords in the list. A lower bound on Equation (1) essentially follows from list-decodability of \mathcal{C} . Indeed, each term (corresponding to lists consisting of distinct codewords) is lower bounded by p , otherwise a list that fits into a ball of radius at most np is found, violating list-decodability of \mathcal{C} . Therefore Equation (1) is at least $p - o(1)$, where $o(1)$ is to account for lists with not-all-distinct codewords.

On the other hand, it is much more tricky to upper bound Equation (1) as, in general, rad_H admits no analytically closed form and can only be computed by solving a min-max problem. Previous proofs [25] then first extract a subcode \mathcal{C}' with highly-regular list structures via the hypergraph Ramsey's theorem. This allows one to assert that all lists have essentially the same radius and all codewords in each list have essentially the same distance to the center of the list. As a result, the min-max expression is “linearized” and Equation (1) can be upper bounded when restricted to \mathcal{C}' . The downside is that the Ramsey reduction step is rather lossy for code size.

⁵ A characterization of $p_*(q, 1, L)$ was announced in [5, 6] whose proof was flawed. The work [25] filled in the gaps therein and characterized $p_*(q, \ell, L)$ for general ℓ .

Weighted average radius. The effect of the Ramsey reduction, put formally, is to enforce the *average radius*:

$$\overline{\text{rad}}_{\text{H}}(\mathbf{c}_1, \dots, \mathbf{c}_L) := \frac{1}{n} \min_{\mathbf{r} \in \{0,1\}^n} \frac{1}{L} \sum_{i=1}^L d_{\text{H}}(\mathbf{c}_i, \mathbf{r}) \quad (2)$$

of every list in the subcode to be approximately equal. To extract the regularity structures in lists without resorting to extremal bounds from Ramsey theory, [1] introduced the notion of *weighted average radius* which “linearizes” the Chebyshev radius in a *weighted* manner:

$$\overline{\text{rad}}_{\omega}(\mathbf{c}_1, \dots, \mathbf{c}_L) := \frac{1}{n} \min_{\mathbf{r} \in \{0,1\}^n} \sum_{i=1}^L \omega(i) d_{\text{H}}(\mathbf{c}_i, \mathbf{r})$$

where ω is a distribution on L elements. For any weighting ω , $\overline{\text{rad}}_{\omega}$ of lists from the code forms a suite of succinct statistics of the list distribution. It turns out $\overline{\text{rad}}_{U_L} = \overline{\text{rad}}$ (where U_L denotes the uniform distribution on $[L]$) is maximal under all ω . Recall that the double-counting argument suggests that in an optimal zero-rate code, the behaviour of the ensemble average of rad is essentially captured by that of $\overline{\text{rad}}$. In particular, list-decodability ensures that $\overline{\text{rad}}$ of *most* lists should be large. However, not too many lists in an optimal code are expected to have large $\overline{\text{rad}}_{\omega}$ for any $\omega \neq U_L$. [1] then managed to quantify the gap between $\overline{\text{rad}} = \overline{\text{rad}}_{U_L}$ and $\overline{\text{rad}}_{\omega}$ (with $\omega \neq U_L$), which yields improved (and sometimes optimal) size-radius trade-off of zero-rate codes.

Generalization to q -ary list-decoding. Our major technical contribution is in extrapolating the above ideas to list-recovery. The challenge lies particularly in defining a proper notion of weighted average radius and proving its properties. Our definition relies crucially on an embedding φ from $[q]$ to the simplex in \mathbb{R}^q and relaxes the center \mathbf{r} of the list to be a fractional vector. Specifically, denoting by $\Delta = \{\mathbf{a} \in \mathbb{R}_{\geq 0}^q : \sum_{i=1}^q a_i = 1\}$ the simplex in \mathbb{R}^q and $\partial\Delta = \{\mathbf{e}_1, \dots, \mathbf{e}_q\}$ its vertices (i.e., the standard basis of \mathbb{R}^q), we let the embedding φ map each symbol $x \in [q]$ to the standard basis vector $\mathbf{e}_x \in \partial\Delta$. Denoting by $\mathbf{x}_1, \dots, \mathbf{x}_L \in (\partial\Delta)^n$ the (element-wise) images of a list $\mathbf{c}_1, \dots, \mathbf{c}_L \in [q]^n$, we define the *weighted average radius* of $\mathbf{x}_1, \dots, \mathbf{x}_L$ as:

$$\overline{\text{rad}}_{\omega}(\mathbf{x}_1, \dots, \mathbf{x}_L) = \frac{1}{n} \min_{\mathbf{y} \in \Delta^n} \frac{1}{2} \mathbb{E}_{i \sim \omega} [\|\mathbf{x}_i - \mathbf{y}\|_1], \quad (3)$$

where ω is any distribution on $[L]$.

The nontriviality and significance of the above notion, especially the embedding used therein, is three-fold.

- First, as the weighting ω varies, $\overline{\text{rad}}_{\omega}$ serves as a bridge between the standard average radius in Equation (2) and the Chebyshev radius. Indeed, $\omega = U_L$ recovers the former, and the maximum $\overline{\text{rad}}_{\omega}$ over ω recovers the latter. However, we caution that the second statement does not hold without the embedding since the Hamming distance between q -ary symbols per se is not and cannot be interpolated by a convex function, which makes the minimax theorem inapplicable. Fortunately, our embedding affinely extends the q -ary Hamming distance to the simplex therefore brings back the applicability of the minimax theorem and connects $\max_{\omega} \overline{\text{rad}}_{\omega}$ to rad .
- Second, our definition in Equation (3) allows \mathbf{y} to take any value on the simplex, instead of only its vertices, i.e., the image of $[q]$ under φ . Though embedding naively to the hypercube $[0,1]^q$ seems convenient, upon solving the expression with fractional \mathbf{y} one

does not necessarily obtain a notion that is guaranteed to closely approximate the original version with integral \mathbf{y} . In contrast, using linear programming duality, we show that our embedding yields relaxed notion of radius which closely approximates the actual Chebyshev radius. Indeed, upon rounding the fractional center \mathbf{y} and taking its preimage under φ , our results guarantee that the resulting radius must have negligible difference from the Chebyshev radius. Precisely speaking, we want to find a vector $\mathbf{y} = (y(i, j))_{[n] \times [q]} \in \Delta$ close to the L images of the codewords $\mathbf{x}_1, \dots, \mathbf{x}_L$ by linear programming. Meanwhile, we want $\mathbf{y}(i) := (y(i, 1), \dots, y(i, q))$ to belong to $\partial\Delta$ so that we can find a preimage of $\mathbf{y}(i)$ in $[q]$. Since $\mathbf{y}(i) \in \Delta$, the components in $\mathbf{y}(i)$ are subject to $\sum_{j=1}^q y(i, j) = 1$. This implies that at least one component of $\mathbf{y}(i)$ is nonzero. The basic feasible solution guarantees that there exists a feasible solution such that most of $y(i, j)$ are 0. Combining with the fact $\sum_{j=1}^q y(i, j) = 1$ forces $(y(i, 1), \dots, y(i, q)) \in \partial\Delta$ for almost all $i \in [n]$. Thus, we obtain a negligible loss in the conversion between Hamming distance and Euclidean distance.

- Finally, under the embedding φ , the weighted average radius $\overline{\text{rad}}_\omega$ still retains the appealing feature that the minimization can be analytically solved, therefore giving rise to an explicit expression (see Equation (14)) which greatly facilitates our analysis.

We then show, via techniques deviating from those in [1], three key properties that are required by the subsequent arguments.

1. For any fixed distribution P , if entries of codewords in the list are generated i.i.d. using P , then

$$f(P, \omega) := \mathbb{E}_{(X_1, \dots, X_L) \sim P^{\otimes L}} \left[1 - \max_{x \in [q]} \sum_{\substack{i \in [L] \\ X_i = x}} \omega(i) \right]$$

is maximized when $\omega = U_L$. Moreover, the equality holds if and only if $\omega = U_L$ for $q \geq 3$ and any L . Our approach is different from [1] as we can not explicitly represent function $f(P, \omega)$.

2. Furthermore, if entries of codewords in the list are generated i.i.d. using a certain P , then $f(P, U_L)$ is upper bounded by $f(P_{q,p}, U_L)$ with $P_{q,p} = (\frac{1-p}{q}, \dots, \frac{1-p}{q}, p)$ and $p = \max_{i \in [q]} P(i)$. This follows from the Schur convexity property proved in [25].
3. Finally, denoting by P_i the distribution of the i -th components of codewords in code \mathcal{C} , Schur convexity promises $f(P_i, U_L) \leq f(P_{q,p_i}, U_L)$. In [25], it is proved that $f(P_{q,p}, U_L)$ is convex for $p \in [1/q, 1]$. Thus, we can conclude that

$$\frac{1}{n} \sum_{i \in [n]} f(P_i, U_L) \leq f(P_{q,p}, U_L)$$

with $p = \frac{1}{n} \sum_{i \in [n]} p_i$.

