An Improved Johnson Bound for List-Decoding with Hints

Swastik Kopparty [*] Harry Sha [†]

Abstract

This paper studies when list-decodability guarantees for error-correcting codes of good distance can go beyond the classical Johnson radius and gives a corresponding efficient list-decoding algorithm for Reed-Solomon codes up to this new radius. The Johnson bound shows that any error-correcting code with distance δ is automatically list-decodable with small list size up to a certain radius $J(\delta)$ around any received word r. We show that if the received word r is known to have some codewords that are somewhat close to it (we call these "hints"), then C is list-decodable with small list size up to an even larger radius around r.

We determine the optimal list-decoding radius in this setting for single and multiple hints over both large and small alphabets and show that this radius is the optimum of a natural convex program. Remarkably, even a single hint that is $(J(\delta) - \epsilon)$-close to r improves the list-decoding radius around r to $J(\delta) + \epsilon'$.

Finally, we give an efficient algorithm, based on the weighted Guruswami-Sudan decoder, that list-decodes Reed-Solomon codes up to this optimal radius.

1 Introduction

This paper studies when list-decodability guarantees for error-correcting codes of good distance can go beyond the classical Johnson radius and gives a corresponding efficient list-decoding algorithm for Reed-Solomon codes up to this new radius. Very briefly, the Johnson bound shows that for any code and any received word r, the ball around r of a certain radius contains very few codewords. We show that if the received word r is known to have some codewords that are somewhat close to it, then a ball of an even larger radius around r contains very few codewords. We give a sharp characterization of this radius and use our understanding to give *efficient* algorithms for list-decoding Reed-Solomon codes up to this larger radius[*]

To describe things in more detail, we quickly introduce some notation. For an alphabet Σ and large n, we work with the metric space Σ^n equipped with the relative Hamming distance, where $\Delta(x, y)$ is the fraction of coordinates $i \in [n]$ for which $x_i \neq y_i$. An error-correcting code of (relative) distance δ over the alphabet Σ is a subset $C \subseteq \Sigma^n$ such that any two elements of C have distance at least δ. By the triangle inequality, such a code immediately satisfies unique decodability up to

[*]Department of Mathematics and Department of Computer Science, University of Toronto. Email: swastik.kopparty@utoronto.ca. Research partially supported by an NSERC Discovery Grant.

[†]Department of Computer Science, University of Toronto. Email: shaharry@cs.toronto.edu. Research partially supported by an NSERC Discovery Grant.

[*]This paper was written before the algorithmic breakthroughs on list-decoding Reed-Solomon codes over prime fields [Bra+26], see also [HKS26]. We remark that our algorithmic results for Reed-Solomon codes are in the regime of *constant* list size bounds for decoding beyond the Johnson radius around certain centers, while [Bra+26] gives a polynomial list size bound around all centers.

radius $\frac{\delta}{2}$: for any string $r \in \Sigma^n$, there can be at most one element of C that is less than $\delta/2$-close to r. This lets us use C to correct $\delta/2$ errors.

To correct even more errors, we have to relax to the notion of *list-decoding*, introduced by Elias in [Eli57]. Given a received word r, the goal of list-decoding is to find all codewords within a given distance ρ of r while ensuring that the number of such codewords is small. The Johnson bound [Joh62] gives a generic nontrivial combinatorial list-decodability result: for any code C of distance δ, the number of codewords within radius $\rho = 1 - \sqrt{1-\delta}$ of any received word is at most polynomial in the code length. Remarkably, when δ is a constant sufficiently close to 1, the radius ρ can get arbitrarily close to 1!

In this generality, the bound is tight: if ρ exceeds this threshold by any constant, then there are codes with exponentially many codewords within radius ρ of some received word. The codes witnessing this are, as far as we know, nonlinear. For linear codes, things are much trickier, but it is still known that there are codes and received words with superpolynomially many codewords at this radius [Gur02; GS02a].

Progress on the algorithmic side of list-decoding began with the classical univariate polynomial evaluation codes, Reed-Solomon codes. The deep and fundamental Guruswami-Sudan algorithm [Sud97; GS99] showed that for Reed-Solomon codes, it is possible to efficiently list-decode (i.e., find all the ρ-close codewords) up to the Johnson radius $\rho = 1 - \sqrt{1-\delta}$. This result (and the underlying algorithm) had a profound impact on coding theory, pseudorandomness, and complexity theory in the following decades, leading to state-of-the-art constructions of list-decodable codes [PV05; GR08; Kop+23; Sri25; CZ25], extractors [SU05; GUV09; Dvi+13; TU12], and pseudorandom generators [STV01; SU05; Uma03], among others.

Despite all this impact, the list-decodability of Reed-Solomon codes itself remains shrouded in mystery. It is known that some Reed-Solomon codes do need list sizes polynomially large in n when decoding slightly beyond the Johnson radius [BKR10]. It is also known that Reed-Solomon codes with random evaluation points are list-decodable up to any radius $\rho < \delta$ with constant list size [BGM23; GZ23; AGL24]. It remains a tantalizing possibility that every Reed-Solomon code can be list-decoded up to any radius $\rho < \delta$ with $\mathrm{poly}(n)$ list size, and furthermore, that this list can be found efficiently[†].

Given this state of affairs, it is of great interest to find new methods that *prove* or *certify* that for a given $r \in \Sigma^n$ and radius ρ, the number of codewords within distance ρ of r is small. There are two known methods for this. The first is the Johnson bound, which applies uniformly to all r and has ρ equal to the Johnson radius $J(\delta)$. The second is the triangle inequality: if we have a codeword $h \in C$ with $\Delta(r, h) < \delta - \rho$, then there are no other codewords within distance ρ of r.

In this paper, we give a new method (that applies to all codes) to *prove* that the list of codewords within a radius $\rho \geq J(\delta)$ of a given received word r is small. This method is based on codewords $h \in C$ that are somewhat close to r. We call such codewords *hints*. The more hints we know about and the closer they are to r, the larger the radius ρ will be. We then show tightness of this improved radius ρ and give an efficient algorithm to list-decode Reed-Solomon codes up to this radius ρ.

We summarize our main results below:

- For general codes C, for every $\epsilon > 0$, there is an $\epsilon' > 0$ such that if there exists even one $h \in C$ with $\Delta(h, r) < J(\delta) - \epsilon$, then list-decoding around r to radius $J(\delta) + \epsilon'$ produces only

[†]We now know, from [Bra+26] (see also [HKS26]), that this is true over prime fields.

constant-size lists.

