Sharp Threshold Rates for Random Codes

Venkatesan Guruswami
Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA
venkatg@cs.cmu.edu

Jonathan Mosheiff
Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA
jmosheif@cs.cmu.edu

Nicolas Resch
Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
nresch@cs.cmu.edu

Shashwat Silas
Computer Science Department, Stanford University, CA, USA
silas@stanford.edu

Mary Wootters
Computer Science Department and Electrical Engineering Department,
Stanford University, CA, USA
marykw@stanford.edu

Abstract

Suppose that $\mathcal{P}$ is a property that may be satisfied by a random code $C \subset \Sigma^n$. For example, for some $p \in (0,1)$, $\mathcal{P}$ might be the property that there exist three elements of C that lie in some Hamming ball of radius pn. We say that R^* is the *threshold rate* for $\mathcal{P}$ if a random code of rate $R^* + \varepsilon$ is very likely to satisfy $\mathcal{P}$, while a random code of rate $R^* - \varepsilon$ is very unlikely to satisfy $\mathcal{P}$. While random codes are well-studied in coding theory, even the threshold rates for relatively simple properties like the one above are not well understood.

We characterize threshold rates for a rich class of properties. These properties, like the example above, are defined by the inclusion of specific sets of codewords which are also suitably "symmetric." For properties in this class, we show that the threshold rate is in fact *equal* to the lower bound that a simple first-moment calculation obtains. Our techniques not only pin down the threshold rate for the property $\mathcal{P}$ above, they give sharp bounds on the threshold rate for *list-recovery* in several parameter regimes, as well as an efficient algorithm for estimating the threshold rates for list-recovery in general.

2012 ACM Subject Classification Mathematics of computing → Coding theory

Keywords and phrases Coding theory, Random codes, Sharp thresholds

Digital Object Identifier 10.4230/LIPIcs.ITCS.2021.5

Related Version `https://arxiv.org/abs/2009.04553`

Funding *Venkatesan Guruswami*: Supported by NSF grants CCF-1563742 and CCF-1814603 and a Simons Investigator Award.
Jonathan Mosheiff: Supported by NSF grants CCF-1563742 and CCF-1814603 and a Simons Investigator Award.
Nicolas Resch: Supported by NSF grants CCF-1563742 and CCF-1814603, ERC H2020 grant No.74079 (ALGSTRONGCRYPTO), and a Simons Investigator Award.
Shashwat Silas: Supported by NSF-CAREER grant CCF-1844628, NSF-BSF grant CCF-1814629, a Sloan Research Fellowship, and a Google Graduate Fellowship.
Mary Wootters: Supported by NSF-CAREER grant CCF-1844628, NSF-BSF grant CCF-1814629, and a Sloan Research Fellowship.

Acknowledgements We would like to thank Ray Li for helpful conversations.

© Venkatesan Guruswami, Jonathan Mosheiff, Nicolas Resch, Shashwat Silas, and Mary Wootters;
licensed under Creative Commons License CC-BY

1 Introduction

Random codes are ubiquitous in the theory of error correcting codes: when thinking about the "right" trade-offs for a particular problem, a coding theorist's first instinct may be to try a random code. A *random code* here is simply a random set. That is, let $C \subseteq \Sigma^n$ be chosen so that each $x \in \Sigma^n$ is included in C with probability $|\Sigma|^{-n(1-R)}$ for some parameter R, which is called the (expected[1]) *rate* of the code C. Random codes are used in the proofs of the Gilbert-Varshamov bound, Shannon's channel coding theorem, and the list-decoding capacity theorem, to name just a few. This success may lead to the intuition that random codes are "easy" to analyze, and that the hard part is finding explicit constructions that match (or in rare cases, exceed) the parameters of random codes. However, there is still much we do not know about random codes, especially if we want extremely precise answers.

In particular, the question of *threshold rates*, of broader interest in probability theory, is something that we do not understand well for random codes. In more detail, suppose that $\mathcal{P}$ is a code property. For example, perhaps $\mathcal{P}$ is the property that there is some pair of codewords $c^{(1)}, c^{(2)} \in C$ that both lie in some Hamming ball of radius pn. Or perhaps $\mathcal{P}$ is the property that there are three codewords $c^{(1)}, c^{(2)}, c^{(3)} \in C$ that lie in such a Hamming ball. A value $R^* \in (0,1)$ is a *threshold rate* for $\mathcal{P}$ if a random code of rate $R^* + \varepsilon$ is very likely to satisfy $\mathcal{P}$, but a random code of rate $R^* - \varepsilon$ is very unlikely to satisfy $\mathcal{C}$. For the first example above, about pairs of codewords, the property in question is just the property of the code having *minimum distance* less than $2pn$, and this is not too hard to understand. However, already for the second example above – called *list-of-two decoding* – the threshold rate was not known.

1.1 Contributions

In this paper, we characterize threshold rates for a rich class of natural properties of random codes. We apply our characterization to obtain threshold rates for list-of-two decoding, as well as to properties like *list-decoding* and *perfect hashing codes,* and more generally to *list-recovery.* We outline our contributions below.

A characterization of the threshold rate R^* for symmetric properties

Suppose that $\mathcal{P}$ is a property defined by the inclusion of certain "bad" sets. For example, the list-of-two decoding property described above is defined by the inclusion of three codewords that lie in a radius-pn Hamming ball. For such properties that are also "symmetric enough," our main technical result, Theorem 1, characterizes the threshold rate R^*. Moreover, we show that this threshold rate is exactly the same as the lower bound that one obtains from a simple first-moment calculation! This is in contrast to recent work of [13] for random *linear* codes, which shows that the corresponding first-moment calculation is not the correct answer in that setting.

Part of our contribution is formalizing the correct notion of "symmetric enough." As we describe in the technical overview in Section 1.2, this definition turns out to be fairly subtle. We also show in the full version of the paper, that this definition is necessary.

[1] Throughout, we refer to R as the rate of the code, and drop the adjective "expected."

Estimates of R^* for list-recovery

We give precise estimates of the threshold rate R^* for *list-recovery.* We say that a code $C \subseteq \Sigma^n$ is (p, ℓ, L)-list-recoverable if for all sets $K_i \subseteq \Sigma$ (for $1 \leq i \leq n$) with $|K_i| \leq \ell$,

$$|\{c \in C : \Pr_{i \sim [n]}[c_i \notin K_i] \leq p\}| < L.$$

List-recovery is a useful primitive in list-decoding, algorithm design, and pseudorandomness (see, e.g., [15, 10, 16]). In particular, it generalizes the list-of-two decoding example above (when $\ell = 1$ and $L = 3$), as well as other interesting properties, such as list-decoding and perfect hashing codes, discussed below.

Our characterization allows us to estimate or even exactly compute the threshold rate for (p, ℓ, L)-list-recovery in a wide variety of parameter regimes. To demonstrate this, we include several results along these lines. First, in Section 4 (Corollary 38), we give estimates that are quite sharp when $\frac{q \log L}{L}$ is small. In Section 5 (Lemma 40), we give an exact formula for the case $p = 0$, which is relevant for perfect hashing codes. In Section 6 (Theorem 42(I)), we give an exact formula for the case that $L = 3$ and $\ell = 1$, relevant for list-of-two decoding. Moreover, in Section 7 (Corollary 47) we use our characterization to develop an efficient algorithm to compute the threshold rate up to an additive error of $\varepsilon > 0$; our algorithm runs in time $O_p(L^q + \text{poly}(q, L, \log(1/\varepsilon))$.

List-of-two decoding and a separation between random codes and random linear codes

We obtain new results for list-of-two decoding, the example discussed above. List-of-two decoding is a special case of *list-decoding,* which itself the special case of list-recovery where $\ell = 1$. We say that a code is (p, L)-list-decodable if there is no Hamming ball of radius pn containing L codewords; list-of-two decoding is the special case of $L = 3$.[2] We show in Section 6 (Theorem 42) that the threshold rate for this question, for random binary codes, is $R^* = 1 - \frac{1 - h_2(3p) + 3p \log_2 3}{3}$. That is, above this rate, a random binary code is very likely to have three codewords contained in a radius pn ball, while below this rate, the code most likely avoids all such triples.