The remaining part of our proof is similar to [1]. We show that a code \mathcal{C} either has radius

$$\text{rad}(\mathcal{C}) = \frac{1}{n} \min_{\mathbf{x} \in [q]^n} \max_{\mathbf{c} \in \mathcal{C}} d_H(\mathbf{c}, \mathbf{x}) \leq 1 - \frac{1}{q} - \delta$$

or most of L -tuples with distinct codewords in \mathcal{C} are distributed close to uniform,. For the former case, we use the convexity property to show that the list-decodability of \mathcal{C} can not exceed $f(U_q, U_L) = p_*(q, L)$ by much. For the latter case, since most of L -tuples of distinct codewords in \mathcal{C}^L looks uniformly at random, we can show that the list-decodability of \mathcal{C} is very close to that of random codes which is $f(U_q, U_L)$.

Generalization to list-recovery. For list-recovery, i.e., $\ell > 1$, we find an embedding φ_ℓ that maps each element in $[q]$ to a superposition of ℓ vertices of the simplex in \mathbb{R}^q , i.e., we map the element in $[q]$ to a vector space $[0, 1]^{\mathcal{X}}$ where $\mathcal{X} = \binom{[q]}{\ell}$ is the collection of all ℓ -subsets in $[q]$. Concretely, we define $\varphi_\ell(i) := \sum_{A \in \mathcal{X}, i \in A} \mathbf{e}_A$ where $(\mathbf{e}_A)_{A \in \mathcal{X}}$ is a standard basis of $\mathbb{R}^{\mathcal{X}}$. The intuition behind this map is that if $i \in X$, we have $\|\varphi_\ell(i) - \mathbf{e}_X\|_1 = \binom{q}{\ell} - 1$ and otherwise $\|\varphi_\ell(i) - \mathbf{e}_X\|_1 = \binom{q}{\ell} + 1$. Similar to the list decoding, given L codewords in $[q]^n$, we obtain L vectors $\mathbf{x}_1, \dots, \mathbf{x}_L$ under the map φ_ℓ . Our goal is to find a vector $\mathbf{y} = (y(i, A))_{[n] \times \mathcal{X}}$ close to these L vectors subject to the constraint that $\sum_{A \in \mathcal{X}} y(i, A) = 1$ for any $i \in [n]$. This constraint combined with the basic feasible solution argument forces that for almost all $i \in [n]$, $(y(i, A))_{A \in \mathcal{X}}$ is of the form \mathbf{e}_X . For such i , we can find an ℓ -subset $X \in \mathcal{X}$ preserving the distance, i.e.,

$$d_{\text{LR}}(i, X) = \mathbb{1}\{i \notin X\} = \frac{1}{2} \left(\|\varphi_\ell(i) - \mathbf{e}_X\|_1 - \binom{q}{\ell} + 1 \right).$$

Besides the linear programming relaxation, further adjustments for the proof of properties analogous to Items 1–3 above are required.

Code construction. As alluded to before, a code that saturates the optimal size-radius trade-off should essentially saturate both the upper and lower bounds on the quantity

$$\frac{1}{M^L} \sum_{(\mathbf{c}_1, \dots, \mathbf{c}_L) \in \mathcal{C}^L} \overline{\text{rad}}_{\text{H}}(\mathbf{c}_1, \dots, \mathbf{c}_L)$$

considered in the double-counting argument. Indeed, our impossibility result implies that any optimal zero-rate code must contain a large fraction of random-like L -tuples $(\mathbf{c}_1, \dots, \mathbf{c}_L)$, i.e., for every $\mathbf{u} \in [q]^L$

$$\sum_{i=1}^n \mathbb{1}\{(\mathbf{c}_1(i), \dots, \mathbf{c}_L(i)) = \mathbf{u}\} \approx \frac{n}{q^L} \quad (4)$$

where $\mathbf{c}_j = (\mathbf{c}_j(1), \dots, \mathbf{c}_j(n)) \in [q]^n$. To match such an impossibility result, an optimal construction should contain as many such L -tuples as possible. A simplex-like code then becomes a natural candidate. This is a natural extension of the construction in [1] to larger alphabet. An $M \times n$ codebook \mathcal{C} consisting of M codewords each of length n is constructed by putting as columns all possible distinct length- M vectors that contains identical numbers of $1, 2, \dots, q$. It is not hard to see by symmetry that (4) becomes equality for every L -tuple with distinct codewords in \mathcal{C} . Thus, \mathcal{C} is the most regular code.

We also remark that, unlike for positive-rate codes, the prototypical random construction (with expurgation) does not lead to favorable size-radius trade-off since the deviation of random sampling is comparatively too large in the zero-rate regime. In contrast, the simplex code is deterministically regular and has no deviation.

1.3 Organization

The remainder is organized as follows. First, Section 2 provides the necessary notations and definitions, together with some preliminary results which will be useful in the subsequent arguments. Sections 3.1–3.4 contain our argument establishing Theorem 1 for list-decoding (i.e. the case $\ell = 1$); in Section 4 we elucidate the changes that need to be made to establish the theorem for general ℓ . Next, Section 5 provides the code construction establishing Theorem 2.

2 Preliminaries

Firstly, for convenience of the reader we begin by summarizing the notation that we use. This is particularly relevant as we will often be in situations where we need multiple indexes for, e.g., lists of vectors where each coordinate lies in a probability simplex, so the reader is encouraged to refer to this table whenever it is unclear what is intended.

■ **Table 1** Notation for list-decoding.

English letter in boldface	$[q]^n$ -valued vector
Greek letter in boldface	$\Delta([q])$ -valued vector
$\Delta := \Delta([q])$	Simplex in $[0, 1]^q$, i.e., $\Delta = \{(x_1, \dots, x_q) \in [0, 1]^q : \sum_{i=1}^q x_i = 1\}$
$\partial\Delta$	Set of vertices of Δ
$e_x \in \partial\Delta$	The image of $x \in [q]$ under φ , i.e., the x -th vertex of Δ
$c_i \in [q]^n$	The i -th codeword in a list
$x_i \in \Delta^n$	Image of c_i under φ (applied component-wise)
$y \in \Delta^n$	Relaxed center of a list
$x(j) \in \partial\Delta, y(j) \in \Delta$	The j -th block (of length q) in $x \in (\partial\Delta)^n, y \in \Delta^n$, respectively
$x(j, k) \in \{0, 1\}, y(j, k) \in [0, 1]$	The (j, k) -th element of $x \in (\partial\Delta)^n, y \in \Delta^n$, respectively
rad_H	(Standard) Chebyshev radius
$\overline{\text{rad}}$	Relaxed Chebyshev radius
$\overline{\text{rad}}_\omega$	Average radius
$f(P, \omega)$	Average radius weighted by $\omega \in \Delta([L])$
$(X_1, \dots, X_L) \sim P^{\otimes L}$	Expected average radius (weighted by ω) of P -distributed symbols
U_k	A list of i.i.d. P -distributed symbols
	Uniform distribution on $[k]$

For a finite set S and an integer $0 \leq k \leq |S|$, we denote $\binom{S}{k} := \{T \subset S : |T| = k\}$. Let $[q] = \{1, \dots, q\}$.

2.1 List-Decoding

Fix $q \in \mathbb{Z}_{\geq 3}$ and $L \in \mathbb{Z}_{\geq 2}$. Let $d_H(\mathbf{c}, \mathbf{r})$ denote the *Hamming distance* between $\mathbf{c}, \mathbf{r} \in [q]^n$, i.e., the number of coordinates on which the strings differ. For $t \in [0, n]$, let $\mathcal{B}_H(\mathbf{y}, t) := \{\mathbf{c} \in [q]^n : d_H(\mathbf{c}, \mathbf{y}) \leq t\}$ denote the *Hamming ball* centered around \mathbf{y} of radius $\lfloor t \rfloor$.

► **Definition 3** (List-decodable code). *Let $p \in [0, 1]$. A code $\mathcal{C} \subseteq [q]^n$ is $(p, L)_q$ -list-decodable if for any $\mathbf{y} \in [q]^n$,*

$$|\mathcal{C} \cap \mathcal{B}_H(\mathbf{y}, np)| \leq L - 1.$$

In [25] the zero-rate regime for list-decoding was derived, which is the supremum over $p \in [0, 1]$ for which $(p - \varepsilon, L)_q$ -list-decodable codes of positive rate exist for all $\varepsilon > 0$. This value was shown to be

$$p_*(q, L) = 1 - \frac{1}{L} \mathbb{E}_{(X_1, \dots, X_L) \sim U_q^{\otimes L}} [\text{pl}(X_1, \dots, X_L)], \quad (5)$$

where the function pl outputs the number of times the most popular symbol appears. In [25] it is shown that $(p_*(q, L) + \varepsilon, L)$ -list-decodable codes have size $O_{\varepsilon, q, L}(1)$, i.e., some constant independent of n . Our target in this work is to show that the correct dependence on ε is $O_{q, L}(1/\varepsilon)$, except for the case of $q = 2$ with odd L .

A “dual” definition of list-decodability is proffered by the Chebyshev radius.