- For general codes C, given a collection of $h_1, \ldots, h_m \in C$ with $\Delta(h_i, r) < J(\delta)$, we show how to compute a radius ρ depending on the relative geometry of the h_i and r so that list-decoding C to radius ρ around r produces only constant-size lists.
- Conversely, we show that the radius ρ computed above is tight in general: for any desired relative geometry that is realizable, there are codes C, received words r, and codewords $h_1, \ldots, h_m$ realizing this relative geometry for which list decoding beyond the radius ρ produces lists of size exponential in n.
- The above results generalize the large-alphabet (a.k.a. alphabet-independent) Johnson bound. We also prove small-alphabet versions of these results, generalizing the small-alphabet Johnson bound.
- Finall, we give efficient algorithms for list-decoding Reed-Solomon codes up to this radius ρ (which, because of the combinatorial results above, results in a constant-size list).

The closest point of comparison is the *average radius* Johnson bound. This is a folklore generalization of the Johnson bound that states that for any code C with distance δ, for any $r \in \Sigma^n$, and for any L codewords $c_1, \ldots, c_L \in C$, their average distance to r must be at least the Johnson radius:

$$\frac{1}{L} \sum_{i=1}^{L} \Delta(c_i, r) \geq J\left(\delta - \frac{\delta}{L}\right) = J(\delta) - o_L(1).$$

It is completely consistent with the average radius Johnson bound that for some constant $\epsilon > 0$, one codeword has distance $J(\delta) - \epsilon$ to r, while there are exponentially many codewords with distance exactly $J(\delta)$ to r.

In contrast, the simplest instantiation of our improved list-decoding bound rules this out. Even one codeword with distance $J(\delta) - \epsilon$ to r forces the number of codewords within distance $J(\delta) + \epsilon'$ of r (where $\epsilon' > 0$ is a constant depending on ϵ and δ) to be bounded by a constant.

For this simple setting, our algorithmic results for Reed-Solomon codes have the following curious consequence. Suppose we are given a received word r for a Reed-Solomon code with distance $\delta = 0.99$. The Johnson radius for this setting is

$$J(\delta) = 1 - \sqrt{1 - 0.99} = 0.9,$$

and thus the Guruswami-Sudan list-decoder can find any codeword within this radius of r. Now suppose we additionally know that there is some codeword h of the Reed-Solomon code with $\Delta(h, r) = 0.85$ – so h only agrees with r on 15% of the coordinates. This fact enables us to efficiently list-decode around r to an even larger radius, $\rho = 0.901$!

We would like to mention one final motivation for results of this type. These results give new and nontrivial information about the set of realizable Hamming distances between codewords of codes $C \subseteq \Sigma^n$ and general elements of Σ^n. These kinds of questions arise naturally in (1) understanding basic questions about rates, distances, and distance distributions of error-correcting codes (as in the paper of Kalai and Linial [KL95]), and (2) decoding concatenated codes using soft information (as in the paper of Guruswami and Sudan [GS02b]). We believe our results and methods could be helpful in the study of these topics.

1.1 Results

It is most convenient to state our results in terms of agreements rather than distance. For $x, y \in \Sigma^n$, we define $\text{agree}(x, y)$ to be the fraction of coordinates $i \in [n]$ such that $x_i = y_i$. We then work with the *maximum agreement* α of a code C rather than the minimum distance δ ($\alpha = 1 - \delta$).

We say that a code $C \subseteq \Sigma^n$ is (β, L)*-list-decodable at* $r \in \Sigma^n$ if the number of codewords $c \in C$ with $\text{agree}(c, r) \geq \beta$ is at most L.

In these terms, the Johnson bound says that any code $C \subseteq \Sigma^n$ with maximum agreement α is $(\sqrt{\alpha}, O(n))$-list-decodable at *every* $r \in \Sigma^n$. The main object of this paper is to understand how much the list-decodability of a code C at a certain r improves (over the Johnson bound) if we know that there are some "hint" codewords $h_1, \ldots, h_m \in C$ that are somewhat close to r.

The relationship between the hints and the received word is captured by the *agreement profile* η. For a single hint, η is simply the agreement between the hint and the received word; for multiple hints, η encodes the full pattern of agreements among the hints and the received word.

Our main (and most general) result is a tight characterization of the agreement β for which we are guaranteed list-decodability with small list size at r: it is expressed as the optimum of a natural convex program whose constraints are determined by η and code parameters such as the maximum agreement and the alphabet size.

Theorem 1. (Theorem 6.4, restated) *Let $C \subseteq \Sigma^n$ be a code of maximum agreement α (minimum distance $1 - \alpha$), where $q := |\Sigma|$, and assume $\alpha > 1/q$. Let $r \in \Sigma^n$ be a received word, and let $h_1, \ldots, h_m \in C$ be hints, with agreement profile η.*

Let β^* be the optimum of the convex program Optimization 6.1.

Then for any $\beta > \beta^*$, there is a constant $L = L(\beta, \beta^*)$ such that C is (β, L)-list-decodable at r.

Conversely, for any $\beta < \beta^*$ and any realizable profile η, there exists a code $C \subseteq \Sigma^n$, a received word $r \in \Sigma^n$, and hints $h_1, \ldots, h_m \in C$ with agreement profile η such that C is **not** *$(\beta, \exp(\Omega(n)))$-list-decodable at r.*

Note that when the number of hints m is constant, Optimization 6.1 is a convex program in a constant number of variables with a constant number of constraints.

Optimization 6.1 can be complicated in general, but simplifies considerably in natural special cases. We study several natural special cases: a single hint over a large alphabet (Optimization 3.1), a single hint over a small alphabet (Optimization 4.1), and multiple hints over a large alphabet (Optimization 5.1). For a single hint over a large alphabet, we solve the optimization analytically. The resulting threshold β^* is strictly less than the Johnson agreement $\sqrt{\alpha}$ whenever the hint's agreement with the received word exceeds the Johnson agreement $\sqrt{\alpha}$.

Theorem 2. (Theorem 3.3, restated) *Let $C \subseteq \Sigma^n$ be a code with maximum agreement α over a large alphabet, let $r \in \Sigma^n$ be a received word, and let $h \in C$ be a hint with* $\text{agree}(r, h) = \eta \in [0, 1]$.