This result is interesting for two reasons. First, it demonstrates that our techniques are refined enough to pin down the threshold rate in this parameter regime. Second, the particular value of R^* is interesting because it is *different* than the corresponding threshold rate for random *linear* codes. A *random linear code* over $\mathbb{F}_q$ of rate R is a random linear subspace of $\mathbb{F}_q^n$, of dimension Rn. The list-decodability of random linear codes has been extensively studied, and it is known (e.g., [19, 7]) that the (p, L)-list-decoding threshold rate for both random linear codes and random codes is $1 - h_q(p)$, for sufficiently large list sizes L.[3]

Limitations of random codes for perfect hashing

Another special case of list-recovery is *perfect hashing codes.* Suppose that $|\Sigma| = q$. A code $C \subseteq \Sigma^n$ is said to be a q-hash code if, for any set of q distinct codewords $c^{(1)}, c^{(2)}, \ldots, c^{(q)} \in C$, there is at least one $i \in [n]$ so that $\{c_i^{(1)}, c_i^{(2)}, \ldots, c_i^{(q)}\} = \Sigma$; that is, if the set of symbols that

[2] It is called list-of-*two* decoding, even though L is *three*, because any Hamming ball contains at most *two* codewords.

[3] Here, $h_q(x) = x \log_q(q-1) - x \log_q(x) - (1-x) \log_q(1-x)$ is the q-ary entropy.

appear in position i are all distinct. Thus, C is a q-hash code if and only if it is $(0, q-1, q)$-list-recoverable. As the name suggests, q-hash codes have applications in constructing small perfect hash families, and it is a classical question to determine the largest rate possible for a q-hash code.[4]

A simple random coding argument shows that a random code of rate $R = \frac{1}{q}\log_q \frac{1}{1-q!/q^q} - o(1)$ is a q-hash code with high probability [5, 12]. However, it is still an area of active research to do significantly better than this bound for any q. It is known that $R < \frac{q!}{q^{q-1}}$ for any q-hash code [5, 9], and for large q, there is a gap of a multiplicative factor of about q^2 between these upper and lower bounds. Körner and Matron gave a construction that beats the random bound for $q = 3$ [11], and recently Xing and Yuan gave a construction that beats the random bound for infinitely many q's [18]. One might have hoped that a random code might in fact do better than the straightforward probabilistic argument (which follows from a union bound). Unfortunately, our results show that this is not the case.

A broader view

Taking a broader view, threshold phenomena in other combinatorial domains, notably random graphs and Boolean functions, have been the subject of extensive study at least since Erdős and Rényi's seminal work [3]. Some of the deeper results in this field (e.g. [6]), deal simultaneously with a wide class of properties, rather than a specific one. Other works, such as the recent [4], are general enough to cover not only multiple properties, but also multiple domains. Our work (as with the work of [13], [8] on random linear codes, discussed below) is not as general as these, but we are able to get more precise results. It would be interesting to find a general framework that connects threshold phenomena in a variety of random code models, with analogues from random graphs and other natural combinatorial structures.

1.2 Technical Overview

As mentioned above, we study properties defined by the inclusion of bad subsets. We organize bad subsets of size b into matrices $B \in \Sigma^{n\times b}$, interpreting the columns of B as the elements of the set. We write "$B \subseteq C$" to mean that the columns of B are all contained in the code C.

As a running example – and also our motivating example – consider list recovery, defined above. The property $\mathcal{P}$ of *not* being (p, ℓ, L)-list-recoverable is defined by the inclusion of "bad" matrices $B \in \Sigma^{n\times L}$ so that for some sets $K_1, \ldots, K_n \subset \Sigma$ of size at most ℓ, $\Pr_{i\sim[n]}[B_{ij} \notin K_i] \leq p$ for each $j \in [L]$. Moreover we require the columns of B to be distinct.

Analyzing a property as a union of types

Following the approach of [13] for random linear codes, we group the bad matrices into *types* based on their row distributions. That is, for a bad matrix $B \in \Sigma^{n\times b}$, let τ denote the row distribution

$$\tau(v) = \frac{|\{i \in [n] : B_{i,\star} = v\}|}{n},$$

where $B_{i,\star}$ denotes the i'th row of B. We say that B has *type* τ. Consider the set $\mathcal{B}$ of all of the matrices of type τ; equivalently, $\mathcal{B}$ is the set of matrices obtained by permuting the

[4] A q-hash code naturally gives rise to a perfect hash family: suppose that C is a universe of items, and define a hash function $h_i : C \to \Sigma$ given by $h_i(c) = c_i$. Then the property of being a q-hash code is equivalent to the property that, for any set of q items in the universe, there exists some hash function h_i for $1 \leq i \leq n$ that maps each item to a different value.

rows of B. We note that possible types τ depend on n, because of divisibility constraints. For simplicity, let us ignore these restrictions for now (we will deal with them later), and suppose that a single type τ can appear for all n.

First-moment bound and main theorem

We can use a simple first-moment approach to give a lower bound on the threshold rate. In more detail, the probability that a particular B is contained in C is $q^{-nb(1-R)}$, assuming that B has b distinct columns. Using the fact that $|\mathcal{B}| \approx q^{H_q(\tau)\cdot n}$, where $H_q(\tau)$ is the base-q entropy of τ (see Section 2), and applying a union bound over all $B \in \mathcal{B}$, we see that the probability that any $B \in \mathcal{B}$ is contained in C is at most

$$q^{nb(H_q(\tau)-(1-R))}.$$

Thus, if $R \leq 1 - \frac{H_q(\tau)}{b} - \varepsilon$ for some small $\varepsilon > 0$, it is very unlikely that τ will be represented in C.

Now suppose that our collection of bad sets, which define the property $\mathcal{P}$, is closed under row permutations. This means that $\mathcal{P}$ can be represented as a collection T of types τ; note that the size of T is polynomial in n. Union bounding over all of these types, the computation above shows that a random code C of rate $R < 1 - \max_{\tau \in T} \frac{H_q(\tau)}{b} - \varepsilon$ will, with high probability, not satisfy $\mathcal{P}$.

The question is, could the rate be larger? Might it be the case that $\mathcal{P}$ still not satisfied (with high probability) by a random code of rate R significantly larger than $1 - \max_\tau H_q(\tau)/b$? In [13], it was shown that the answer for random *linear* codes is "yes." If $\mathcal{P}$ exhibits certain linear structure, then it may be possible that a higher rate random linear code still does not satisfy $\mathcal{P}$ with high probability. One may conjecture that something similar holds for random codes.

Our main technical result, Theorem 30, is that, for random codes, for sufficiently symmetric properties, the answer to this question is "no." That is, the simple calculation above *does* give the right answer for random codes!

▶ **Theorem 1** (Informal; see Theorem 30 for the formal version)**.** *Let $\mathcal{P}$ be a "symmetric" property defined by the inclusion of a type among the types in T. Let*

$$R^* = 1 - \frac{\max_{\tau \in T} H_q(\tau)}{b}$$

Then for all $\varepsilon > 0$, a random code of rate $R \geq R^* + \varepsilon$ satisfies $\mathcal{P}$ with probability $1 - o(1)$, while a random code of rate $R^* - \varepsilon$ satisfies $\mathcal{P}$ with probability $o(1)$.

Sketch of proof: second moment method

Below, we sketch the proof of Theorem 1, and explain what the assumption of "symmetry" means. As noted above, it is straightforward to show that the threshold rate R^* is at least $1 - \max_{\tau \in T} \frac{H_q(\tau)}{b}$, so the challenge is to show that it is not larger. The proof of Theorem 1 uses the second-moment method to show that for any *histogram type* τ (we discuss histogram types more below), a random code C of rate $1 - H_q(\tau)/b + \varepsilon$ is very likely to contain some matrix B with type τ. Thus, the threshold rate is at most $1 - \max_\tau H_q(\tau)/b$, where the maximum is over all histogram types τ that appear in T. Our eventual definition of "symmetric" will guarantee that it is legitimate to restrict our attention to histogram types.

Histogram types and the meaning of "symmetry"

In order to apply the second moment method, we bound the variance of $\sum_{B \sim \tau} \mathbf{1}[B \subset C]$, where the sum is over all matrices B of type τ. This turns out to be possible when τ has the following symmetry property: for any $u \in \Sigma^b$, and for any permutation $\pi : [b] \to [b]$, it holds that $\tau(u) = \tau(\pi(u))$, where $\pi(u)$ denotes the corresponding coordinate permutation of u. We call such a type τ a *histogram-type* (Definition 27) because the probability of a particular vector u under τ depends only on the histogram of u.