► **Definition 4** (Chebyshev radius). *The Chebyshev radius of a list of distinct vectors $\mathbf{c}_1, \dots, \mathbf{c}_L \in [q]^n$ is defined as*

$$\text{rad}_H(\mathbf{c}_1, \dots, \mathbf{c}_L) := \frac{1}{n} \min_{\mathbf{r} \in [q]^n} \max_{i \in [L]} d_H(\mathbf{c}_i, \mathbf{r}).$$

Observe that a code $\mathcal{C} \subseteq [q]^n$ is (p, L) -list-decodable if and only if

$$\min\{\text{rad}_H(\mathbf{c}_1, \dots, \mathbf{c}_L) : \mathbf{c}_1, \dots, \mathbf{c}_L \in \mathcal{C} \text{ distinct}\} > p. \quad (6)$$

In particular, to show a code fails to be list-decodable, it suffices to upper bound the Chebyshev radius of L distinct codewords.

Recall that our main target is an upper bound on the size of list-decodable/-recoverable codes (in the zero-rate regime). A natural approach is to derive from Equation (6) the desired bound on the code; however, this quantity is quite difficult to work with directly. We therefore work with a *relaxed* version, which we now introduce.

We require the following definitions. Let us embed $[q]^n$ into the simplex $\Delta([q])$ via the following map φ :

$$\begin{aligned} \varphi: [q] &\rightarrow \Delta([q]) \\ x &\mapsto \mathbf{e}_x \end{aligned} \quad (7)$$

where \mathbf{e}_x is the q -dimensional vector with a 1 in the x -th location and 0 everywhere else. Denote by $\Delta = \Delta([q])$ the simplex and $\partial\Delta = \{\mathbf{e}_1, \dots, \mathbf{e}_q\}$ its vertices. For $\boldsymbol{\chi} = \mathbf{e}_x \in \partial\Delta$ and $\boldsymbol{\eta} \in \Delta$, let

$$d(\boldsymbol{\chi}, \boldsymbol{\eta}) := \frac{1}{2} \|\boldsymbol{\chi} - \boldsymbol{\eta}\|_1 = \frac{1}{2} \left(1 - \boldsymbol{\eta}(x) + \sum_{x' \in [q] \setminus \{x\}} \boldsymbol{\eta}(x') \right). \quad (8)$$

Note that if $\boldsymbol{\eta} = \mathbf{e}_y \in \partial\Delta$, then

$$d(\boldsymbol{\chi}, \boldsymbol{\eta}) = d_H(x, y).$$

From now on we will only work with Δ^n -valued vectors and will still denote such length- qn vectors by boldface letters, abusing the notation. For $\mathbf{y} \in \Delta^n$, we use $y(j, k) \in [0, 1]$ to denote its (j, k) -th element and use $\mathbf{y}(j) = (y(j, 1), \dots, y(j, q)) \in \Delta$ to denote its j -th block of size q . For $\mathbf{c} \in [q]^n$, we use $\mathbf{c}(j) \in [q]$ to denote its j -th element.

For $\mathbf{x} \in (\partial\Delta)^n$ and $\mathbf{y} \in \Delta^n$, the definition of $d(\cdot, \cdot)$ can be extended to length- qn vectors in the natural way. Specifically,

$$d(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^n d(\mathbf{x}(j), \mathbf{y}(j)). \quad (9)$$

We may now define the *relaxed Chebyshev radius*.

► **Definition 5.** *The relaxed Chebyshev radius of a list of distinct vectors $\mathbf{x}_1, \dots, \mathbf{x}_L \in (\partial\Delta)^n$ is*

$$\text{rad}(\mathbf{x}_1, \dots, \mathbf{x}_L) := \frac{1}{n} \min_{\mathbf{y} \in \Delta^n} \max_{i \in [L]} d(\mathbf{x}_i, \mathbf{y}). \quad (10)$$

Observe that

$$\text{rad}(\mathbf{x}_1, \dots, \mathbf{x}_L) \leq \text{rad}_H(\mathbf{c}_1, \dots, \mathbf{c}_L). \quad (11)$$

where $\varphi(\mathbf{c}_i) = \mathbf{x}_i$ (here we extend the definition of φ to length- n inputs in a similar way as in Equation (9)). This justifies the “relaxation” terminology.

As a last piece of terminology, we define the radius of a code.

► **Definition 6.** *For any code $\mathcal{C} \subseteq [q]^n$, we define the Chebyshev radius of \mathcal{C} as*

$$\text{rad}(\mathcal{C}) = \frac{1}{n} \min_{\mathbf{x} \in [q]^n} \max_{\mathbf{c} \in \mathcal{C}} d_H(\mathbf{c}, \mathbf{x}).$$

3 Zero-Rate List-Decoding

3.1 Linear Programming Relaxation

We have shown in Equation (11) that rad is smaller than rad_H . Conversely, Lemma 7 below establishes that rad and rad_H do not differ much. That is, for any list, if a center $\mathbf{y} \in \Delta^n$ achieves a relaxed radius t , then there must exist $\mathbf{r} \in [q]^n$ attaining approximately the same t for sufficiently large n .

► **Lemma 7** (rad is close to rad_H). *Let $\mathbf{c}_1, \dots, \mathbf{c}_L \in [q]^n$. Denote by $\mathbf{x}_1, \dots, \mathbf{x}_L \in (\partial\Delta)^n$ the images of $\mathbf{c}_1, \dots, \mathbf{c}_L$ under the embedding φ . Then*

$$\text{rad}_H(\mathbf{c}_1, \dots, \mathbf{c}_L) \leq \text{rad}(\mathbf{x}_1, \dots, \mathbf{x}_L) + \frac{L}{n}.$$

Proof. Suppose $\text{rad}(\mathbf{x}_1, \dots, \mathbf{x}_L) = t$. Then there exists $\mathbf{y} \in \Delta^n$ such that for every $i \in [L]$,

$$d(\mathbf{x}_i, \mathbf{y}) = \frac{1}{2} \sum_{j=1}^n \left(1 - y(j, \mathbf{c}_i(j)) + \sum_{x \in [q] \setminus \{\mathbf{c}_i(j)\}} y(j, x) \right) \leq t,$$

where the first equality is by Equations (8) and (9). That is, the following polytope is nonempty:

$$\left\{ \mathbf{y} \in \Delta^n : \forall i \in [L], d(\mathbf{x}_i, \mathbf{y}) \leq t \right\} = \left\{ (y(j, k))_{(j, k) \in [n] \times [q]} : \begin{array}{l} \forall (j, k) \in [n] \times [q], y(j, k) \geq 0, \\ \forall j \in [n], \sum_{k=1}^q y(j, k) = 1, \\ \forall i \in [L], \frac{1}{2} \sum_{j=1}^n \left(1 - y(j, \mathbf{c}_i(j)) + \sum_{x \in [q] \setminus \{\mathbf{c}_i(j)\}} y(j, x) \right) \leq t \end{array} \right\}.$$

Equivalently, the following linear program (LP) is feasible:

$$\begin{array}{ll} \max & 0 \\ (y(j, k))_{(j, k) \in [n] \times [q]} & \\ \text{s.t.} & \forall (j, k) \in [n] \times [q], y(j, k) \geq 0, \\ & \forall j \in [n], \sum_{k=1}^q y(j, k) = 1, \\ & \forall i \in [L], \frac{1}{2} \sum_{j=1}^n \left(1 - y(j, \mathbf{c}_i(j)) + \sum_{x \in [q] \setminus \{\mathbf{c}_i(j)\}} y(j, x) \right) \leq t. \end{array}$$

Since the equality $\langle \mathbf{a}, \mathbf{y} \rangle \leq b$ is equivalent to the equality $\langle \mathbf{a}, \mathbf{y} \rangle + z = b, z \geq 0$, the above LP can be written in equational form:

$$\begin{array}{ll} \max & 0 \\ (y(j, k))_{(j, k) \in [n] \times [q]}, (z(i))_{i \in [L]} & \\ \text{s.t.} & \forall (j, k) \in [n] \times [q], y(j, k) \geq 0, \\ & \forall i \in [L], z(i) \geq 0, \\ & \forall j \in [n], \sum_{k=1}^q y(j, k) = 1, \\ & \forall i \in [L], \frac{1}{2} \sum_{j=1}^n \left(1 - y(j, \mathbf{c}_i(j)) + \sum_{x \in [q] \setminus \{\mathbf{c}_i(j)\}} y(j, x) \right) + z(i) \leq t, \end{array} \tag{12}$$

or more compactly in matrix form:

$$\begin{array}{ll} \max & 0 \\ \mathbf{y} \in \mathbb{R}^{nq}, \mathbf{z} \in \mathbb{R}^L & \\ \text{s.t.} & \begin{bmatrix} A & I_L \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} t \mathbf{1}_L \\ \mathbf{1}_n \end{bmatrix}, \\ & \mathbf{y}, \mathbf{z} \geq \mathbf{0}. \end{array}$$