Define $\beta^*(\alpha, \eta)$ by

$$\beta^* = \begin{cases} \sqrt{\alpha}, & \eta \leq \sqrt{\alpha}, \\ \alpha + \sqrt{(1-\eta)\left(\alpha - \frac{\alpha^2}{\eta}\right)}, & \sqrt{\alpha} < \eta \leq \eta_{\text{crit}}, \\ \alpha + 1 - \eta, & \eta > \eta_{\text{crit}}, \end{cases}$$

where $\eta_{\text{crit}} = \frac{(1-\alpha)+\sqrt{1-2\alpha+5\alpha^2}}{2}$. *(In particular, observe that* $\beta^* < \sqrt{\alpha}$ *whenever* $\eta > \sqrt{\alpha}$*.)*

Then for any $\beta > \beta^*$*, there is a constant* $L = L(\beta, \beta^*)$ *such that* C *is* (β, L)*-list-decodable at* r*.*

Conversely, for any $\beta < \beta^*$ *and* $\eta \in [0, 1]$*, there exists a code* $C \subseteq \Sigma^n$*, a received word* $r \in \Sigma^n$*, and a hint* $h \in C$ *with* $\text{agree}(r, h) = \eta$ *such that* C *is* **not** $(\beta, \exp(\Omega(n)))$*-list-decodable at* r*.*

These combinatorial results were stated above for list-decodability. We remark that they also hold for the slightly stronger notion of average-list-decodability (see Definition 2.6).

Finally, we show that these combinatorial bounds can be made algorithmic for Reed-Solomon codes, using the weighted Guruswami-Sudan list decoder.

Theorem 3. (Theorem 7.2, restated) *Suppose* C *is a Reed-Solomon code; let* $h_1, \ldots, h_m \in C$ *be hints, and let* r *be a received word. Let* β^* *be the list-decoding threshold from Theorem 1. Then for any constant* $\beta > \beta^*$*, there exists a polynomial-time algorithm that finds all codewords* $c \in C$ *with* $\text{agree}(c, r) \geq \beta$*.*

Remark: Chronologically, we first discovered our algorithmic result for Reed-Solomon codes with one hint. Specifically, we asked whether it was possible to use the fact that there is one codeword close to our given received word to improve the guarantees of the Guruswami-Sudan list-decoding algorithm. After we managed to answer this, further investigation led us to discover that the underlying improved combinatorial bound actually holds for all error-correcting codes. The initial modification to the Guruswami–Sudan list-decoding algorithm that utilized a nearby codeword did not use weights and instead multiplied the interpolant by the nearby polynomial itself. We discuss this approach in Appendix A.

1.2 Proof Overview

We give a high-level overview of the three parts of our argument: the upper bound on list size, the matching lower bound, and the efficient decoding algorithm for Reed-Solomon codes.

1.2.1 Upper bounds on list size

We start with the single-hint, large-alphabet case, which contains the main ideas in their simplest form, and then explain how the argument extends to the general setting.

Single hint, large alphabet. Let $L \subseteq C \setminus \{h\}$ be a list of codewords with agreement at least β with the received word r. Let $p_{i,\sigma} \coloneqq \Pr_{c \sim L}[c_i = \sigma]$ be the fraction of list elements that have symbol σ at index i.

Since all distinct codewords have pairwise agreement at most α, the expected pairwise agreement of two random elements of L is at most $\alpha + \frac{1}{|L|}$.

On the other hand, the expected pairwise agreement of two random elements of L equals

$$\frac{1}{n} \sum_{i,\sigma} p_{i,\sigma}^2,$$

and thus

$$\frac{1}{n} \sum_{i,\sigma} p_{i,\sigma}^2 \;\leq\; \alpha + \frac{1}{|L|}. \tag{1}$$

We now use the fact that the elements of L all have high agreement with r to lower bound the left-hand side of the above inequality. Without a hint, the sum $\frac{1}{n}\sum_i p_{i,r_i}$ is at least β, and by the Cauchy-Schwarz inequality, $\frac{1}{n}\sum_i p_{i,r_i}^2 \geq \beta^2$. Combined with (1), this gives $\beta^2 \leq \alpha + 1/|L|$, recovering the Johnson bound $\beta^* = \sqrt{\alpha}$.

Crucially, the Cauchy-Schwarz lower bound is tight when all of the p_{i,r_i} are equal to β, i.e., when the list's agreement with r is spread uniformly across coordinates. The introduction of a hint can prevent this uniform distribution from being feasible, leading to a stronger bound. Suppose we have a hint h with $\text{agree}(r,h) = \eta$. We partition the coordinates into U (where r and h agree, $|U| = \eta n$) and $V = [n] \setminus U$. Define macroscopic variables $a = \frac{1}{n}\sum_{i\in U} p_{i,r_i}$ (the average agreement coming from U) and $b = \frac{1}{n}\sum_{i\in V} p_{i,r_i}$ (the average agreement coming from V), so that $a + b \geq \beta$. Since on U we have $r_i = h_i$, the quantity a also lower bounds the list's average agreement with h. Since h is an element of the code, this gives the capacity constraint $a \leq \alpha$. Applying Cauchy-Schwarz separately to U and V yields

$$\frac{a^2}{\eta} + \frac{b^2}{1-\eta} \leq \alpha + 1/|L|. \tag{2}$$

When $\eta > \sqrt{\alpha}$, the capacity constraint $a \leq \alpha$ prevents the variables from achieving the unconstrained optimum, forcing an uneven split between U and V that increases the quadratic sum. This gives a list-decoding threshold $\beta^* < \sqrt{\alpha}$, defined as the maximum of $a + b$ subject to the quadratic and capacity constraints (Optimization 3.1).

To get the quantitative bound $|L| = O(1/\epsilon)$ when $\beta = \beta^* + \epsilon$, we need to show the LHS of (2) is $\alpha + \Omega(\epsilon)$. This can be done by hand (e.g., by calculating gradients at the optimum) for the current single-hint, large-alphabet case. In the more general cases, however, this can be done using Lagrangian duality, which gives tools for understanding how the optimum of a convex optimization changes under perturbations of the constraints. Specifically, we use the relaxed Slater condition to show that strong duality holds, allowing us to work with the Lagrangian and perform a first-order analysis of it. We do this in Lemma 6.2 (which shows that the optimization satisfies the relaxed Slater condition) and Lemma 6.3 (which does the first-order analysis).

The other cases follow the same template with several modifications.