A first attempt to formulate a definition of "symmetry" for Theorem 1 is thus to require $\mathcal{P}$ to be defined by histogram types. This results in a true statement, but unfortunately it is too restrictive: it is not hard to see that, for example, the property of not being list-decodable contains types τ that are not histogram types. Fortunately, for the logic above to go through, it is enough to show that T contains a type τ that is *both* a maximum entropy distribution in T, and is also a histogram type. Thus, the assumption of "symmetry" we will use is that T, the collection of types represented in the property $\mathcal{P}$, forms a convex set. Then, using the fact that $\mathcal{P}$ is defined by the inclusion of bad sets (which do not care about the order of the columns in the corresponding matrices), we can always find a maximum entropy histogram type by "symmetrizing" and taking a convex combination of column permutations of some maximum entropy type τ. One might wonder if this symmetrization step (and the resulting assumption about convexity) is necessary. In the full version of this paper show that it is.

Taking a limit as $n \to \infty$

There is one more challenge to consider, which is that in the description above, we have ignored the fact that we would like our characterization to work for a sequence of values of n. However, a type τ only works for certain values of n due to divisibility restrictions. To get around this, we work instead with a sequence of types τ_n which tend to τ. This leads us to our final definition of "symmetric" (Definition 20). Suppose that $\mathcal{P}$ is a property defined by the inclusion of size-b bad sets. Then for each n, there is some collection T^n of bad types τ_n, each of which is a distribution on Σ^b. We say that $\mathcal{P}$ is *symmetric* if the sets T^n approach some convex set T as n goes to infinity. The logic above then goes through to give Theorem 1.

Applications to list-recovery

Finally, in order to apply Theorem 1, we need to understand the maximum entropy distribution τ for our property $\mathcal{P}$. We do this for the property $\mathcal{P}$ of not being (p, ℓ, L)-list-recoverable in a variety of parameter regimes in Sections 4, 5 and 6, and along the way obtain our results about list-of-two decoding and perfect hashing codes. Finally, in Section 7, we use our framework to develop an algorithm to efficiently calculate the threshold rate for (p, ℓ, L)-list-recovery.

1.3 Organization

In Section 2, we introduce notation, and also set up the definitions we need about types, thresholds, properties, and various notions of symmetry. We also introduce (non-)list-recoverability as a property, and prove in Corollary 24 that it is symmetric.

In Section 3, we state and prove Theorem 30, the formal version of the characterization theorem (Theorem 1 above). At the end of Section 3, we begin to apply Theorem 30 to list-recovery, and in particular define several notions we will need to analyze list recovery in the subsequent sections.

In the remaining sections, we specialize to list-recovery. Note that the proofs of the claims in the remaining sections are available in the full version. In Section 4, we develop bounds on the threshold rate R^* for list-recovery that are tight when $(q \log L)/L$ is small. In Section 5, we compute the threshold rate R^* exactly for zero-error list-recovery (that is, when $p = 0$), and use this to compute the threshold rate for perfect hashing. In Section 6, we compute the threshold rate R^* for list-of-two decoding (e.g., list-recovery when $\ell = 1$ and $L = 3$), and use this to quantify the gap between random codes and random linear codes for list-of-two decoding. Finally, in Section 7, we give an efficient algorithm to compute the threshold rate.

2 Preliminaries

First, we fix some basic notation. Throughout, we consider codes $C \subseteq \Sigma^n$ of block length n over an alphabet Σ, where $|\Sigma| = q$. When we use $\log(x)$ without an explicit base, we mean $\log_2(x)$. We use H_q to denote the base-q entropy: for a distribution τ,

$$H_q(\tau) := -\sum_x \tau(x) \log_q(\tau(x)).$$

When q is clear from context, we will use $H(\tau)$ to denote $H_q(\tau)$. If u is a random variable distributed according to τ, then we abuse notation slightly and define $H(u) := H(\tau)$. We use $h_q(x) := x \log_q(q-1) - x \log_q(x) - (1-x) \log_q(1-x)$ to denote the q-ary entropy of $x \in (0,1)$. Again, when q is clear from context we will use $h(x)$ to denote $h_q(x)$.

For a vector $x \in \Sigma^k$ and $I \subseteq [k]$, we use x_I to refer to the vector $(x_i)_{i \in I} \in \Sigma^I$. Given a vector $u \in \Sigma^k$ and a permutation $\pi : [k] \to [k]$, we let $\pi(u) \in \Sigma^k$ denote the corresponding coordinate permutation of u.

Given distributions τ, μ on the same finite set, we define their ℓ_∞-distance by

$$d_\infty(\tau, \mu) := \max_x |\tau(x) - \mu(x)| \ .$$

Given a set of distributions T, we define the ℓ_∞ distance from μ to T by

$$d_\infty(\mu, T) := \inf_{\tau \in T} d_\infty(\mu, \tau) \ .$$

2.1 Basic notions

As mentioned in the introduction, we will organize our "bad" sets into matrices. We formalize this with the following two definitions.

▶ **Definition 2** (Matrices with distinct columns). *Let $\Sigma^{n \times b}_{\text{distinct}}$ denote the collection of all matrices $B \in \Sigma^{n \times b}$ such that each column of B is distinct.*

▶ **Definition 3** (Subsets as matrices). *Let $C \subseteq \Sigma^n$ be a code, and let $B \in \Sigma^{n \times b}$ be a matrix. We write $B \subseteq C$ to mean that each column of B is an element of C. If $A \subseteq \Sigma^n$, let $\mathcal{B}_A \subseteq \Sigma^{n \times |A|}$ denote the collection all matrices $B \in \Sigma^{n \times |A|}_{\text{distinct}}$ such that the columns of B are the elements of A.*

For completeness, we reiterate our definition of a random code from the introduction.

▶ **Definition 4** (Random code). *Let Σ be a finite set with $q := |\Sigma| \geq 2$. For $n \in \mathbb{N}$ and $R \in [0,1]$, let $C^n_{\text{RC}}(R)$ denote an* expected-rate R random code (over the alphabet Σ) *$C \subseteq \Sigma^n$. Namely, for each $x \in \mathbb{F}_q^n$ we have $\Pr[x \in C] = q^{-n(1-R)}$, and these events are independent over all x.*

We record a useful fact about random codes, which is the probability that any particular matrix B is contained in one.

▶ **Fact 5** (Probability that a random code contains a matrix). *Let $B \in \Sigma^{n\times b}$. Then,*

$$\Pr\left[B \subseteq C^n_{\mathrm{RC}}(R)\right] = q^{-n(1-R)t},$$

where t is the number of distinct columns in B.

We study (noisy) list-recovery, which generalizes both the list-decoding and perfect hashing examples mentioned in the introduction. We repeat the definition, so that we may formally define a "bad" matrix for list-recovery.

▶ **Definition 6** (Noisy list-recovery). *Let $p \in [0,1]$, $1 \le \ell \le q$, and $L \in N$. Say that a matrix $B \in \Sigma^{L\times n}_{\mathrm{distinct}}$ is (p,ℓ,L)-*bad *for (p,ℓ,L)-list-recovery if there exist sets $K_i \subseteq \Sigma$ $(1 \le i \le n)$, each of size ℓ, such that for every $1 \le j \le L$,*

$$\Pr_{i\sim[n]}\left[B_{i,j} \notin K_i\right] \le p.$$

A code $C \subseteq \Sigma^n$ is (p,ℓ,L)-list-recoverable *if it does not contain a (p,ℓ,L)-bad matrix.*

2.2 Monotone-increasing properties and thresholds

We study the threshold rate R^* for random codes to satisfy certain properties. This was discussed informally in the introduction and the definitions below formalize what "threshold rate" means.

▶ **Definition 7** (Monotone-increasing property). *A code property $\mathcal{P}$ is* monotone-increasing *if given a code C satisfying $\mathcal{P}$, it holds that every code C' such that $C \subseteq C'$ also satisfies $\mathcal{P}$.*

For example, the property of being *not* (p,ℓ,L)-list-recoverable (that is, the property of containing a (p,ℓ,L)-bad matrix) is a monotone-increasing property.