Here $A \in \mathbb{R}^{L \times (nq)}$ and $B \in \mathbb{R}^{n \times (nq)}$ encode respectively the fourth and third constraints in Equation (12), and $I_L \in \mathbb{R}^{L \times L}$, $\mathbf{1}_L \in \mathbb{R}^L$ denote respectively the $L \times L$ identity matrix and the all-one vector of length L . It is clear that $\text{rk} \left(\begin{bmatrix} A & I_L \\ B & 0 \end{bmatrix} \right) \leq n + L$. This implies that there exists a feasible solution \mathbf{y}, \mathbf{z} that has at most $n + L$ nonzeros and thus $\mathbf{y} = (y(j, k))_{(j,k) \in [n] \times [q]}$ has at most $n + L$ nonzeros. Indeed, such solutions are known as the *basic feasible solutions*. Note that for every block $j \in [n]$, $\sum_{k=1}^q y(j, k) = 1$. This implies that $y(j, 1), \dots, y(j, q)$ cannot be simultaneously 0. Moreover, if $q - 1$ out of them are 0, the remaining one is forced to be 1. Since there are n blocks in total, by the pigeonhole principle, there are at least $n - L$ choices of $j \in [n]$ such that $\mathbf{y}(j) = (y(j, 1), \dots, y(j, q)) \in \partial\Delta$. Without loss of generality, we assume that these $n - L$ indices are $1, \dots, n - L$. Let $\mathbf{r} \in [q]^n$ be such that $\varphi(\mathbf{r}(j)) = \mathbf{y}(j)$ for $j = 1, \dots, n - L$ and $r(j)$ is any value in $[q]$ for $j = n - L + 1, \dots, n$. Since $d(\mathbf{x}_i(j), \mathbf{y}(j)) \in [0, 1]$ and $d_H(\mathbf{c}_i(j), \mathbf{r}(j)) \in \{0, 1\}$, the difference between $d(\mathbf{x}_i, \mathbf{y})$ and $d_H(\mathbf{c}_i, \mathbf{r})$ is at most L . The proof is completed. \blacktriangleleft

We further relax rad by defining the *weighted average radius*. For $\mathbf{x}_1, \dots, \mathbf{x}_L \in (\partial\Delta)^n$ and $\omega \in \Delta([L])$, let

$$\overline{\text{rad}}_\omega(\mathbf{x}_1, \dots, \mathbf{x}_L) := \frac{1}{n} \min_{\mathbf{y} \in \Delta^n} \mathbb{E}_{i \sim \omega} [d(\mathbf{x}_i, \mathbf{y})] = \frac{1}{n} \min_{\mathbf{y} \in \Delta^n} \sum_{i \in [L]} \omega(i) d(\mathbf{x}_i, \mathbf{y}).$$

In words, weighted average radius is obtained by replacing the maximization over $i \in [L]$ in the definition of relaxed radius (see Equation (10)) with an average with respect to a distribution ω .

Since the objective of the minimization is separable, one can minimize over each $y(j)$ individually and obtain an alternative expression. Suppose $\mathbf{x}_1, \dots, \mathbf{x}_L \in (\partial\Delta)^n$ are the images of $\mathbf{c}_1, \dots, \mathbf{c}_L \in [q]^n$ under the embedding φ . Then

$$\begin{aligned} \overline{\text{rad}}_\omega(\mathbf{x}_1, \dots, \mathbf{x}_L) &= \frac{1}{n} \min_{\mathbf{y} \in \Delta^n} \sum_{i \in [L]} \omega(i) d(\mathbf{x}_i, \mathbf{y}) \\ &= \frac{1}{2n} \min_{(\mathbf{y}_1, \dots, \mathbf{y}_n) \in \Delta^n} \sum_{j=1}^n \left[\sum_{i \in [L]} \omega(i) \left(1 - \mathbf{y}_j(\mathbf{c}_i(j)) + \sum_{x \in [q] \setminus \{\mathbf{c}_i(j)\}} \mathbf{y}_j(x) \right) \right] \\ &= \frac{1}{2n} \sum_{j=1}^n \min_{\mathbf{y}_j \in \Delta} \left[\sum_{i \in [L]} \omega(i) \left(1 - 2\mathbf{y}_j(\mathbf{c}_i(j)) + \sum_{x \in [q]} \mathbf{y}_j(x) \right) \right] \end{aligned} \quad (13)$$

$$\begin{aligned} &= \frac{1}{2n} \sum_{j=1}^n \min_{\mathbf{y}_j \in \Delta} \left[\sum_{i \in [L]} \omega(i) (2 - 2\mathbf{y}_j(\mathbf{c}_i(j))) \right] \\ &= \frac{1}{n} \sum_{j=1}^n \min_{\mathbf{y}_j \in \Delta} \left[1 - \sum_{i \in [L]} \omega(i) \mathbf{y}_j(\mathbf{c}_i(j)) \right] = 1 - \frac{1}{n} \sum_{j=1}^n \max_{x \in [q]} \sum_{\substack{i \in [L] \\ \mathbf{c}_i(j)=x}} \omega(i). \end{aligned} \quad (14)$$

Equation (13) holds since the objective in brackets only depends on \mathbf{y}_j , not on other $(\mathbf{y}_{j'})_{j' \in [n] \setminus \{j\}}$. To see Equation (14), we note that a maximizer $\mathbf{y}^* \in \Delta$ to the following problem $\max_{\mathbf{y} \in \Delta} \sum_{i \in [L]} \omega(i) y(x_i)$, where $\omega \in \Delta([L])$ and $(x_1, \dots, x_L) \in [q]^L$ are fixed, is given by $\mathbf{y}^* = \mathbf{e}_{x^*}$ where $x^* \in [q]$ satisfies $x^* \in \arg\max_{x \in [q]} \sum_{i \in [L]} \omega(i) \mathbf{1}\{x_i = x\}$.

Obviously, by definition, for any $\mathbf{x}_1, \dots, \mathbf{x}_L \in (\partial\Delta)^n$ and $\omega \in \Delta([L])$, $\overline{\text{rad}}_\omega(\mathbf{x}_1, \dots, \mathbf{x}_L) \leq \text{rad}(\mathbf{x}_1, \dots, \mathbf{x}_L)$. In fact, the following lemma shows that rad is equal to the maximum $\overline{\text{rad}}_\omega$ over ω .

82:12 Tight Bounds on List-Decodable and List-Recoverable Zero-Rate Codes

► **Lemma 8** (rad equals maximum $\overline{\text{rad}}_\omega$). *For any $\mathbf{x}_1, \dots, \mathbf{x}_L \in (\partial\Delta)^n$,*

$$\text{rad}(\mathbf{x}_1, \dots, \mathbf{x}_L) = \max_{\omega \in \Delta([L])} \overline{\text{rad}}_\omega(\mathbf{x}_1, \dots, \mathbf{x}_L).$$

Proof. Note that

$$\text{rad}(\mathbf{x}_1, \dots, \mathbf{x}_L) := \frac{1}{n} \min_{\mathbf{y} \in \Delta^n} \max_{i \in [L]} d(\mathbf{x}_i, \mathbf{y}) = \frac{1}{n} \min_{\mathbf{y} \in \Delta^n} \max_{\omega \in \Delta([L])} \sum_{i \in [L]} \omega(i) d(\mathbf{x}_i, \mathbf{y}),$$

since the inner maximum is anyway achieved by a singleton distribution. Note also that the objective function

$$\sum_{i \in [L]} \omega(i) d(\mathbf{x}_i, \mathbf{y}) = \frac{1}{2} \sum_{i \in [L]} \omega(i) \sum_{j=1}^n \left(1 - y(j, \mathbf{c}_i(j)) + \sum_{x \in [q] \setminus \{\mathbf{c}_i(j)\}} y(j, x) \right)$$

is affine in ω and linear in \mathbf{y} . Therefore, von Neumann's minimax theorem allows us to interchange min and max and obtain

$$\text{rad}(\mathbf{x}_1, \dots, \mathbf{x}_L) = \frac{1}{n} \max_{\omega \in \Delta([L])} \min_{\mathbf{y} \in \Delta^n} \sum_{i \in [L]} \omega(i) d(\mathbf{x}_i, \mathbf{y}) = \max_{\omega \in \Delta([L])} \overline{\text{rad}}_\omega(\mathbf{x}_1, \dots, \mathbf{x}_L),$$

as claimed by the lemma. ◀

In fact, we can say something stronger: it is not necessary to maximize over the entire (uncountable) probability simplex $\Delta([L])$. Instead, we can extract a finite subset $\Omega_L \subset \Delta([L])$ and maximize over this set to recover rad. The following lemma is analogous to [1, Lemma 6].

► **Lemma 9** (rad is achieved by finitely many ω). *For every L , there exists a finite set of probability measures $\Omega_L \subseteq \Delta([L])$ such that*

$$\text{rad}(\mathbf{x}_1, \dots, \mathbf{x}_L) = \max_{\omega \in \Omega_L} \overline{\text{rad}}_\omega(\mathbf{x}_1, \dots, \mathbf{x}_L).$$

for all $\mathbf{x}_1, \dots, \mathbf{x}_L \in \partial\Delta^n$.

Proof. The idea is to view the computation of $\max_{\omega \in \Omega_L} \overline{\text{rad}}_\omega(\mathbf{x}_1, \dots, \mathbf{x}_L)$ as finding the maximum among some finite set of linear program maxima over some convex polytopes, and then to take Ω_L to be the set of vertices of the defined convex polytopes.