Addressing smaller alphabets. Let's again start by recovering the standard small-alphabet Johnson bound (without a hint). Recall that in the large-alphabet case, we lower bounded the sum in (1) by ignoring all the $p_{i,\sigma}$ for $\sigma \neq r_i$ and applying Cauchy-Schwarz. When the alphabet size q is small, the terms that we ignored are no longer negligible. Write β for the average agreement $\frac{1}{n}\sum_i p_{i,r_i}$. Applying Cauchy-Schwarz gives $\frac{1}{n}\sum_i p_{i,r_i}^2 \geq \beta^2$. For the non-received symbols, the total mass is $\frac{1}{n}\sum_{i,\sigma\neq r_i} p_{i,\sigma} = 1 - \beta$, so Cauchy-Schwarz gives

$$\frac{1}{n}\sum_i \sum_{\sigma\neq r_i} p_{i,\sigma}^2 \geq \frac{(1-\beta)^2}{q-1}.$$

Thus,

$$\frac{1}{n}\sum_{i,\sigma} p_{i,\sigma}^2 \;\geq\; \beta^2 + \frac{(1-\beta)^2}{q-1}.$$

Substituting this lower bound into (1) and solving for β recovers the standard small-alphabet Johnson bound.

When a hint is present, we partition the sum $\sum_{i,\sigma} p_{i,\sigma}^2$ more finely. Within U (coordinates where $r_i = h_i$), there are two groups: $\sigma = r_i$ and $\sigma \neq r_i$. Within V (coordinates where $r_i \neq h_i$), there are three groups: $\sigma = r_i$, $\sigma = h_i$, and σ equal to neither. As in the large-alphabet case, we introduce macroscopic variables corresponding to this partition and derive constraints based on the agreement between list elements (the ellipsoid constraint) and the agreement between the hint and list elements (the capacity constraint). This yields Optimization 4.1.

Addressing multiple hints. When there are multiple hints, $h_1, \ldots, h_m$, the hints can differ not only in their agreement with the received word but also in their agreement with one another. To account for this, we take a finer partition of coordinates. Each coordinate i is classified by the full pattern of symbol equalities among the received word and all hints, forming a partition π_i of $\{0, 1, \ldots, m\}$ (where 0 represents r). Coordinates with the same partition type are grouped together, and we define macroscopic variables $a_{\pi,S}$ tracking the normalized sum of $p_{i,\sigma}$ for each part S of each partition π, along with a residual variable b_π for the remaining symbols. Applying Cauchy-Schwarz within each group gives an ellipsoid bound similar to (2) where the LHS is a weighted sum of squares of the variables and the RHS is $\alpha + 1/|L|$ (the ellipsoid constraint), while the hint agreements impose linear capacity constraints. This yields Optimization 6.1, whose optimum β^* is the list-decoding threshold.

1.2.2 Lower bounds on list size

To show tightness, we prove that for any $\beta = \beta^* - \epsilon$ there exist codes with exponentially many codewords all having agreement at least β with r.

First, recall the standard proof of the tightness of the Johnson agreement $\sqrt{\alpha}$ (without hints). We take the received word r to be the all-zeros word. Now we create a code with maximum agreement α and with exponentially many codewords having agreement $\sqrt{\alpha} - \epsilon$ with r. This is done by the probabilistic method: the codewords are chosen independently, with a random $(\sqrt{\alpha}-\epsilon)$-fraction of coordinates set to 0 and the remaining coordinates chosen independently and uniformly from Σ. Their agreement with r is guaranteed by construction, and their pairwise maximum agreement follows from exponentially good concentration bounds (and a union bound over all pairs of codewords).

We now describe the construction for the single-hint, large-alphabet case; the general case (Section 6) follows the same template with a finer partition of coordinates.

As above, we will take r to be the all-zeros word. We have to create a code C and a hint $h \in C$. The hint h is chosen to be 0 (and hence agree with r) on the first ηn coordinates and to be 1 on the remaining coordinates. Partition $[n]$ into blocks U (where r and h agree) and V (where they disagree).

We now create the rest of the code C with exponentially many codewords close to r. We have to balance this with the fact that (1) all the new codewords should have agreement at most α with each other and (2) all the new codewords have to have agreement at most α with h. To do this, we again use the probabilistic method, this time guided by an optimal solution (a^*, b^*) to Optimization 3.1.

After a mild scaling, consider the feasible solution $(a, b) = (1 - \epsilon/\beta^*)(a^*, b^*)$ with objective $a + b = \beta = \beta^* - \epsilon$ and strict slack $\delta = \Theta(\epsilon)$ in the ellipsoid constraint.

We now take C to contain exponentially many independently picked codewords c, uniformly at

random subject to having an 0s among the U coordinates and bn 0s among the V coordinates. (Thus, the remaining coordinates are picked uniformly at random from $\Sigma \setminus \{0,1\}$.) The pairwise agreement of two codewords picked from this distribution on the U coordinates and the V coordinates will be $(a/\eta)^2 + \delta/2$ and $(b/(1-\eta))^2 + \delta/2$, respectively, with high probability, and thus the total agreement will be

$$a^2/\eta + b^2/(1-\eta) + \delta/2 \leq \alpha - \delta/2 < \alpha,$$

where the first inequality follows from the fact that (a, b) is feasible for Optimization 3.1. (Note that in the large-alphabet setting, the collisions on symbols not equal to 0 are negligible and can be rolled into the δ term.)

Furthermore, notice that the agreement between h and a codeword picked from this distribution is exactly a, which is at most α since a is feasible. Finally, every codeword in $C \setminus \{h\}$ has exactly $an + bn = \beta n$ 0s and hence agreement exactly β with r.

1.2.3 Decoding algorithm for RS codes

Since Reed-Solomon codes have $q \geq n$, they fall into the large-alphabet regime. We are given a received word r and m hints $h_1, \ldots, h_m \in C$. In this regime, we partition the coordinates based on which subset $S \subseteq [m]$ of hints agree with r at that coordinate: U_S is the set of coordinates i where the set of hints that agree with r on coordinate i is S. Let $\eta_S = |U_S|/n$, which captures the size of the part U_S.

At a high level, the algorithm for efficiently list-decoding Reed-Solomon codes from agreement β will use the weighted Guruswami-Sudan list-decoding algorithm [GS99; Gur04] with weights derived from an optimal solution $(a^*_S)_{S \subseteq [m]}$ to Optimization 5.1.

The weighted GS list-decoder takes a weight vector $w \in \mathbb{R}^n_{\geq 0}$ and returns all codewords c such that the weighted agreement between c and r exceeds the threshold $\sqrt{\alpha n \sum_{i=1}^n w_i^2} + \epsilon w_{\max}$.