▶ **Definition 8** (Minimal-set). *Let P_n be a monotone-increasing property of length-n codes. A set $A \subseteq \Sigma^n$ is a* minimal element *of P_n if A satisfies P_n but no strict subset of A satisfies P_n. The* minimal set *for P_n is the collection of matrices*

$$\bigcup_{A \text{ is a minimal element of } P_n} \mathcal{B}_A.$$

For example, the minimal set for the property P_n of being *not* (p,ℓ,L)-list-recoverable is the set of (p,ℓ,L)-bad matrices.

Note that a code satisfies P_n if and only if it contains some matrix belonging to the minimal set of P_n. If $\mathcal{P}$ is a monotone-increasing property of codes, we define its associated *threshold rate* by

$$R^n_{\mathrm{RC}}(\mathcal{P}) := \begin{cases} \sup\left\{R \in [0,1] : \Pr\left[C^n_{\mathrm{RC}}(R) \text{ satisfies } \mathcal{P}\right] \le \frac{1}{2}\right\} & \text{if there is such an } R \\ 0 & \text{otherwise.} \end{cases}$$

▶ Remark 9. If $\mathcal{P}$ is monotone-increasing then the function $\Pr\left[C^n_{\mathrm{RC}}(R)\text{satisfies } \mathcal{P}\right]$ is monotone-increasing in R. This can be proved by a standard coupling argument, akin to [1, Thm. 2.1].

▶ **Definition 10** (Sharpness for random codes). *A monotone-increasing property $\mathcal{P}$ is* sharp for random codes *if*

$$\lim_{n\to\infty} \Pr\left[C^n_{\mathrm{RC}}\left(R^n_{\mathrm{RC}}(\mathcal{P}) - \varepsilon\right) \text{ satisfies } \mathcal{P}\right] = 0$$

and

$$\lim_{n\to\infty} \Pr\left[C^n_{\mathrm{RC}}\left(R^n_{\mathrm{RC}}(\mathcal{P}) + \varepsilon\right) \text{ satisfies } \mathcal{P}\right] = 1$$

for every $\varepsilon > 0$.

2.3 Local and row-symmetric properties

As discussed in the introduction, we study properties that can be written as a union of "types," where each type corresponds to a row distribution τ of a matrix M. The following definitions make this notion precise.

▶ **Definition 11** (Row-permutation invariant collection of matrices). *A collection of matrices $\mathcal{B} \subseteq \Sigma^{n\times b}$ is* row-permutation invariant *if, given a matrix $B \in \mathcal{B}$, every row permutation of B (that is, a matrix resulting from applying the same coordinate permutation to each column of B) also belongs to $\mathcal{B}$.*

▶ **Definition 12** (Local and row-symmetric properties). *Let $\mathcal{P} = \{P_n\}_{n\in\mathbb{N}}$ be a monotone-increasing property, and let M_n denote the minimal set of P_n.*

- *If there exists some $b \in \mathbb{N}$ such that $M_n \subseteq \Sigma^{n\times b}$ for every n, we say that $\mathcal{P}$ is* b-local.
- *If every M_n is row-permutation invariant, we say that $\mathcal{P}$ is* row-symmetric.

▶ Remark 13. Every monotone-increasing property is trivially *column-symmetric*, in the sense that permuting the columns of a matrix in M_n results in another matrix in M_n. This naturally reflects the fact that containment of a matrix does not depend on the ordering of the columns, and follows immediately from the definition of a minimal set.

Let $B \in \Sigma^{n\times b}$, and consider the collection $\mathcal{B}$ of all row-permutations of B. Let τ_B denote the *row-distribution* of B. That is, τ is the probability distribution, over Σ^b, of the row $B_{i,\star}$, where i is sampled uniformly from $[n]$. Observe that every matrix in $\mathcal{B}$ has the same row-distribution as B. Moreover, $\mathcal{B}$ can be characterized as the set of all matrices with the row distribution τ_B. These observations motivate the following definitions.

▶ **Definition 14** (Type of a matrix). *Let $B \in \Sigma^{n\times b}$. We define its* type *τ_B as the distribution of a uniformly random row of B. That is, τ_B is the distribution over Σ^b, such that*

$$\tau_B(x) = \frac{|\{i \in [n] \mid B_i = x\}|}{n}$$

for every $x \in \Sigma^b$. Let

$$\mathcal{T}^n_b = \{\tau_B \mid B \in \Sigma^{n\times b}_{\mathrm{distinct}}\}$$

denote the set of all possible types of $n \times b$ matrices with distinct columns. Given $\tau \in \mathcal{T}^n_b$, we denote

$$M_\tau = \{B \in \Sigma^{n\times b} \mid \tau_B = \tau\}.$$

▶ Remark 15. The type of a matrix $B \in \Sigma^{n\times b}$ determines whether $B \in \Sigma^{n\times b}_{\text{distinct}}$. Therefore, for $\tau \in \mathcal{T}^n_b$,

$$M_\tau = \{B \in \Sigma^{n\times b}_{\text{distinct}} \mid \tau_B = \tau\}.$$

The following fact now follows from the above discussion.

▶ **Fact 16** (Decomposition of a row-permutation invariant collection). *Let $\mathcal{B} \subseteq \Sigma^{n\times b}$ be a row-permutation invariant collection. Then, there exists a set of types $T \subseteq \mathcal{T}_{n,b}$ such that*

$$\mathcal{B} = \bigcup_{\tau\in T} M_\tau.$$

Note that a type in $\mathcal{T}^n_b$ is defined by the number of occurrences of each of $|\Sigma^b|$ possible rows, in a matrix consisting of n rows. In particular, each row occurs between 0 and n times. Thus,

$$|\mathcal{T}^n_b| \le (n+1)^{|\Sigma^b|} = (n+1)^{q^b}.$$

Crucially for our purposes, this upper bound is polynomial in n.

2.4 Symmetric properties and convex approximations

▶ **Definition 17.** *Let $\mathcal{T}_b$ denote the simplex of all probability distributions over Σ^b.*

It is generally more convenient to work in $\mathcal{T}_b$ rather than $\mathcal{T}^n_b$, since the former is continuous, while the latter is discrete and involves certain divisibility conditions. This motivates the following definition.

▶ **Definition 18** (Permutation-closed type sets). *A set $T \subseteq \mathcal{T}_b$ is called* permutation-closed *if for every $\tau \in T$ and every permutation $\pi : [b] \to [b]$, the distribution of $\pi(u)$ (where $u \sim \tau$) also belongs to T.*

▶ **Definition 19** (Approximating sets of types). *Fix $b \in \mathbb{N}$. Let $\{T^n\}_{n\in\mathbb{N}}$ be a sequence of sets of types, such that $T^n \subseteq \mathcal{T}^n_b$. A (topologically) closed and permutation-closed set $T \subseteq \mathcal{T}_b$ is an* approximation *for $\{T^n\}_{n\in\mathbb{N}}$ if $T^n \subseteq T$ for every n, and*

$$\lim_{n\to\infty} \max_{\tau\in T} d_\infty(\tau, T^n) = 0.$$

▶ **Definition 20** (Symmetric property and convex approximation). *Let $\mathcal{P} = \{P_n\}_{n\in\mathbb{N}}$ be a b-local, row-symmetric, monotone-increasing property. Due to Fact 16, for every n there exists a set $T_n \subseteq \mathcal{T}_{n,b}$ such that the minimal set of P_n is $\bigcup_{\tau\in T_n} M_\tau$. If the sequence $\{T_n\}_{n\in N}$ has a convex approximation T, we say that T is a* convex approximation *for $\mathcal{P}$. In this case, we say that $\mathcal{P}$ is* symmetric*.*

2.5 Non-list-recoverability as a property

Our motivating property is that of being *not* list-recoverable. In this section, we show that non-(p,ℓ,L)-list-recoverability is a symmetric property, and we define the convex set $T_{p,\ell,L}$ that is a convex approximation for it.

Fix $p \in [0,1]$, $1 \le \ell \le q$ and $L \in \mathbb{N}$. Let $\mathcal{P} = (P_n)_{n\in\mathbb{N}}$ denote the property of being *not* (p,ℓ,L)-list-recoverable. That is, a code $C \subseteq \Sigma^n$ satisfies P_n if it contains a (p,ℓ,L)-bad matrix. We now show that $\mathcal{P}$ is a symmetric property.