First, we define the convex polytopes based on a (q -ary version of a) signature. For each $\omega \in \Delta([L])$, we define a *signature* for ω which is a function $S_\omega : [q]^L \rightarrow [q]$ such that

$$S_\omega(\mathbf{u}) \in \operatorname{argmax}_{x \in [q]} \sum_{i: \mathbf{u}(i)=x} \omega(i)$$

for $\mathbf{u} \in [q]^L$. Define further the q halfspaces $H_{\mathbf{u},x} := \{\omega \in \Delta([L]) : \sum_{i: \mathbf{u}(i)=x} \omega(i) \geq 1/q\}$ for $x \in [q]$. Observe that if $S(\mathbf{u}) = x$ where S is a signature for ω then $\omega \in H_{\mathbf{u},x}$. Thus, by ranging over the choices for $\mathbf{u} \in [q]^L$ and $x \in [q]$ we obtain q^{L+1} halfspaces that partition the $(L-1)$ -dimensional space $\Delta([L])$ into at most $\sum_{j \leq L-1} \binom{q^{L+1}}{j}$ regions.

For each possible signature $S : [q]^L \rightarrow [q]$, let $\Omega_S = \{\omega \in \Delta([L]) : S \text{ is a signature for } \omega\}$, and note that Ω_S is a convex polytope. Indeed, it is an intersection over $\mathbf{u} \in [q]^L$ of the convex polytopes

$$\begin{aligned} & \{\omega \in \Delta([L]) : \exists \text{ signature } S_\omega \text{ for } \omega \text{ s.t. } S_\omega(\mathbf{u}) = S(\mathbf{u})\} \\ &= \bigcap_{\mathbf{y} \in [q] \setminus S(\mathbf{u})} \left\{ \omega \in \Delta([L]) : \sum_{i: \mathbf{u}(i)=S(\mathbf{u})} \omega(i) \geq \sum_{i: \mathbf{u}(i)=\mathbf{y}} \omega(i) \right\} \end{aligned}$$

where S_ω is a signature for ω . Now, to maximize

$$\overline{\text{rad}}_\omega(\mathbf{x}_1, \dots, \mathbf{x}_L) = \frac{1}{n} \min_{\mathbf{y} \in \Delta^n} \sum_{i \in [L]} \omega(i) d(\mathbf{x}_i, \mathbf{y})$$

over $\omega \in \Omega_L$, consider the set $T_{\mathbf{u}} = \{i \in [n] : (\mathbf{x}_1(i), \dots, \mathbf{x}_L(i)) = \mathbf{u}\}$ for $\mathbf{u} \in [q]^L$ and let $a_{\mathbf{u}} = \frac{|T_{\mathbf{u}}|}{n}$. We claim it suffices to find the maximum of the following linear function:

$$\sum_{\mathbf{u} \in [q]^L} a_{\mathbf{u}} y_{\mathbf{u}}, \quad \text{s.t.} \quad y_{\mathbf{u}} = \sum_{i: \mathbf{u}(i)=S(\mathbf{u})} \omega(i) \quad (15)$$

over all $\omega \in \Omega_S$. Indeed, by Equation (14), we have

$$\overline{\text{rad}}_\omega(\mathbf{x}_1, \dots, \mathbf{x}_L) = 1 - \frac{1}{n} \sum_{j=1}^n \max_{x \in [q]} \sum_{\substack{i \in [L] \\ c_i(j)=x}} \omega(i).$$

This implies that a maximizer only depends on the index set $T_{\mathbf{u}}$, and furthermore that its value is determined by the $a_{\mathbf{u}}$'s as in Equation (15).

We can thus take the union of all vertex sets of all polytopes Ω_S for all signatures S . Multiplying this by the $O_{q,L}(1)$ regions defined by all the halfspaces $H_{\mathbf{u},x}$ we obtain a finite set of vertices, as desired. \blacktriangleleft

3.2 Properties of $f(P, \omega)$

Now, we consider the expected weighted average radius of a sequence of i.i.d. symbols. Specifically, for $P \in \Delta([q])$ and $\omega \in \Delta([L])$, let

$$f(P, \omega) := \mathbb{E}_{(X_1, \dots, X_L) \sim P^{\otimes L}} \left[\overline{\text{rad}}_\omega(\mathbf{e}_{X_1}, \dots, \mathbf{e}_{X_L}) \right].$$

[1] studies $f(P, \omega)$ for $q = 2$ and even L . In this case, one can take advantage of the fact that $P \in \Delta([2])$ may be parametrized by a single real number, and thereby yield a fairly simple expression for $f(P, \omega)$.

Nonetheless, in this subsection, we will show that all properties of $f(P, \omega)$ in [1] holding for $q = 2$ and even L can be generalized to any $q \geq 3$ and any L . Let us first provide a more explicit expression for $f(P, \omega)$ using Equation (14):

$$\begin{aligned} f(P, \omega) &:= \mathbb{E}_{(X_1, \dots, X_L) \sim P^{\otimes L}} \left[1 - \max_{x \in [q]} \sum_{\substack{i \in [L] \\ X_i = x}} \omega(i) \right] \\ &= 1 - \sum_{(x_1, \dots, x_L) \in [q]^L} \left(\prod_{i=1}^L P(x_i) \right) \max_{x \in [q]} \sum_{i \in [L]} \omega(i) \mathbb{1}\{x_i = x\}. \end{aligned}$$

We define the shorthand notation

$$\max_\omega(x_1, \dots, x_L) := \max_{\substack{x \in [q] \\ x_i = x}} \sum_{i \in [L]} \omega(i) \quad (16)$$

for any $\omega \in \Delta([L])$ and $(x_1, \dots, x_L) \in [q]^L$.

82:14 Tight Bounds on List-Decodable and List-Recoverable Zero-Rate Codes

The first property that we would like to establish is that $f(P, \omega)$ only increases if ω is replaced by U_L , and furthermore that the maximum is uniquely obtained at U_L if $P(x) > 0$ for all $x \in [q]$. In order to do this, we will regularly “average-out” coordinates of ω and then show that the function value increases (or at least, does not decrease). To introduce some terminology, for $S \subseteq [L]$ we say that $\bar{\omega}$ is obtained from ω by *averaging-out* the subset S of coordinates if $\bar{\omega}$ is defined as

$$\bar{\omega}(i) = \begin{cases} \frac{\sum_{j \in S} \omega(j)}{|S|} & i \in S \\ \omega(i) & i \notin S \end{cases}.$$

The following lemma gives a simple criterion for establishing that, if $\bar{\omega}$ is obtained from ω by averaging two coordinates, then $f(P, \bar{\omega}) \leq f(P, \omega)$, and it furthermore gives a criterion for the inequality to be strict. The main thrust of the proof of Lemma 11 is thus to show that this criterion is always satisfied.

► **Lemma 10.** *Let $P \in \Delta([q])$ and $\omega \in \Delta([L])$. Suppose $\omega(L-1) \neq \omega(L)$ and that $\bar{\omega} \in \Delta([L])$ is obtained by averaging-out the last two coordinates of ω . Suppose that for all $(x_1, \dots, x_L) \in [q]^L$ we have*

$$\frac{1}{2} (\max_{\omega}(x_1, \dots, x_{L-1}, x_L) + \max_{\omega}(x_1, \dots, x_L, x_{L-1})) \geq \max_{\bar{\omega}}(x_1, \dots, x_{L-1}, x_L). \quad (17)$$

Then $f(P, \bar{\omega}) \geq f(P, \omega)$.

Furthermore, suppose that additionally there exists $(x_1, \dots, x_L) \in [q]^L$ with $\prod_{i=1}^L P(x_i) > 0$ such that the inequality in Equation (17) is strict. Then $f(P, \bar{\omega}) > f(P, \omega)$.

Proof. Define $\omega' \in \Delta([L])$ as

$$\omega'(i) = \begin{cases} \omega(i), & i \in [L] \setminus \{L-1, L\} \\ \omega(L), & i = L-1 \\ \omega(L-1), & i = L \end{cases}.$$

That is, ω' is obtained by swapping the last two components of ω . By symmetry, we have $f(P, \omega) = f(P, \omega')$ and so

$$\begin{aligned} f(P, \omega) &= \frac{1}{2} (f(P, \omega) + f(P, \omega')) \\ &= \frac{1}{2} \left(1 - \sum_{(x_1, \dots, x_L) \in [q]^L} \left(\prod_{i=1}^L P(x_i) \right) \max_{\omega}(x_1, \dots, x_{L-1}, x_L) \right. \\ &\quad \left. + 1 - \sum_{(x_1, \dots, x_L) \in [q]^L} \left(\prod_{i=1}^L P(x_i) \right) \max_{\omega'}(x_1, \dots, x_{L-1}, x_L) \right) \\ &= 1 - \sum_{(x_1, \dots, x_L) \in [q]^L} \left(\prod_{i=1}^L P(x_i) \right) \frac{1}{2} (\max_{\omega}(x_1, \dots, x_{L-1}, x_L) + \max_{\omega}(x_1, \dots, x_L, x_{L-1})) \\ &\leq 1 - \sum_{(x_1, \dots, x_L) \in [q]^L} \left(\prod_{i=1}^L P(x_i) \right) \max_{\bar{\omega}}(x_1, \dots, x_{L-1}, x_L) \\ &= f(P, \bar{\omega}), \end{aligned}$$

where the inequality follows from Equation (17). From the above sequence of inequalities, it is also clear that if additionally there exists $(x_1, \dots, x_L) \in [q]^L$ with $\prod_{i=1}^L P(x_i) > 0$ for which the inequality in Equation (17) is strict, then $f(P, \bar{\omega}) > f(P, \omega)$. ◀

We now establish that the function value cannot decrease if ω is replaced by U_L .