We set the weight for all the coordinates $i \in U_S$ to be the same:

$$w_i = a^*_S/\eta_S.$$

With this choice of weights (and using the feasibility of the a^*_S), the Guruswami-Sudan decoding threshold simplifies to αn. For any codeword c with $\text{agree}(c, r) \geq \beta^* + \epsilon$, let $a_S = \frac{1}{n}|\{i \in U_S : r_i = c_i\}|$. Then the weighted agreement between c and r equals

$$n \sum_S \frac{a^*_S a_S}{\eta_S},$$

which, based on the properties of Optimization 5.1, can be shown to exceed αn. Thus the codeword c will be found by our invocation of the weighted Guruswami-Sudan decoder.

1.2.4 Organization

We begin in Section 3 with the simplest case—a single hint over a large alphabet—which contains the main ideas. The next three sections extend the analysis to small alphabets (Section 4), multiple hints over a large alphabet (Section 5), and the fully general case (Section 6). Since the first two are special cases of the general setting, we focus on the additional complications each introduces and the resulting optimization problem, deferring the full proof to Section 6. Section 7 shows how to make these bounds algorithmic for Reed-Solomon codes using the weighted Guruswami-Sudan algorithm.

AI use

The results were obtained without any AI involvement. The writing was done without any AI involvement. We used AI for computing the optimum of our optimization for assorted instantiations, and for feedback on our writeup.

Acknowledgements

We are grateful to the anonymous reviewers of FOCS 2026 and SODA 2027 for valuable comments and corrections. SK thanks John Kim for valuable discussions.

2 Preliminaries and Notation

Let Σ be an alphabet of size $q = |\Sigma|$, and let Σ^n be the set of words of length n over Σ. Throughout, n denotes the codeword length and m the number of hints; we think of n as growing and m as fixed unless otherwise stated.

For $x, y \in \Sigma^n$, define the *Hamming agreement* $\mathrm{Agree}(x, y) = |\{i \in [n] : x_i = y_i\}|$ and the *normalized agreement* $\mathrm{agree}(x, y) = \mathrm{Agree}(x, y)/n$. A code $C \subseteq \Sigma^n$ has *maximum agreement* α if $\mathrm{agree}(x, y) \leq \alpha$ for all distinct $x, y \in C$. For a subset $S \subseteq [n]$, we write $\mathrm{Agree}_S(x, y) = |\{i \in S : x_i = y_i\}|$ for the agreement restricted to coordinates in S.

We use $\circ$ to denote concatenation of strings.

2.1 Random Codes with a Certain Type

In our lower bounds, we construct codes in which each alphabet symbol occurs an exact number of times in every codeword. A *type* specifies this distribution of symbols.

Definition 2.1 (Type)**.** Let $t : \Sigma \to \mathbb{N}$ be such that $\sum_{\sigma \in \Sigma} t(\sigma) = n$. A string $x \in \Sigma^n$ has type t if for each $\sigma \in \Sigma$, the number of occurrences of σ in x is exactly $t(\sigma)$.

Exponentially large codes of a prescribed type exist; this can be proved using the method of averaged bounded differences (see, e.g., [DP09]).

Lemma 2.2. *Fix an alphabet Σ, $n \in \mathbb{N}$, a type $t : \Sigma \to \mathbb{N}$, and a constant $\epsilon > 0$ independent of n. If*

$$\alpha \geq \sum_{\sigma \in \Sigma} \left(\frac{t(\sigma)}{n}\right)^2 + \epsilon, \qquad \text{and} \qquad \ell^2 \exp(-\epsilon^2 n/2) < 1,$$

then there exists a code of size ℓ in which every codeword has type t and the maximum agreement is at most α. In particular, we can take $\ell = \exp(\Omega(n))$.

2.2 Expected agreement

We will need the following basic bound on the expected agreement between two distinct elements of a set.

Lemma 2.3. *Let $L \subseteq \Sigma^n$. Let $p_{i,\sigma} = \Pr_{x \sim L}[x_i = \sigma]$. Then*

$$\mathop{\mathbb{E}}_{x,y \sim L, x \neq y}[\mathrm{agree}(x, y)] \geq \left(\frac{1}{n} \sum_{i=1}^{n} \sum_{\sigma \in \Sigma} p_{i,\sigma}^2\right) - \frac{1}{|L|}.$$

This follows immediately from the next two claims.

Claim 2.4. *Let $L \subseteq \Sigma^n$. Let $p_{i,\sigma} = \Pr_{x\sim L}[x_i = \sigma]$. Then*

$$\mathop{\mathbb{E}}_{x,y\sim L}[\mathrm{agree}(x,y)] = \frac{1}{n}\sum_{i=1}^{n}\sum_{\sigma\in\Sigma} p_{i,\sigma}^2.$$

Proof.

$$\begin{aligned}
\mathop{\mathbb{E}}_{x,y\sim L}[\mathrm{agree}(x,y)] &= \mathop{\mathbb{E}}_{x,y\sim L, i\sim[n]} \mathbf{1}_{x_i=y_i} \\
&= \mathop{\mathbb{E}}_{x,y\sim L, i\sim[n]} \left[\sum_{\sigma\in\Sigma} \mathbf{1}_{x_i=y_i=\sigma}\right] \\
&= \mathop{\mathbb{E}}_{i\sim[n]} \sum_{\sigma\in\Sigma} \left[\mathop{\mathbb{E}}_{x,y\sim L}[\mathbf{1}_{x_i=y_i=\sigma}]\right] && \text{(Linearity)} \\
&= \mathop{\mathbb{E}}_{i\sim[n]} \sum_{\sigma\in\Sigma} \left[\Pr_{x,y\sim L}[x_i = y_i = \sigma]\right] \\
&= \mathop{\mathbb{E}}_{i\sim[n]} \sum_{\sigma\in\Sigma} \left[\Pr_{x\sim L}[x_i = \sigma]^2\right] && \text{(Independence)} \\
&= \mathop{\mathbb{E}}_{i\sim[n]} \sum_{\sigma\in\Sigma} p_{i,\sigma}^2.
\end{aligned}$$

□

Claim 2.5. *Let $L \subseteq \Sigma^n$. Then*

$$\mathop{\mathbb{E}}_{c\neq c'\sim L}[\mathrm{agree}(c,c')] \geq \mathop{\mathbb{E}}_{c,c'\sim L}[\mathrm{agree}(c,c')] - \frac{1}{|L|}.$$

Proof. Let $\ell \coloneqq |L|$. Sample (c, c') by drawing two elements from L independently and uniformly at random with replacement. Then

$$\mathop{\mathbb{E}}_{c,c'\sim L}[\mathrm{agree}(c,c')] = \Pr[c = c']\ \mathbb{E}[\mathrm{agree}(c,c') \mid c = c'] + \Pr[c \neq c']\ \mathbb{E}[\mathrm{agree}(c,c') \mid c \neq c'].$$