Clearly, $\mathcal{P}$ is monotone-increasing, and its minimal set is exactly the set of (p,ℓ,L)-bad matrices, which we denote $M_n \subseteq \Sigma_{\text{distinct}}^{n\times L}$. It follows immediately that $\mathcal{P}$ is L-local. Furthermore since the left-hand side of (6) is invariant to row-permutations of B, the collection M_n is row-permutation invariant, and so $\mathcal{P}$ is row-symmetric.

Fact 16 says that we can write $M_n = \bigcup_{\tau\in T^n_{p,\ell,L}} M_\tau$ for some $T^n_{p,\ell,L} \subseteq \mathcal{T}^n_L$. Indeed, (6) yields the following description of $\mathcal{T}^n_{p,\ell,L}$: A type $\tau \in \mathcal{T}^n_L$ belongs to $T^n_{p,\ell,L}$ if and only if there exists a distribution ρ over $\Sigma^L \times \binom{\Sigma}{\ell}$ such that, given $(u,K)\sim\rho$, the following holds:

1. The distribution of u is τ.
2. For every $1 \le j \le L$, it holds that $\Pr[u_j \notin K] \le p$.
3. $n\cdot\rho((u,K)) \in \mathbb{N}$ for every u and K.

To see this, let ρ be the joint distribution (B_i, K_i) for i uniformly sampled from $[n]$, where B and K are as in (6). Note that ρ must satisfy the three conditions above. In the other direction, it is not hard to see that any such distribution ρ as above gives rise to a matrix of type τ, satisfying (6).

We next construct a convex approximation for $\mathcal{P}$. Let $T_{p,\ell,L}$ denote the set of all types $\tau \in \mathcal{T}_L$ for which there exists a distribution ρ satisfying Conditions 1 and 2, but not necessarily Condition 3:

▶ **Definition 21.** *Let $1 \le \ell \le 1$, $L \in \mathbb{N}$ and $0 \le p \le 1$. Let τ be a distribution over Σ^L. We say that τ belongs to the set $T_{p,\ell,L}$ if there exists a distribution ρ over $\Sigma^L \times \binom{\Sigma}{L}$ such that:*

1. *If $(u,K)\sim\rho$ then the vector u is τ-distributed.*
2. *For every $1 \le j \le L$ it holds that*

$$\Pr_{(u,K)\sim\rho}[u_j \notin K] \le p.$$

Clearly, $T^n_{p,\ell,L} \subseteq T_{p,\ell,L}$ for all $n \in \mathbb{N}$. It is also immediate to verify that $T_{p,\ell,L}$ is permutation-closed.

▶ **Lemma 22.** *The set $T_{p,\ell,L}$ is convex.*

Proof. Let $\tau_0, \tau_1 \in T_{p,\ell,L}$. Let $t \in [0,1]$ and let τ_t denote the mixture distribution $(1-t)\tau_0 + t\tau_1$. Let ρ_0 and ρ_1 be distributions over $\Sigma^L \times \binom{\Sigma}{\ell}$, satisfying Conditions 1 and 2 for τ_0 and τ_1, respectively. Let ρ_t be the mixture distribution $(1-t)\rho_0 + t\rho_1$. It is straightforward to verify that ρ_t satisfies Conditions 1 and 2 with respect to τ_t. Hence, $\tau_t \in T_{p,\ell,L}$. ◀

The following lemma, proven in the appendix of the full version of this paper, shows that $T_{p,\ell,L}$ satisfies (19). Namely, every type in $T_{p,\ell,L}$ can be realized with low error as a type from $T^n_{p,\ell,L}$, for large enough n.

▶ **Lemma 23.**

$$\lim_{n\to\infty} \sup_{\tau\in T_{p,\ell,L}} d_\infty(\tau, T^n_{p,\ell,L}) = 0.$$

We record the results of this section in the following corollary.

▶ **Corollary 24.** *Being* not (p,ℓ,L)*-list-recoverable is a symmetric property. Furthermore, $T_{p,\ell,L}$ is a convex approximation for this property.*

3 Characterization theorem

In this section, we prove our main characterization theorem, Theorem 1, which is formally stated below as Theorem 30. Before stating and proving the theorem, we record a few useful lemmas.

▶ **Lemma 25** ([2, Lemma 2.2]). *Let $\tau \in \mathcal{T}_b^n$. Then,*

$$q^{H(\tau)n} \cdot n^{-O_{q,b}(1)} \leq |M_\tau| \leq q^{H(\tau)n}.$$

▶ **Lemma 26.** *Let $M \subseteq \Sigma^{n \times b}$. Then,*

$$|M| \leq n^{q^b} \cdot q^{n \cdot \max_{B \in M} H(\tau_B)}.$$

Proof. Let $T = \{\tau_B \mid B \in M\}$. Note that

$$M \subseteq \bigcup_{\tau \in T} M_\tau.$$

Thus,

$$|M| \leq \sum_{\tau \in T} |M_\tau| \leq |T| \cdot \max_{\tau \in T} |M_\tau| \leq |\mathcal{T}_{n,b}| \cdot \max_{\tau \in T} |M_\tau|.$$

The claim follows from (2.3) and Lemma 25. ◀

We say that a type is a *histogram type* if it is indifferent to the ordering of a given vector's entries, and thus, only cares about the histogram of the vector. Formally, we make the following definition.

▶ **Definition 27** (Histogram type). *A type $\tau \in T_b$ is called a* histogram-type *if $\tau(u) = \tau(\pi(u))$ for every $u \in \Sigma^b$ and every permutation $\pi : [b] \to [b]$.*

▶ **Lemma 28.** *Let $T \subseteq \mathcal{T}_b$ be a closed, permutation-closed, convex, set of types. Then there exists a histogram type $\tau \in T$ such that $H(\tau) = \max_{\tau' \in T} H(\tau')$.*

Proof. Since T is closed and bounded, it is compact. Thus, there is some $\tau' \in T$ such that $H(\tau')$ is maximal. Given a permutation $\pi : [b] \to [b]$, let $\pi(\tau')$ denote the distribution of the vector $\pi(u)$, where $u \sim \tau$. Let

$$\tau = \frac{\sum_{\pi \in \mathrm{Sym}_b} \pi(\tau')}{b!}.$$

Since T is permutation-closed and convex, $\tau \in T$. By concavity of entropy,

$$H(\tau) \geq \frac{\sum_{\pi \in \mathrm{Sym}_b} H(\pi(\tau'))}{b!} = \frac{\sum_{\pi \in \mathrm{Sym}_k} H(\tau')}{b!} = H(\tau').$$

Thus, τ has maximum entropy in T, and is clearly a histogram-type. ◀

The following technical lemma, proven in the appendix of the full version, facilitates our use of an approximation for a set of types.

▶ **Lemma 29.** *Let $\tau, \tau' \in T_b$ such that $d_\infty(\tau, \tau') \leq \varepsilon$. Then,*

$$|H_{u \sim \tau}(u \mid u_I) - H_{u \sim \tau'}(u \mid u_I)| \leq O_{b,q}\left(\varepsilon \cdot \log \frac{1}{\varepsilon}\right)$$

for any $I \subseteq [b]$.

We now prove that every monotone-increasing, local and row-symmetric property with a convex approximation is sharp for random codes. Furthermore, we identify the threshold rate as the maximal entropy in the approximating set.

▶ **Theorem 30.** *Fix $b \in \mathbb{N}$. Let $\mathcal{P} = \{P_n\}_{n\in\mathbb{N}}$ be a symmetric property with locality parameter b, and let T be a convex approximation for $\mathcal{P}$. Denote $R^* = 1 - \frac{\max_{\tau\in T} H(\tau)}{b}$. Fix $\varepsilon > 0$ and let $R \in [0,1]$. The following now holds.*

1. *If $R \le R^* - \varepsilon$ then*

$$\lim_{n\to\infty} \Pr\left[C^n_{\mathrm{RC}}(R) \text{ satisfies } \mathcal{P}\right] = 0.$$

2. *If $R \ge R^* + \varepsilon$ then*

$$\lim_{n\to\infty} \Pr\left[C^n_{\mathrm{RC}}(R) \text{ satisfies } \mathcal{P}\right] = 1.$$

Proof. For $b \in \mathbb{N}$ and a matrix $B \in \Sigma^{b\times n}_{\mathrm{distinct}}$, let X_B be an indicator variable for the event that $B \in C^n_{\mathrm{RC}}(R)$. For a set $M \subseteq \Sigma^{b\times n}_{\mathrm{distinct}}$, let $X_M = \sum_{B\in M} X_B$. By Fact 5,

$$\mathbb{E}\left[X_M\right] = |M| \cdot q^{-n(1-R)b}.$$

Let M_n denote the minimal set for P_n and let $T_n = \{\tau_B \mid B \in M_n\}$.