► **Lemma 11.** For any $P \in \Delta([q])$ and $\omega \in \Delta([L])$, $f(P, \omega) \leq f(P, U_L)$.

Proof. Fix any $(x_1, \dots, x_L) \in [q]^L$. Let $\omega \in \Delta([L])$ be non-uniform. Without loss of generality, assume $\omega(L-1) \neq \omega(L)$. Let $\bar{\omega} \in \Delta([L])$ be obtained by uniformizing the last two components of ω , i.e.,

$$\bar{\omega}(i) = \begin{cases} \omega(i), & i \in [L] \setminus \{L-1, L\} \\ \frac{1}{2}(\omega(L-1) + \omega(L)), & i \in \{L-1, L\} \end{cases}.$$

We claim $f(P, \bar{\omega}) \geq f(P, \omega)$. By Lemma 10, we just need to establish Equation (17).

Equation (17) trivially holds if $x_{L-1} = x_L$. We therefore assume below $x_{L-1} \neq x_L$. Let $x_{L-1} = a$ and $x_L = b$. Let

$$\omega^{(a)} = \sum_{\substack{i \in [L-2] \\ x_i = a}} \omega(i), \quad \omega^{(b)} = \sum_{\substack{i \in [L-2] \\ x_i = b}} \omega(i).$$

Then, we have

$$\sum_{\substack{i \in [L] \\ x_i = a}} \bar{\omega}(i) = \omega^{(a)} + \frac{1}{2}(\omega(L-1) + \omega(L)), \quad \sum_{\substack{i \in [L] \\ x_i = b}} \bar{\omega}(i) = \omega^{(b)} + \frac{1}{2}(\omega(L-1) + \omega(L)).$$

We first assume that there exists $c \notin \{a, b\}$ such that

$$\max_{\bar{\omega}}(x_1, \dots, x_L) = \sum_{\substack{i \in [L] \\ x_i = c}} \bar{\omega}(i) = \sum_{\substack{i \in [L] \\ x_i = c}} \omega(i)$$

where the second equality follows since the set $\{i \in [q] : x_i = c\}$ does not contain $L-1, L$. Equation (17) therefore holds as

$$\max_{\omega}(x_1, \dots, x_{L-1}, x_L) \geq \sum_{\substack{i \in [L] \\ x_i = c}} \omega(i), \quad \max_{\omega}(x_1, \dots, x_L, x_{L-1}) \geq \sum_{\substack{i \in [L] \\ x_i = c}} \omega(i).$$

We proceed to the case that

$$\max_{\bar{\omega}}(x_1, \dots, x_L) = \max \left\{ \frac{1}{2}(\omega(L-1) + \omega(L)) + \omega^{(a)}, \frac{1}{2}(\omega(L-1) + \omega(L)) + \omega^{(b)} \right\}.$$

Equation (17) holds as

$$\max_{\omega}(x_1, \dots, x_{L-1}, x_L) \geq \max \left\{ \omega^{(a)} + \omega(L-1), \omega^{(b)} + \omega(L) \right\}$$

and

$$\max_{\omega}(x_1, \dots, x_L, x_{L-1}) \geq \max \left\{ \omega^{(a)} + \omega(L), \omega^{(b)} + \omega(L-1) \right\}.$$

Thus, Lemma 10 implies $f(P, \bar{\omega}) \geq f(P, \omega)$, as desired.

We can then continue averaging components of ω and in this way obtain a sequence $(\omega_i)_{i \in \mathbb{N}}$ of distributions with $\omega_1 = \omega$. This sequence converges in ℓ_∞ -norm to the uniform distribution U_L and satisfies $f(P, \omega_{i+1}) \geq f(P, \omega_i)$ for all $i \in \mathbb{N}$. Observing that $\omega \mapsto f(P, \omega)$ is a continuous function – the term $\max_{x \in [q]} \sum_{i \in [L]} \omega(i) \mathbb{1}\{x_i = x\}$ is a maximum over linear functions of ω , hence linear, implying that $f(P, \cdot)$ is a linear combination of continuous functions – it follows that $f(P, U_L) = \lim_{i \rightarrow \infty} f(P, \omega_i)$, and in particular that $f(P, U_L) \geq f(P, \omega_1) = f(P, \omega)$, as desired. ◀

82:16 Tight Bounds on List-Decodable and List-Recoverable Zero-Rate Codes

We now strengthen the conclusion of Lemma 11 by showing that for all $q \geq 3$ and $L \geq 2$ the function $\omega \mapsto f(P, \omega)$ is *uniquely* maximized by the setting $\omega = U_L$, except for degenerate cases where $P(x) = 0$ for some $x \in [q]$.

Before stating and proving this fact, we note that the proof of Lemma 11 in fact shows that we can average out any subset of coordinates of ω and only increase the value of $f(P, \omega)$. We formalize this fact in the following lemma, which will be useful in the following arguments.

► **Lemma 12.** *Let $P \in \Delta([q])$, $\omega \in \Delta([L])$ and $S \subseteq [L]$. Let $\bar{\omega}$ be obtained from ω by averaging-out the subset S of coordinates. Then $f(P, \bar{\omega}) \geq f(P, \omega)$.*

► **Theorem 13.** *Let $q \geq 3$, $L \geq 2$ and let $P \in \Delta([q])$ be such that $P(x) > 0$ ⁶ for all $x \in [q]$. Then for all $\omega \in \Delta([L])$, $f(P, \omega) \leq f(P, U_L)$ with equality if and only if $\omega = U_L$.*

Proof. The inequality was already established in Lemma 11, so we focus on showing $\omega = U_L$ when $f(P, \omega) = f(P, U_L)$. As $q \geq 3$, let a, b and c denote 3 distinct elements of $[q]$. Let $\omega \neq U_L$ and suppose for a contradiction that $f(P, \omega)$ is a maximum of the function $\omega \mapsto f(P, \omega)$. The proof proceeds via a number of cases.

1. **L is even.** Without loss of generality, $\omega(L-1) < \omega(L)$. If $L \geq 4$, let ω' be obtained from ω by averaging-out the first $L-2$ coordinates; by Lemma 12, $f(P, \omega') \geq f(P, \omega)$. If $L = 2$, set $\omega' = \omega$.

If $L \geq 4$, since $2|(L-2)$, we can set $x_1 = \dots = x_{L/2-1} = a$ and $x_{L/2} = \dots = x_{L-2} = b$. Set further $x_{L-1} = a$ and $x_L = b$. We observe that for this $(x_1, \dots, x_L) \in [q]^L$ and $\bar{\omega}$ obtained from ω' by averaging-out the last two coordinates, Equation (17) strictly holds. Indeed,

$$\max_{\omega'}(x_1, \dots, x_{L-1}, x_L) + \max_{\omega'}(x_1, \dots, x_L, x_{L-1}) = 2 \sum_{j=1}^{(L-2)/2} \omega(i) + 2\omega(L)$$

and

$$2\max_{\bar{\omega}}(x_1, \dots, x_{L-1}, x_L) = 2 \sum_{j=1}^{(L-2)/2} \omega(i) + \omega(L-1) + \omega(L).$$

Thus Lemma 10 implies $f(P, \bar{\omega}) > f(P, \omega') \geq f(P, \omega)$, a contradiction.

2. **L is odd and at least three components of ω take distinct values, or the components in ω only take two different values and at least two of them take the minimum value.** Without loss of generality $\omega(L-2) \leq \omega(L-1) < \omega(L)$. If $L \geq 5$, let ω' be obtained from ω by averaging-out the first $L-3$ coordinates; by Lemma 12, $f(P, \omega') \geq f(P, \omega)$. If $L = 3$, set $\omega' = \omega$.

Since $2|(L-3)$, if $L \geq 5$, we set $x_1 = \dots = x_{(L-1)/2-1} = a$ and $x_{(L-1)/2} = \dots = x_{L-3} = b$. Let $x_{L-2} = c$, $x_{L-1} = a$ and $x_L = b$. We observe that for this $(x_1, \dots, x_L) \in [q]^L$ and $\bar{\omega}$ obtained from ω' by averaging-out the last two coordinates, Equation (17) strictly holds. Indeed, since $\omega(L-2) < \omega(L)$ we have

$$\max_{\omega'}(x_1, \dots, x_{L-1}, x_L) + \max_{\omega'}(x_1, \dots, x_L, x_{L-1}) = 2 \sum_{j=1}^{(L-2)/2} \omega(i) + 2\omega(L)$$

⁶ In fact, our proof only apply with $P = U_q$ which clearly satisfies the condition.

and

$$2\max_{\bar{\omega}}(x_1, \dots, x_{L-1}, x_L) = 2 \sum_{j=1}^{(L-3)/2} \omega(j) + \omega(L-1) + \omega(L).$$

Thus Lemma 10 implies $f(P, \bar{\omega}) > f(P, \omega') \geq f(P, \omega)$, a contradiction.