When $c = c'$, we have $\mathrm{agree}(c, c') = 1$. Also $\Pr[c = c'] = 1/\ell$. Denote $A_{\neq} \coloneqq \mathbb{E}[\mathrm{agree}(c,c') \mid c \neq c'] = \mathbb{E}_{c\neq c'\sim L}[\mathrm{agree}(c,c')]$. Therefore

$$\mathop{\mathbb{E}}_{c,c'\sim L}[\mathrm{agree}(c,c')] = \frac{1}{\ell}\cdot 1 + \left(1 - \frac{1}{\ell}\right) A_{\neq}.$$

Rearranging gives

$$A_{\neq} = \frac{\mathbb{E}_{c,c'\sim L}[\mathrm{agree}(c,c')] - 1/\ell}{1 - 1/\ell} \geq \mathop{\mathbb{E}}_{c,c'\sim L}[\mathrm{agree}(c,c')] - \frac{1}{\ell},$$

where the inequality uses that $1/(1 - 1/\ell) \geq 1$. This proves the claim. □

2.3 Convex Optimization and Duality

We show that the list-decoding threshold is the optimum of a convex program. Our proofs of the upper bound on list size rely heavily on several properties of such programs. See [BV04] for a reference.

Our convex programs take the following form:

$$\begin{aligned} \beta^* \;=\; \max_{x \in \mathcal{X}} \quad & c^T x \\ \text{s.t.} \quad & f(x) \le \alpha, \end{aligned}$$

where f is a convex and differentiable function, $\mathcal{X}$ is a non-empty convex set defined by linear constraints, and c is a fixed vector. The *Lagrangian* of this program, with respect to the convex constraint $f(x) \le \alpha$, is

$$\mathcal{L}(x, \lambda) \;=\; c^T x - \lambda\big(f(x) - \alpha\big),$$

where $\lambda \ge 0$ is a dual variable (or *KKT multiplier*) associated with the constraint $f(x) \le \alpha$. Weak duality asserts that $\beta^* \;\le\; \min_{\lambda \ge 0} \max_{x \in \mathcal{X}} \; \mathcal{L}(x, \lambda)$. *Strong duality* holds when this inequality is tight, i.e., when there exists $\lambda^* \ge 0$ such that

$$\beta^* \;=\; \max_{x \in \mathcal{X}} \; \mathcal{L}(x, \lambda^*).$$

When $\lambda^* > 0$, *complementary slackness* further requires that $f(x^*) = \alpha$ at any primal optimum x^*.

The relaxed Slater condition provides a convenient sufficient condition for strong duality: if there exists a point $z \in \mathcal{X}$ satisfying $f(z) < \alpha$ (i.e., a *strictly* feasible point for the convex constraint), then strong duality holds. We call this point the *Slater point*. In each of our optimization problems, we identify such a Slater point explicitly.

2.4 List decoding with a hint

We consider four settings, depending on whether one or more hints are given and whether the alphabet is large or small. In the large-alphabet setting, $q = |\Sigma|$ is assumed to be sufficiently large as a function of the other parameters. The small-alphabet setting treats q as an explicit parameter. These two regimes are analogous to the large-alphabet[‡] and small-alphabet versions of the classical Johnson bound.

The problem is governed by the following parameters:

- $\alpha \in [0, 1]$, the maximum agreement of the code C.
- η, capturing the relationship between the hints and the received word. For a single hint, $\eta = \text{agree}(h, r)$. For multiple hints, η must also capture the relationship between the hints. The definition there is slightly more involved and is given in the relevant sections.
- m, the number of hints.
- $q = |\Sigma|$, the alphabet size.

An instance of the list-decoding problem with hints is characterized by a code $C \subseteq \Sigma^n$, a received word $r \in \Sigma^n$, a collection of one or more hints, and an alphabet size q. We write P for the tuple of parameters characterizing a given setting (e.g., $P = (\alpha, \eta)$ for a single hint over a large alphabet).

[‡]The large-alphabet Johnson bound is sometimes called the alphabet-free Johnson bound.

We call an agreement profile η (q, α)-realizable if there exist a blocklength n, an alphabet Σ of size q, a received word $r \in \Sigma^n$, and hints $h_1, \ldots, h_m \in \Sigma^n$ having agreement profile η such that $\text{agree}(h_j, h_k) \leq \alpha$ for every pair of distinct hints $j, k \in [m]$. Equivalently, the hints are contained in a code of maximum agreement at most α. In the large-alphabet setting, we simply say that η is α-realizable if it is (q, α)-realizable for some q; it is then realizable over every larger alphabet by embedding.

Definition 2.6 ((Average) list-decodability)**.** We say $C \subseteq \Sigma^n$ is (β, ℓ)-list-decodable at r if every collection $L \subseteq C$ of ℓ codewords contains some $c \in L$ such that

$$\text{agree}(c, r) < \beta.$$

We say $C \subseteq \Sigma^n$ is (β, ℓ)-average-list-decodable at r if for every collection $L \subseteq C$ of ℓ codewords, we have

$$\mathbb{E}_{c \in L}[\text{agree}(c, r)] < \beta.$$

Definition 2.7 (Average list-decoding threshold)**.** The *average list-decoding threshold* for a list-decoding problem with parameters P is a value $\beta^*(P) \in [0, 1]$ such that for every $\epsilon > 0$:

- There is a constant C_P depending only on P such that if $\beta = \beta^*(P) + \epsilon$, then in every instance with parameters P, C is $(\beta, \frac{C_P}{\epsilon})$-average-list-decodable at r.
- If $\beta \leq \beta^*(P) - \epsilon$ for any $\epsilon > 0$, then there exists an instance with parameters P such that C is **not** $(\beta, \exp(\Omega(n)))$-list-decodable.

In the large-alphabet case, the lower bound (the second bullet point) is only required to hold when the alphabet size is sufficiently large as a function of the other parameters and $1/\epsilon$. More specifically, our lower bounds hold so long as the alphabet has size at least $m + 1 + \lceil \frac{4}{\epsilon\alpha} \rceil$.

Remark 1. Average list decodability implies standard list decodability. Also note that the negative direction in the above definition asks for large list sizes for standard list-decodability, not average list decodability.