The first statement now follows from Markov's inequality, (3), and Lemma 26:

$$\begin{aligned}
\Pr\left[C \text{ satisfies } \mathcal{P}\right] &= \Pr\left[\exists B \in M_n \;\; B \subseteq C^n_{\mathrm{RC}}(R)\right] \\
&\le \Pr\left[X_M \ge 1\right] \\
&\le \mathbb{E}\left[X_M\right] \\
&= |M| \cdot q^{-n(1-R)b} \\
&\le n^{q^b} \cdot q^{n\cdot\max_{\tau\in T_n} H(\tau)} \cdot q^{-n(1-R)b} \\
&\le n^{q^b} \cdot q^{n\cdot\max_{\tau\in T} H(\tau)} \cdot q^{-n(1-R)b} \\
&\le n^{q^b} \cdot q^{-nb\varepsilon} \le e^{-\Omega(n)}.
\end{aligned}$$

Above, we used the fact that $T_n \subseteq T$.

For the second statement, let $\tau \in T$ have maximum entropy. By definition 19, T is closed and permutation-closed, in addition to being convex. Consequently, due to Lemma 28, we may assume that τ is a histogram-type. Let $\tau_n \in T_n$ such that $d_\infty(\tau, \tau_n) = o_{n\to\infty}(1)$. Our plan is to use a second-moment argument to show that $C^n_{\mathrm{RC}}(R)$ likely contains a matrix of type τ_n.

By (3) and Lemma 25,

$$\mathbb{E}\left[X_{M_{\tau_n}}\right] = |M_{\tau_n}| q^{-n(1-R)b} \ge q^{(H(\tau_n)-(1-R)b)+o(1)} \ge q^{(H(\tau)-(1-R)b)+o(1)}.$$

We turn to bounding the variance of $X_{M_{\tau_n}}$. Fact 5 yields

$$\begin{aligned}
\mathrm{Var}\left[X_{M_{\tau_n}}\right] &= \sum_{B,B'\in M_{\tau_n}} \left(\Pr\left[X_B = X_{B'} = 1\right] - \Pr\left[X_B = 1\right]\Pr\left[X_{B'} = 1\right]\right) \\
&= \sum_{B,B'\in M_{\tau_n}} \left(q^{-n(1-R)(2b-\alpha(B,B'))} - q^{-2n(1-R)b}\right) \\
&\le \sum_{\substack{B,B'\in M_{\tau_n} \\ \alpha(B,B')\ge 1}} q^{-n(1-R)(2b-\alpha(B,B'))}
\end{aligned}$$

where $\alpha(B,B')$ is the number of columns in B' that also appear in B.

In order to bound this sum, we need an estimate on the number of pairs B, B' with a given $\alpha(B, B')$. For $0 \le r \le b$, let

$$W_r = \{(B, B') \mid B, B' \in M_{\tau_n} \text{ and } \alpha(B, B') = r\}$$

and denote $S_r = \{\tau_{B\|B'} \mid (B, B') \in W_r\}$. Here, $B\|B'$ is the $n \times 2b$ matrix whose first (resp. last) b columns are B (resp. B'). By Lemma 26,

$$|W_r| \le n^{2q^b} \cdot q^{n \max_{\nu \in S_r} H(\nu)}.$$

Let $(B, B') \in W_r$ and let $\nu = \tau_{B\|B'}$. Assume without loss of generality that the first r columns of B are identical to the first r columns of B'. Let $u \sim \nu$. Note that, since $B, B' \in M_{\tau_n}$, the random variables $u_{[b]}$ and $u_{[2b]\setminus[b]}$ are both τ_n-distributed. Hence,

$$\begin{aligned} H(\nu) = H(u) &= H(u_{[2b]\setminus[b]}) + H(u_{[b]} \mid u_{[2b]\setminus[b]}) = H(\tau_n) + H(u_{[b]} \mid u_{[2b]\setminus[b]}) \\ &\le H(\tau_n) + H(u_{[b]} \mid u_{[r]}) = H(\tau_n) + H(u_{[b]\setminus[r]} \mid u_{[r]}). \end{aligned}$$

Lemma 29 yields

$$\begin{aligned} H(u_{[b]\setminus[r]} \mid u_{[r]}) &\le H_{v\sim\tau}(v_{[b]\setminus[r]} \mid v_{[r]}) + o(1) \\ &= \sum_{i=r+1}^{b} H_{v\sim\tau}(v_i \mid v_{[i-1]}) + o(1) \\ &= \sum_{i=r+1}^{b} H_{v\sim\tau}(v_b \mid v_{[i-1]}) + o(1), \end{aligned}$$

where the last equality is due to τ being a histogram-type. Writing

$$f(r) = \sum_{i=r+1}^{b} H_{v\sim\tau}(v_b \mid v_{[i-1]}),$$

we conclude that

$$H(\nu) \le f(r) + H(\tau) + o(1),$$

so that

$$|W_r| \le q^{(f(r)+H(\tau))n+o(n)},$$

and

$$\begin{aligned} \mathrm{Var}\left[X_{M_{\tau_n}}\right] &\le \sum_{r=1}^{b} |W_r| \cdot q^{-n(1-R)(2b-r)} \le \sum_{r=1}^{b} q^{(f(r)+H(\tau)-(1-R)(2b-r))n+o(n)} \\ &\le \max_{1\le r\le b} q^{(f(r)+H(\tau)-(1-R)(2b-r))n+o(n)}. \end{aligned}$$

By Chebyshev's inequality,

$$\Pr\left[X_{M_{\tau_n}} = 0\right] \le \frac{\mathrm{Var}\left[X_{M_{\tau_n}}\right]}{\mathbb{E}\left[X_{M_{\tau_n}}\right]^2} \le \max_{1\le r\le b} q^{(f(r)-H(\tau)+r(1-R))n+o_{b,q}(n)}.$$

We claim that $(f(r))_{r=0}^{b}$ is a convex sequence. Indeed,

$$f(r-1) + f(r+1) - 2f(r) = H_{v\sim\tau}(v_b \mid v_{[r-1]}) - H_{v\sim\tau}(v_b \mid v_{[r]}) \ge 0.$$

Therefore, the maximum in the right-hand side of (3) is achieved either by $r = 1$ or $r = b$. In the former case, note that

$$f(1) = \sum_{i=2}^{b} H_{v\sim\tau}(v_b \mid v_{[i-1]}) = \sum_{i=2}^{b} H_{v\sim\tau}(v_i \mid v_{[i-1]}) = H_{v\sim\tau}(v \mid v_1)$$
$$\leq H(\tau) - H_{v\sim\tau}(v_1) \leq H(\tau) \cdot \frac{b-1}{b}.$$

In the last inequality above, we used the fact that $H_{v\sim\tau}v_1 = H_{v\sim\tau}v_i$ for all $i \in [b]$, due to τ being a histogram-type. Thus, for $r = 1$, the corresponding exponent in (3) is

$$(f(1) - H(\tau) + (1-R))n \leq \left((1-R) - \frac{H(\tau)}{b}\right) n \leq -\varepsilon n.$$

In the latter case, since $f(b) = 0$, the exponent is

$$(-H(\tau) + (1-R)b)n \leq -\varepsilon bn.$$

We conclude that

$$\Pr\left[C^n_{\mathrm{RC}}(R) \text{ does not satisfy } \mathcal{P}\right] \leq \Pr(X_{M_\rho} = 0) \leq q^{-\varepsilon n + o(n)}.$$ ◀

Applying the framework to list-recovery

In the rest of the paper, we use Theorem 30 to compute the threshold rate for (p, ℓ, L) list-recovery in several different settings. In order to do that, we set up a few useful definitions.