- 3. L is odd and only one component takes the minimum value.** That is, $\omega(1) = \omega(2) = \dots = \omega(L-1) < \omega(L)$. Let ω' be obtained from ω by averaging-out the subset $\{L-1, L\}$. Then $f(P, \omega') \geq f(P, \omega)$ by Lemma 12 and moreover ω' is such that at least two coordinates take on the minimum value, as $\omega'(1) = \dots = \omega'(L-2) > \omega'(L-1) = \omega'(L)$. The argument from the previous case can now be applied to derive a contradiction. ◀

Thus, except for degenerate choices for $P \in \Delta([q])$, it follows that the function $\omega \mapsto f(P, \omega)$ is maximized by the choice of $\omega = U_L$. The next step is to determine the distribution $P \in \Delta([q])$ maximizing $P \mapsto f(P, U_L)$. At this point, we can rely on a main result of [25], as the function $P \mapsto 1 - f(P, U_L)$ is the same as the function $f_{q,L}(P)$ defined in [25, Equation (17)]. It is shown therein that $f_{q,L}(P)$ is *strictly Schur convex*, which in particular means that $f_{q,L}(P)$ has a unique minimum at $P = U_q$. That is, $f(P, U_L)$ has a unique maximum at $P = U_q$.

The (strict) Schur convexity also implies the following: if $p = \max_{x \in [q]} P(x)$, then $f_{q,L}(P) \geq f_{q,L}(P_{q,p})$ where

$$P_{q,p}(x) = \begin{cases} \frac{1-p}{q-1} & x \in \{1, 2, \dots, q-1\} \\ p & x = q \end{cases}. \quad (18)$$

That is, we can conclude that $f(P, U_L) \leq f(P_{q,p}, U_L)$. We encapsulate these facts in the following proposition.

- **Proposition 14** (Theorem 1,2 [25]). *Let $q \geq 2$, $L \geq q$ and $P \in \Delta([q])$. Suppose $p = \max_{x \in [q]} P(x)$. Then $f(P, U_L) \leq f(P_{q,p}, U_L)$. Furthermore, $f(P_{q,p}, U_L) \leq f(U_q, U_L)$ is monotone decreasing for $p \geq 1/q$. Lastly, $f(P_{q,p}, U_L)$ is concave for $p \in [1/q, 1]$, i.e., $\frac{1}{n} \sum_{i=1}^n f(P_{q,p_i}, U_L) \leq f(P_{q,p}, U_L)$ with $p = \frac{1}{n} \sum_{i=1}^n p_i$.*

A further fact that we have from [25] is that

$$p_*(q, L) = f(U_q, U_L).$$

In fact, this was taken as the definition of $p_*(q, L)$. To end this subsection, we prove the following theorem by utilizing the concavity of $f(P_{q,p}, U_L)$.

- **Theorem 15.** *Assume $\text{rad}_H(\mathcal{C}) \leq p$, then we have*

$$\mathbb{E}_{(\mathbf{c}_1, \dots, \mathbf{c}_L) \in \mathcal{C}^L} [\text{rad}_\omega(\varphi(\mathbf{c}_1), \dots, \varphi(\mathbf{c}_L))] \leq f(P_{q,p}, U_L).$$

Proof. Let \mathbf{y} be the center attaining $\text{rad}_H(\mathcal{C})$. Without loss of generality, we can assume \mathbf{y} is a all zero vector. Let P_i be the distribution of symbols in the i -th index of \mathcal{C} , i.e., $P_i(j) = \Pr[\mathbf{c}(i) = j]$ with the distribution taken over $\mathbf{c} \in \mathcal{C}$. Let $p_i = \max_{x \in [q]} P_i(x)$ and $p' = \frac{1}{n} \sum_{i=1}^n p_i$. Clearly, $p_i \geq 1/q$. Then, we have

$$\begin{aligned} \mathbb{E}_{(\mathbf{c}_1, \dots, \mathbf{c}_L) \in \mathcal{C}^L} [\text{rad}_\omega(\varphi(\mathbf{c}_1), \dots, \varphi(\mathbf{c}_L))] &= \frac{1}{n} \sum_{i=1}^n f(P_i, \omega) \\ &\leq \frac{1}{n} \sum_{i=1}^n f(P_i, U_L) \leq \frac{1}{n} \sum_{i=1}^n f(P_{q,p_i}, U_L) \leq f(P_{q,p'}, U_L) \leq f(P_{q,p}, U_L). \end{aligned}$$

The first inequality is due to Lemma 11 and the second and third inequalities are due to Proposition 14. The last inequality is due to $\text{rad}_H(\mathcal{C}) \leq p$ and the center \mathbf{y} is all zero vector. The proof is completed. \blacktriangleleft

3.3 Abundance of Random-Like L -tuples

Recall the notations

$$\text{rad}(\mathcal{C}) = \frac{1}{n} \min_{\mathbf{y} \in [q]^n} \max_{\mathbf{c} \in \mathcal{C}} d_H(\mathbf{c}, \mathbf{y})$$

and

$$\text{type}_{\mathbf{u}}(\mathbf{c}_1, \dots, \mathbf{c}_L) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{(\mathbf{c}_1(i), \dots, \mathbf{c}_L(i)) = \mathbf{u}\}$$

where $\mathbf{c}_i = (\mathbf{c}_i(1), \dots, \mathbf{c}_i(n)) \in [q]^n$ and $\mathbf{u} \in [q]^L$. In this subsection, we prove a code $\mathcal{C} \subseteq [q]^n$ either contains a large subcode $\mathcal{C}' \subseteq \mathcal{C}$ with radius $\text{rad}(\mathcal{C}') \leq 1 - \frac{1}{q} - \varepsilon$, or most of its L -tuples are of type close to the uniform distribution (for all $\mathbf{u} \in [q]^L$).

We first show that for any projection π_A with $|A| \geq \mu n$ (for some parameter $\mu \in [0, 1]$), the projection $\pi_A(\mathcal{C})$ almost preserves the radius $\text{rad}(\mathcal{C})$ with small loss. Then, if $\text{rad}(\mathcal{C}') > 1 - \frac{1}{q} - \varepsilon$ for any subcode \mathcal{C}' with large size, we find a codeword \mathbf{c}_1 in \mathcal{C} whose symbols' distribution is close to the uniform. In fact, most codewords in \mathcal{C} satisfies this requirement. Let A_i be the index set of \mathbf{c}_1 taking value i . We apply π_{A_i} to \mathcal{C} and claim that $\pi_{A_i}(\mathcal{C})$ preserves the radius. Thus, we can find a codeword \mathbf{c}_2 such that for every $i \in [q]$, the symbol's distribution of $\pi_{A_i}(\mathbf{c}_2)$ is close to uniform. Moreover, most of codewords in \mathcal{C} satisfy this requirement. The proof is the completed by induction.

► **Lemma 16.** *Let $\pi_A : [q]^n \rightarrow [q]^A$ be the projection on a set A of size m . Suppose $\mathcal{C} \subseteq [q]^n$ is a code of size qs satisfying $\text{rad}_H(\pi_A(\mathcal{C})) \leq 1 - \frac{1}{q} - \varepsilon$. Then, there exists a subcode $\mathcal{C}' \subseteq \mathcal{C}$ of size at least s such that $\text{rad}_H(\mathcal{C}') \leq 1 - \frac{1}{q} - \frac{m}{n}\varepsilon$.*

Proof. The proof is omitted due to the limited space. \blacktriangleleft

► **Theorem 17.** *Let L be fixed. For every $\varepsilon > 0$, there exists a $\delta > 0$ with the following property. If s is a natural number, there exist constants $M_0 = M_0(s)$ and $c(s)$ such that for any code $\mathcal{C} \subseteq [q]^n$ with size $M \geq M_0$, at least one of the following must hold:*

1. *There exists $\mathcal{C}' \subseteq \mathcal{C}$ such that $|\mathcal{C}'| \geq s$ and $\text{rad}_H(\mathcal{C}') \leq 1 - \frac{1}{q} - \delta$.*
2. *There exist at least $M^L - c(s)M^{L-1}$ many L -tuples of distinct codewords $(\mathbf{c}_1, \dots, \mathbf{c}_L)$ in \mathcal{C} such that for all $\mathbf{u} \in [q]^L$ we have*

$$|\text{type}_{\mathbf{u}}(\mathbf{c}_1, \dots, \mathbf{c}_L) - q^{-L}| \leq \varepsilon$$

and thus

$$|\overline{\text{rad}}_{\omega}(\varphi(\mathbf{c}_1), \dots, \varphi(\mathbf{c}_L)) - f(U_q, \omega)| \leq q^L \varepsilon.$$

Proof. The proof is omitted due to the limited space. \blacktriangleleft

3.4 Putting Everything Together

The argument follows the same line of reasoning as [1]. We provide the proof for completeness. Define $\rho_L(\mathcal{C}) = \min \text{rad}(\varphi(\mathbf{c}_1), \dots, \varphi(\mathbf{c}_L))$ with minimum taken over all L -tuples $(\mathbf{c}_1, \dots, \mathbf{c}_L) \in \mathcal{C}^L$ with distinct elements, where we recall that rad denotes the *relaxed* Chebyshev radius (Definition 5).