3 Johnson Bound with a hint, q large

The list-decoding threshold in this setting is captured by the following optimization problem. [§]

Optimization 3.1. *$\alpha, \eta \in [0, 1]$ are fixed parameters, and the decision variables are a, b.*

$$\begin{aligned} \max \quad & a + b \\ s.t. \quad & \frac{a^2}{\eta} + \frac{b^2}{1 - \eta} \leq \alpha \\ & 0 \leq a \leq \min(\eta, \alpha) \\ & 0 \leq b \leq 1 - \eta \end{aligned}$$

Theorem 3.2. *Consider the list-decoding problem with a single hint. Let $\eta \in [0, 1]$ be the agreement between the hint and the received word, and let $\alpha \in (0, 1)$ be the maximum agreement of the code. Let*

[§] When a denominator such as η or $1 - \eta$ is zero, we use the convention that the corresponding variable is forced to be zero and the associated quadratic term is omitted. We use the same convention for η_S and η_π, which appear in later sections.

$\beta^*(\alpha, \eta)$ be the optimum of Optimization 3.1. Then $\beta^*(\alpha, \eta)$ is the average list-decoding threshold for the one-hint, large-alphabet list-decoding problem.

Proof of the upper bound ($\beta > \beta^*$). Let C be a code of maximum agreement α over a large alphabet Σ. Let $r \in \Sigma^n$ be a received word and $h \in C$ a hint such that $\text{agree}(r, h) = \eta$. Let $\epsilon > 0$, and let $\beta = \beta^*(\alpha, \eta) + \epsilon$ be the target agreement. Suppose $L \subseteq C \setminus \{h\}$ is a list of codewords such that $\mathbb{E}_{c\sim L}[\text{agree}(c, r)] \geq \beta$. We show that $|L| \leq \frac{1}{\epsilon\alpha}$.

Let $p_{i,\sigma} \coloneqq \Pr_{c\sim L}[c_i = \sigma]$ be the fraction of codewords in the list that place symbol σ at coordinate i. By Lemma 2.3, we have the following inequality.

$$\left(\frac{1}{n}\sum_{i=1}^{n}\sum_{\sigma\in\Sigma} p_{i,\sigma}^2\right) - \frac{1}{|L|} \leq \alpha. \tag{3}$$

Let $U \subseteq [n]$ be the set of coordinates where r and h agree, and note that $|U| = \eta n$. Define

$$a \coloneqq \frac{1}{n}\sum_{i\in U} p_{i,r_i}, \qquad b \coloneqq \frac{1}{n}\sum_{i\notin U} p_{i,r_i}.$$

Since the average agreement of codewords in L with r is at least β, linearity of expectation gives $a + b = \mathbb{E}_{c\sim L}[\text{agree}(c, r)] \geq \beta$.

We claim that a, b satisfy all of the linear constraints of Optimization 3.1. For the $a \leq \alpha$ constraint, note that

$$a = \frac{1}{n}\sum_{i\in U} p_{i,r_i} = \frac{1}{n}\sum_{i\in U} p_{i,h_i} \leq \frac{1}{n}\sum_{i=1}^{n} p_{i,h_i}.$$

Since the RHS of the above is the average agreement of codewords in L with h, and since every codeword in C has agreement at most α with h, we have $a \leq \alpha$. The constraints $a \leq \eta$ and $b \leq 1-\eta$ follow trivially from the definitions.

Next, we lower bound the sum in (3). First, drop all the terms $p_{i,\sigma}$ where $\sigma \neq r_i$. Then split the remaining sum over U and its complement and apply Cauchy-Schwarz to each part. Thus, we have

$$\frac{a^2}{\eta} + \frac{b^2}{1-\eta} \;\leq\; \alpha + \frac{1}{|L|}. \tag{4}$$

Define $f(a, b) \coloneqq \frac{a^2}{\eta} + \frac{b^2}{1-\eta}$. We show that $f(a, b) \geq \alpha + \Omega(\epsilon)$, which, combined with (4), gives $|L| = O(1/\epsilon)$.

Let $\mathcal{X}$ denote the polyhedron defined by the linear constraints of the optimization (i.e., all the constraints except for $f(a, b) \leq \alpha$). Let $x^* = (a^*, b^*)$ be the optimal solution to Optimization 3.1. Since $z = (0, 0) \in \mathcal{X}$ satisfies $f(z) = 0 < \alpha$, strong duality holds by Slater's relaxed condition. Thus, there exists a KKT multiplier $\lambda \geq 0$ such that x^* maximizes the Lagrangian

$$\mathcal{L}(a, b) \;=\; (a + b) - \lambda\big(f(a, b) - \alpha\big)$$

over $\mathcal{X}$, with complementary slackness $\lambda(f(x^*) - \alpha) = 0$. If $\lambda = 0$, then x^* maximizes $a + b$ over $\mathcal{X}$ unconditionally, contradicting $a + b > \beta^* = a^* + b^*$ for our $x = (a, b) \in \mathcal{X}$. So $\lambda > 0$. Additionally, we have $f(x^*) = \alpha$ by complementary slackness. By the first-order condition for convexity of f,

$$f(x) \;\geq\; f(x^*) + \langle \nabla f(x^*),\, x - x^* \rangle \;=\; \alpha + \langle \nabla f(x^*),\, x - x^* \rangle.$$

So it suffices to show $\langle \nabla f(x^*),\, x - x^* \rangle \geq \epsilon\alpha$. Since x^* maximizes $\mathcal{L}$ over $\mathcal{X}$ and $x \in \mathcal{X}$, the first-order optimality condition gives $\langle \nabla \mathcal{L}(x^*),\, x - x^* \rangle \leq 0$. Expanding $\nabla \mathcal{L}(x^*) = (1,1) - \lambda \nabla f(x^*)$ yields

$$\epsilon \;\leq\; (a+b) - (a^*+b^*) \;\leq\; \lambda \langle \nabla f(x^*),\, x - x^* \rangle.$$

To bound λ, use $\mathcal{L}(z) \leq \mathcal{L}(x^*)$, which gives $\lambda\alpha \leq \beta^*$, so $\lambda \leq \beta^*/\alpha$. Combining, we get $\langle \nabla f(x^*),\, x - x^* \rangle \geq \epsilon/\lambda \geq \epsilon\alpha/\beta^* \geq \epsilon\alpha$ as required. □

Remark 2 (List size)**.** The proof gives the quantitative bound

$$|L| \leq \frac{\beta^*}{\alpha(\beta - \beta^*)}.$$

Proof of the lower bound ($\beta < \beta^*$). Fix $\epsilon > 0$ and suppose $\beta = \beta^*(\alpha, \eta) - \epsilon$. Let (a^*, b^*) be an optimal solution to Optimization 3.1, so that $a^* + b^* = \beta^*$ and (a^*, b^*) satisfies all constraints. Define $(a, b) \coloneqq (1 - \epsilon/\beta^*)\,(a^*, b^*)$. Then $a + b = \beta^* - \epsilon = \beta$, and the linear constraints remain satisfied since we are scaling down. For the ellipsoid constraint, by the scaling of (a, b) we have $f(a, b) = (1 - \epsilon/\beta^*)^2 f(a^*, b^*)$. Since $\beta^* \leq 1$, the factor $(1 - \epsilon/\beta^*)^2 \leq (1 - \epsilon)^2 \leq 1 - \epsilon$. Furthermore, $f(a^*, b^*) \leq \alpha$. Combining these inequalities, we obtain $f(a, b) \leq (1 - \epsilon)\alpha = \alpha - \delta$, where $\delta \coloneqq \epsilon\alpha$. Assume $|\Sigma| \geq 2 + \lceil 4/\delta \rceil$, and write $\Sigma = \{0, 1\} \cup \Gamma$ where $|\Gamma| \geq \lceil 4/\delta \rceil$.