▶ **Definition 31** ($\beta(p,\ell,L)$ and $\bar{T}_{p,\ell,L}$). *Given $L \in \mathbb{N}$, $\ell \leq L$ and $p \in [0,1)$, let $\bar{T}_{p,\ell,L}$ denote the set of all histogram-types in $T_{p,\ell,L}$. Let*

$$\beta(p,\ell,L) = \max_{\tau \in \bar{T}_{p,\ell,L}} H(\tau).$$

Theorem 30 allows us to characterize the threshold rate for (p, ℓ, L)-list recovery in terms of $\beta(p, \ell, L)$:

▶ **Corollary 32.** *Fix $L \in \mathbb{N}$, $\ell \leq L$ and $p \in [0,1)$. The threshold rate for (p,ℓ,L) list-recovery is*

$$R^* = 1 - \frac{\beta(p,\ell,L)}{L}.$$

Proof. By Corollary 24 and Lemma 28,

$$\beta(p,\ell,L) = \max_{\tau \in T_{p,\ell,L}} H(\tau).$$

The claim now follows from Corollary 24 and Theorem 30. ◀

Finally, we introduce the following notation, which will be used for the rest of the paper.

▶ **Definition 33** ($P_\ell(\cdot)$ and $D_{d,\ell,L}$). *Fix $\ell \leq L$. Given a vector $v \in \Sigma^L$ let*

$$P_\ell(v) = \min_{A \in \binom{\Sigma}{\ell}} |\{i \in [L] \mid v_i \notin A\}|$$

We use the notation $D_{d,\ell,L} = \{v \in \Sigma^L \mid P_\ell(v) = d\}$.

4 Bounds on the threshold rate for noisy list-recovery

The main result in this section is an estimate of $\beta(p, \ell, L)$ (Proposition 37 below), which leads to an estimate on the threshold rate for list-recovery (Corollary 38). This estimate is very sharp when $\frac{q \log L}{L}$ is small; in subsequent sections we will derive estimates which are more precise for certain parameter regimes.

Before coming to these bounds, we begin with a few useful lemmas that bound $|D_{d,\ell,L}|$ and characterize $\bar{T}_{p,\ell,L}$.

▶ **Lemma 34.** *Let $r = 1 - \frac{\ell}{q}$ and $s = \frac{d}{L}$. Suppose that $s < r$. Then,*

$$\binom{q}{rq}\binom{L}{sL}\binom{(1-s)L}{\underbrace{\frac{(1-s)L}{(1-r)q}, \dots \frac{(1-s)L}{(1-r)q}}_{\ell}}\binom{sL}{\underbrace{\frac{sL}{rq}, \dots \frac{sL}{rq}}_{q-\ell}} \leq$$

$$|D_{d,\ell,L}| \leq \binom{q}{rq} \cdot \left(\sum_{i=0}^{sL} \binom{L}{i} \cdot ((1-r)q)^{L-i} \cdot (rq)^i \right).$$

Using Stirling's approximation, Lemma 34 immediately yields the following.

▶ **Corollary 35.** *In the setting of Lemma 34, suppose that $s < r$. Then,*

$$\log_q |D_{d,\ell,L}| = L(1 - D_{\mathrm{KL}_q}(s \parallel r)) \pm O(q \log L),$$

where the underlying constant is universal.

In order to compute $\beta(p, \ell, L)$, we will make use of the following characterization of $\bar{T}_{p,\ell,L}$ (Definition 31). Intuitively, this lemma says that a histogram-type τ is bad for (p, ℓ, L)-list-recovery if and only if it has many symbols inside the most frequent ℓ symbols in expectation.

▶ **Lemma 36.** *Let $1 \leq \ell \leq q$, $L \in \mathbb{N}$ and $0 \leq p \leq 1$. Let τ be a distribution over Σ^L and suppose that τ is a histogram-type. Then, $\tau \in \bar{T}_{p,\ell,L}$ if and only if*

$$\mathop{\mathbb{E}}_{u \sim \tau}[P_\ell(u)] \leq pL.$$

Now, we come to our estimate on the threshold rate for (p, ℓ, L) list-recovery in the regime where $L \to \infty$ and $q \leq o(\frac{\log L}{L})$. We begin with the following proposition, which bounds the quantity $\beta(p, \ell, L)$.

▶ **Proposition 37.** *Let $r = 1 - \frac{\ell}{q}$ and suppose that $p \leq r$. Then,*

$$\beta(p, \ell, L) = L(1 - D_{\mathrm{KL}_q}(p \parallel r)) \pm O(q \log L).$$

▶ **Corollary 38.** *The threshold rate for (p, ℓ, L) list-recovery of a random code is*

$$R^* = \begin{cases} D_{\mathrm{KL}_q}(p \parallel r) \pm O\left(\frac{q \log L}{L}\right) & \text{if } p < r \\ 0 & \text{if } p \geq r, \end{cases}$$

where $r = 1 - \frac{\ell}{q}$.

▶ Remark 39. To make sense of the threshold in Corollary 38, one can verify the identity

$$D_{\mathrm{KL}_q}(p \,\|\, 1-\ell/q) = 1 - p\log_q\left(\frac{q-\ell}{p}\right) - (1-p)\log_q\left(\frac{\ell}{1-p}\right).$$

Substituting $\ell = 1$, we find $D_{\mathrm{KL}_q}(p \,\|\, 1-1/q) = 1 - h_q(p)$, agreeing with the list decoding capacity theorem. For larger ℓ, this expression agrees with the *list-recovery capacity theorem*, as stated in e.g. [14].

5 Zero-error list-recovery and perfect hashing codes

In this section we analyze the threshold rate for zero-error list-recovery (that is, when $p = 0$), and give a more precise version of Corollary 38 in this setting.

▶ **Lemma 40.** *Let $p^* = |D_{0,\ell,L}|/q^L$. The threshold rate for $(0,\ell,L)$ list-recovery of a random code is*

$$R^* = \frac{-\log_q(p^*)}{L}.$$

We use this to compute the threshold rate for a random code be to a perfect hash code, which is the same as being $(0, q-1, q)$ list-recoverable.

▶ **Corollary 41.** *The threshold rate for $(0, q-1, q)$ list-recovery of a random code is*

$$R^* = \frac{1}{q}\log_q\left(\frac{1}{1-q!/q^q}\right).$$

The corollary follows from the lemma in a straightforward manner by verifying that $|D_{0,q-1,q}| = q^q - q!$.

6 List of two decoding of random and random linear codes

In this section, we study the list-of-2 decodability of two random ensembles of codes. In detail, we precisely compute the threshold rate for $(p,3)$-list-decoding for random codes and for random *linear* codes. Denote by $\mathcal{P}$ the monotone increasing property of *not* being $(p,3)$-list-decodable. Note that we cannot immediately apply Corollary 38, as the error term of $O\left(\frac{q\log L}{L}\right)$ is not negligible in this regime. We specialize to the case of $q = 2$, and recall our convention that log denotes the base-2 logarithm. Recall from the introduction that whenever $p < 1/4$ there exist $(p,3)$-list-decodable codes with positive rate, but whenever $p > 1/4$ the only $(p,3)$-list-decodable codes are of bounded size, independent of n.

Our main result of this section is a demonstration that the list-of-2 decoding threshold rate for random *linear* codes is in fact greater than the corresponding threshold rate for random codes. This result demonstrates that our techniques are precise enough to allow us to sharply delineate between different natural ensembles of codes.

In the following, $C^n_{\mathrm{RLC}}(R)$ denotes a random linear code of block length n and rate R. We define the threshold rate for random linear codes in a manner analogous to the definition for random codes:

$$R^n_{\mathrm{RLC}}(\mathcal{P}) := \begin{cases} \sup\left\{R \in [0,1] \,:\, \Pr\left[C^n_{\mathrm{RLC}}(R) \text{ satisfies } \mathcal{P}\right] \le \frac{1}{2}\right\} & \text{if there is such an } R \\ 0 & \text{otherwise.} \end{cases}$$

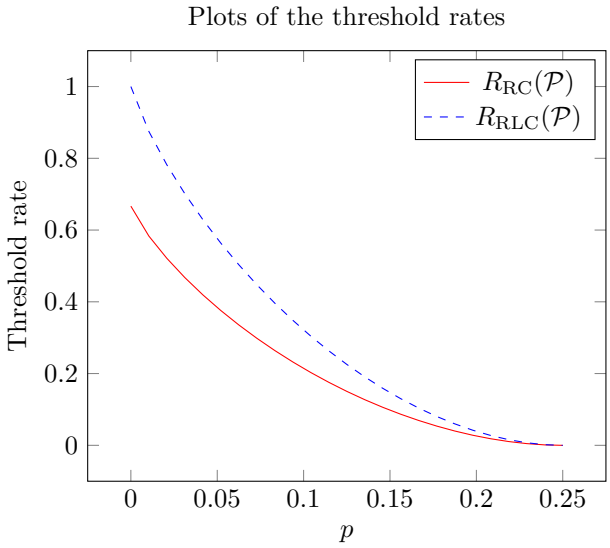

Figure 1 The threshold rate R_{RC} (red) for $(p, 3)$-list-decodability of random codes, and the threshold rate R_{RLC} (blue, dashed) for $(p, 3)$-list-decodability of random *linear* codes. Note that, uniformly over p, random linear codes have the greater threshold rate.