► **Theorem 18.** *Let $L \geq 2$ and $q \geq 3$. If $\mathcal{C} \subseteq [q]^n$ is $(p_*(q, L) + \varepsilon, L)$ -list-decodable, then $|\mathcal{C}| = O_{q,L}(\frac{1}{\varepsilon})$.*

Proof. The proof is omitted due to the limited space. ◀

4 Zero-Rate List-Recovery

In this section, we show how our results on list-decoding can naturally be extended to list-recovery. Due to the page limit, we refer the reader to the full version. The result is summarized in Theorem 1.

5 Code Construction

In this section, we present a simple simplex-like code construction and show that it attains the optimal size-radius trade-off by analyzing its list-decoding and -recovery radius.

Our construction will be identical for list-decoding and -recovery and therefore we will directly analyze its list-recovery radius. Before presenting the construction and its analysis, let us define the average radius $\overline{\text{rad}}_\ell$. This is a standard notion that “linearizes” the Chebyshev radius rad and often finds its use in the analysis of list-recoverable codes in the literature. The definition reads as follows: for any $\mathbf{c}_1, \dots, \mathbf{c}_L \in [q]^n$,

$$\overline{\text{rad}}_\ell(\mathbf{c}_1, \dots, \mathbf{c}_L) := \frac{1}{L} \min_{\mathbf{Y} \in \mathcal{X}^n} \sum_{i=1}^L d_{\text{LR}}(\mathbf{c}_i, \mathbf{Y}).$$

It is well-known and easy to verify (by, e.g., following the derivations leading to Equation (14)) that the above minimization admits the following explicit solution:

$$\overline{\text{rad}}_\ell(\mathbf{c}_1, \dots, \mathbf{c}_L) := \sum_{j=1}^n \left(1 - \frac{1}{L} \text{pl}_\ell(\mathbf{c}_1(j), \dots, \mathbf{c}_L(j)) \right), \quad (19)$$

Equation (19) should be interpreted as the average distance from each \mathbf{c}_i to the “centroid” $\mathbf{Y}^* \in \mathcal{X}^n$ of the list defined as⁷

$$\mathbf{Y}^*(j) := \operatorname{argmax}_{A \in \mathcal{X}} \sum_{i=1}^L \mathbb{1}\{\mathbf{c}_i(j) \in A\}.$$

for each $j \in [n]$.

Finally, for integers $q \geq 1$ and $L \geq 0$, denote by

$$\mathcal{A}_{q,L} = \left\{ (a_1, \dots, a_q) \in \mathbb{Z}_{\geq 0}^q : \sum_{i=1}^q a_i = L \right\}$$

⁷ If there are multiple maximizers, take an arbitrary one and the value of $\overline{\text{rad}}_\ell$ remains the same.

the set of q -partitions of L , i.e., a_i is the number of indices taking value i . For $\mathbf{a} \in \mathcal{A}_{q,L}$, we shorthand $\binom{L}{\mathbf{a}} = \binom{L}{a_1, \dots, a_q}$ where $\mathbf{a} = (a_1, \dots, a_q)$. Define $\max_{\ell}\{\mathbf{a}\} = \max_{A \in \mathcal{X}} \sum_{i \in A} a_i$, i.e., the sum of ℓ largest components in \mathbf{a} .

► **Theorem 19** (Construction of zero-rate list-recoverable codes). *Fix any integers $q \geq 3$, $\ell \geq 1$ and $L \geq 2$. For any sufficiently large m , there exists a (p, ℓ, L) -list-recoverable code \mathcal{C} with blocklength*

$$n = \binom{qm}{\underbrace{m, \dots, m}_q}, \quad (20)$$

and the trade-off between code size M and (relative) radius p given by:

$$M = qm, \quad p = p_*(q, \ell, L) + c_{q, \ell, L} m^{-1} + O(m^{-2}),$$

where

$$c_{q, \ell, L} := q^{-L} \sum_{\mathbf{a} \in \mathcal{A}_{q, L}} \frac{\max_{\ell}\{\mathbf{a}\}}{L} \binom{L}{\mathbf{a}} \left[\sum_{i=1}^q \binom{a_i}{2} - \frac{1}{q} \binom{L}{2} \right] > 0. \quad (21)$$

Proof. The proof is omitted due to the page limit. ◀

References

- 1 Noga Alon, Boris Bukh, and Yury Polyanskiy. List-decodable zero-rate codes. *IEEE Transactions on Information Theory*, 65(3):1657–1667, 2018. doi:10.1109/TIT.2018.2868957.
- 2 László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Comput. Complex.*, 3:307–318, 1993. doi:10.1007/BF01275486.
- 3 L. A. Bassalygo. New upper bounds for error-correcting codes. *Probl. of Info. Transm.*, 1:32–35, 1965.
- 4 Vladimir M Blinovskiy. Bounds for codes in the case of list decoding of finite volume. *Problems of Information Transmission*, 22:7–19, 1986.
- 5 Vladimir M Blinovskiy. Code bounds for multiple packings over a nonbinary finite alphabet. *Problems of Information Transmission*, 41:23–32, 2005. doi:10.1007/S11122-005-0007-5.
- 6 Vladimir M Blinovskiy. On the convexity of one coding-theory function. *Problems of Information Transmission*, 44:34–39, 2008. doi:10.1134/S0032946008010031.
- 7 Philippe Delsarte. An algebraic approach to the association schemes of coding theory. *Philips Res. Rep. Suppl.*, 10:vi+–97, 1973.
- 8 Dean Doron, Dana Moshkovitz, Justin Oh, and David Zuckerman. Nearly optimal pseudorandomness from hardness. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1057–1068. IEEE, 2020. doi:10.1109/FOCS46700.2020.00102.
- 9 Dean Doron and Mary Wootters. High-probability list-recovery, and applications to heavy hitters. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ICALP.2022.55.
- 10 Peter Elias. List decoding for noisy channels. *Wescon Convention Record, Part 2*, pages 94–104, 1957.
- 11 Peter Elias. Error-correcting codes for list decoding. *IEEE Transactions on Information Theory*, 37(1):5–12, 1991. doi:10.1109/18.61123.
- 12 Edgar N Gilbert. A comparison of signalling alphabets. *The Bell System Technical Journal*, 31(3):504–522, 1952.

- 13 Oded Goldreich and Leonid A Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC)*, pages 25–32. ACM, 1989. doi:10.1145/73007.73010.
- 14 Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from parvaresh–vardy codes. *Journal of the ACM (JACM)*, 56(4):1–34, 2009. doi:10.1145/1538902.1538904.
- 15 Iftach Haitner, Yuval Ishai, Eran Omri, and Ronen Shaltiel. Parallel hashing via list recoverability. In *Annual Cryptology Conference*, pages 173–190. Springer, 2015. doi:10.1007/978-3-662-48000-7_9.
- 16 Justin Holmgren, Alex Lombardi, and Ron D Rothblum. Fiat–shamir via list-recoverable codes (or: parallel repetition of gmw is not zero-knowledge). In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 750–760, 2021. doi:10.1145/3406325.3451116.
- 17 Piotr Indyk, Hung Q Ngo, and Atri Rudra. Efficiently decodable non-adaptive group testing. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1126–1142. SIAM, 2010. doi:10.1137/1.9781611973075.91.
- 18 Jeffrey C Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55(3):414–440, 1997. doi:10.1006/JCSS.1997.1533.
- 19 Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the Fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993. doi:10.1137/0222080.
- 20 VI Levenshtein. Application of hadamard matrices on coding problem. *Problems of Cybernetica*, 5:123–136, 1961.
- 21 Richard J Lipton. Efficient checking of computations. In *Proceedings of the 7th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 207–215. Springer, 1990. doi:10.1007/3-540-52282-4_44.
- 22 Robert J. McEliece, Eugene R. Rodemich, Howard Rumsey, Jr., and Lloyd R. Welch. New upper bounds on the rate of a code via the Delsarte–MacWilliams inequalities. *IEEE Trans. Inform. Theory*, IT-23(2):157–166, 1977. doi:10.1109/tit.1977.1055688.
- 23 Hung Q Ngo, Ely Porat, and Atri Rudra. Efficiently decodable error-correcting list disjoint matrices and applications. In *International Colloquium on Automata, Languages, and Programming*, pages 557–568. Springer, 2011.
- 24 Nicolas Resch. List-decodable codes:(randomized) constructions and applications. *School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep., CMU-CS-20-113*, 2020.
- 25 Nicolas Resch, Chen Yuan, and Yihan Zhang. Zero-rate thresholds and new capacity bounds for list-decoding and list-recovery. *arXiv preprint*, 2022. doi:10.48550/arXiv.2210.07754.
- 26 Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001. doi:10.1006/JCSS.2000.1730.
- 27 Michael A Tsfasman, SG Vlăduț, and Th Zink. Modular curves, shimura curves, and goppa codes, better than varshamov-gilbert bound. *Mathematische Nachrichten*, 109(1):21–28, 1982.
- 28 RR Varshamov. Estimate of the number of signals in error correcting codes. *Doklady Akad. Nauk, SSSR*, 117:739–741, 1957.
- 29 Lloyd R. Welch, Robert J. McEliece, and Howard Rumsey, Jr. A low-rate improvement on the Elias bound. *IEEE Trans. Inform. Theory*, IT-20:676–678, 1974. doi:10.1109/tit.1974.1055279.
- 30 Jack Wozencraft. List decoding. *Quarter Progress Report*, 48:90–95, 1958.