▶ **Theorem 42.** *Let $p \in (0, 1/4)$.*

1. *The threshold rate for $(p, 3)$-list-decoding for random codes satisfies*

$$\lim_{n\to\infty} R^n_{\mathrm{RC}}(\mathcal{P}) = 1 - \frac{1 + h(3p) + 3p\log 3}{3}.$$

2. *The threshold rate for $(p, 3)$-list-decoding for random* linear *codes satisfies*

$$\lim_{n\to\infty} R^n_{\mathrm{RLC}}(\mathcal{P}) = 1 - \frac{h(3p) + 3p\log 3}{2}.$$

Note that the threshold rate for random linear codes is greater than the threshold rate for random codes, uniformly over $p \in (0, 1/4)$. See Figure 1.

7 Computing the threshold rate for list-recovery efficiently

In the previous sections, we gave precise analytical expressions for the threshold rate for list-recovery in certain parameter regimes. However, there are some regimes where these bounds aren't precise. In this section, we consider the question of computing the threshold rate R^* algorithmically, given p, ℓ and L. We use tools from the study of entropy-maximizing distributions to develop a simple binary-search-based procedure to pinpoint R^* up to arbitrarily small additive error.

We begin with a lemma that shows that we can compute the cardinality $|D_{d,\ell,L}|$ efficiently; we will use this as a subroutine in our final algorithm.

▶ **Lemma 43.** *Given $0 \le d \le L$ and $1 \le \ell \le q$, the cardinality $|D_{d,\ell,L}|$ can be computed in time*

$$O\left((L+1)^q + \mathrm{poly}(q, L)\right).$$

We recall the following standard facts from the theory of entropy-maximizing distributions.

▶ **Lemma 44** ([17, Sec. 3]). *Let Ω be a finite nonempty set, $f : \Omega \to \mathbb{R}$ and $t \in \mathbb{R}$. Let S_t denote the set of all distributions τ over Ω such that $\mathbb{E}_{\omega\sim\tau}[f(\omega)] = t$. Let*

$$F(t) = \max_{\tau\in S_t} H(\tau).$$

Then

$$F(t) = \inf_{\alpha\in\mathbb{R}} \left[\log_q \left(\sum_{\omega\in\Omega} q^{\alpha\cdot f(\omega)}\right) - \alpha t\right].$$

Furthermore:

1. *If τ is the entropy maximizing distribution, then $\tau(\omega) = \tau(\omega')$ for every $\omega, \omega' \in \Omega$ such that $f(\omega) = f(\omega')$.*
2. *Let $t^* = \mathbb{E}_{\omega\sim\mathrm{Uniform}(\Omega)}[f(\omega)]$. Then, $F(t^*) = \log|\Omega|$, and $F(t)$ is nondecreasing (resp. nonincreasing) in the range $t < t^*$ (resp. $t > t^*$).*
3. *The function*

$$\log_q \left(\sum_{\omega\in\Omega} q^{\alpha\cdot f(\omega)}\right) - \alpha t$$

is convex in α.

▶ **Lemma 45.** *Let $\ell \le q$, $L \in \mathbb{N}$ and $0 < p \le 1$, and let $t^* = q^{-L}\cdot\sum_{d=0}^{L} d\cdot|D_{d,\ell,L}|$. Then,*

$$\beta(p,\ell,L) = \begin{cases} \inf_{\alpha\in\mathbb{R}}\left[\log_q\left(\sum_{d=0}^{L}|D_{d,\ell,L}|\cdot q^{\alpha d}\right) - \alpha pL\right] & \text{if } pL < t^* \\ L & \text{if } pL \ge t^*. \end{cases}$$

▶ Remark 46. In general, $\frac{t^*}{L}$ is slightly smaller than $1 - \frac{\ell}{q}$. Thus, Lemma 45 extends the range in which the threshold is 0 from $\left[1-\frac{\ell}{q}, 1\right]$ (Corollary 38) to $[t^*, 1]$.

▶ **Corollary 47.** *There is an algorithm, that, given p, ℓ, L and $\varepsilon > 0$, computes the threshold-rate for (p,ℓ,L)-list-recovery, within an additive error of ε. The runtime of this algorithm is $O\left((L+1)^q + \mathrm{poly}(q, L, \log\frac{1}{\varepsilon}, \beta(p))\right)$, where*

$$\beta(p) = \begin{cases} \log\frac{1}{p} & \text{if } p > 0 \\ 1 & \text{if } p = 0. \end{cases}$$

References

1. Béla Bollobás. *Random Graphs, Second Edition*, volume 73 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 2001. doi:10.1017/CBO9780511814068.
2. Imre Csiszár and Paul C Shields. *Information theory and statistics: A tutorial*. Now Publishers Inc, 2004.
3. P. Erdős and A. Rényi. On random graphs. I. *Publ. Math. Debrecen*, 6:290–297, 1959.
4. Keith Frankston, Jeff Kahn, Bhargav Narayanan, and Jinyoung Park. Thresholds versus fractional expectation-thresholds. *arXiv preprint*, 2019. arXiv:1910.13433.
5. Michael L Fredman and János Komlós. On the size of separating systems and families of perfect hash functions. *SIAM Journal on Algebraic Discrete Methods*, 5(1):61–68, 1984.
6. Ehud Friedgut. Sharp thresholds of graph properties, and the k-sat problem. *J. Amer. Math. Soc.*, 12(4):1017–1054, 1999. With an appendix by Jean Bourgain. doi:10.1090/S0894-0347-99-00305-7.

7 Venkatesan Guruswami, Johan Håstad, and Swastik Kopparty. On the list-decodability of random linear codes. *IEEE Trans. Information Theory*, 57(2):718–725, 2011. doi:10.1109/TIT.2010.2095170.

8 Venkatesan Guruswami, Ray Li, Jonathan Mosheiff, Nicolas Resch, Shashwat Silas, and Mary Wootters. Bounds for list-decoding and list-recovery of random linear codes. *arXiv preprint*, 2020. arXiv:2004.13247.

9 Venkatesan Guruswami and Andrii Riazanov. Beating fredman-komlós for perfect k-hashing. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

10 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.

11 J Korner and Katalin Marton. New bounds for perfect hashing via information theory. *European Journal of Combinatorics*, 9(6):523–530, 1988.

12 Jénos Körner. Fredman–komlós bounds and information theory. *SIAM Journal on Algebraic Discrete Methods*, 7(4):560–570, 1986.

13 Jonathan Mosheiff, Nicolas Resch, Noga Ron-Zewi, Shashwat Silas, and Mary Wootters. Ldpc codes achieve list decoding capacity. *arXiv preprint*, 2019. arXiv:1909.06430.

14 Nicolas Resch. *List-Decodable Codes:(Randomized) Constructions and Applications*. PhD thesis, Carnegie Mellon University, 2020.

15 Atri Rudra and Mary Wootters. Average-radius list-recovery of random linear codes. In *Proceedings of the 2018 ACM-SIAM Symposium on Discrete Algorithms, SODA*, 2018.

16 Salil P. Vadhan. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science*, 7(1-3):1–336, 2012. doi:10.1561/0400000010.

17 Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008. doi:10.1561/2200000001.

18 Chaoping Xing and Chen Yuan. Beating the probabilistic lower bound on perfect hashing. *arXiv preprint*, 2019. arXiv:1908.08792.

19 Victor Vasilievich Zyablov and Mark Semenovich Pinsker. List concatenated decoding. *Problemy Peredachi Informatsii*, 17(4):29–33, 1981.