We construct a received word r, a hint h, and a code L of size $\exp(\Omega(n))$ such that every codeword in L has agreement at least β with r, and $L \cup \{h\}$ has maximum pairwise agreement at most α.

Let r be the all-zeros word. Let h be the word with the first ηn coordinates equal to 0 and the remaining $(1 - \eta)n$ coordinates equal to 1. Let $U = \{1, \ldots, \eta n\}$ and $V = [n] \setminus U$, so that U is the set of coordinates where r and h agree. We build L as follows so that each codeword has exactly an zeros on U and exactly bn zeros on V.[¶]

Define a type t_U over Σ of length $|U|$ by $t_U(0) = an$, $t_U(1) = 0$, and $t_U(\gamma) = (|U| - an)/|\Gamma|$ for each $\gamma \in \Gamma$. By Lemma 2.2, there exists a code $L_U \subseteq \Sigma^U$ of size $\exp(\Omega(|U|))$, all of whose codewords have type t_U, and whose maximum agreement (normalized by $|U|$) is at most

$$\sum_{\sigma \in \Sigma} \left(\frac{t_U(\sigma)}{|U|}\right)^2 + \frac{\delta}{4} \;=\; \left(\frac{a}{\eta}\right)^2 + \frac{(1 - a/\eta)^2}{|\Gamma|} + \frac{\delta}{4} \;\leq\; \left(\frac{a}{\eta}\right)^2 + \frac{\delta}{2},$$

where the last inequality uses $|\Gamma| \geq 4/\delta$.

Similarly, define a type t_V over Σ of length $|V|$ by setting $t_V(0) = bn$, $t_V(1) = 0$, and spreading the remaining $|V| - bn$ symbols evenly across Γ. By Lemma 2.2, there exists $L_V \subseteq \Sigma^V$ of size $\exp(\Omega(|V|))$ with maximum agreement (normalized by $|V|$) at most $(b/(1 - \eta))^2 + \delta/2$.

Suppose one block, say V, has size $o(n)$. Then $1 - \eta = o(1)$, hence any agreement on V is $o(1)$ and we can define a code with the desired properties by taking L_U and setting the coordinates in V to an arbitrary value $\sigma \in \Gamma$. Since the agreement in V is $o(1)$, this can be absorbed in the constant slack δ.

Thus, assume both blocks have size $\Omega(n)$. Let $N \coloneqq \min(|L_U|, |L_V|) = \exp(\Omega(n))$, and fix arbitrary enumerations $L_U = \{u_1, \ldots\}$ and $L_V = \{v_1, \ldots\}$. Set $L \coloneqq \{u_j \circ v_j : j \in [N]\}$, so $|L| = \exp(\Omega(n))$.

[¶]Fixing the type ensures that the agreement of every codeword in L with the hint h is exactly $a \leq \alpha$; sampling coordinates independently could produce codewords exceeding the agreement bound with non-negligible probability.

It remains to verify the required properties. First, we show that $L \cup \{h\}$ has maximum agreement at most α. For distinct $c = u_j \circ v_j,\ c' = u_k \circ v_k$ in L (with $j \neq k$, so $u_j \neq u_k$ and $v_j \neq v_k$),

$$\begin{aligned}\operatorname{agree}(c, c') &= \frac{1}{n}\big(\operatorname{Agree}_U(u, u') + \operatorname{Agree}_V(v, v')\big) \\ &\leq \eta\Big(\Big(\frac{a}{\eta}\Big)^2 + \frac{\delta}{2}\Big) + (1-\eta)\Big(\Big(\frac{b}{1-\eta}\Big)^2 + \frac{\delta}{2}\Big) \\ &= \frac{a^2}{\eta} + \frac{b^2}{1-\eta} + \frac{\delta}{2} \ \leq\ \alpha - \frac{\delta}{2}.\end{aligned}$$

For any $c \in L$, its agreement with the hint is $\operatorname{agree}(c, h) = a \leq \alpha$, since each codeword in L has exactly an zeros on U (where h is also zero) and no ones on V (where h equals 1). Finally, each $c \in L$ satisfies $\operatorname{agree}(c, r) = a + b = \beta$. □

Next, we solve Optimization 3.1 analytically to find the exact threshold β^*.

Theorem 3.3. *Fix parameters $(\alpha, \eta) \in [0,1]^2$. Define the critical threshold*

$$\eta_{\text{crit}} := \frac{(1-\alpha) + \sqrt{1 - 2\alpha + 5\alpha^2}}{2}.$$

The exact average list-decoding threshold $\beta^* = \beta^*(\alpha, \eta)$ of Optimization 3.1 is

$$\beta^* = \begin{cases} \sqrt{\alpha}, & \eta \leq \sqrt{\alpha}, \\ \alpha + \sqrt{(1-\eta)\Big(\alpha - \frac{\alpha^2}{\eta}\Big)}, & \sqrt{\alpha} < \eta \leq \eta_{\text{crit}}, \\ \alpha + 1 - \eta, & \eta > \eta_{\text{crit}}. \end{cases}$$

Moreover, the unique optimal solution is

$$(a^*, b^*) = \begin{cases} (\eta\sqrt{\alpha}, (1-\eta)\sqrt{\alpha}), & \eta \leq \sqrt{\alpha}, \\ \Big(\alpha, \sqrt{(1-\eta)\Big(\alpha - \frac{\alpha^2}{\eta}\Big)}\Big), & \sqrt{\alpha} < \eta \leq \eta_{\text{crit}}, \\ (\alpha, 1-\eta), & \eta > \eta_{\text{crit}}. \end{cases}$$

Proof of Theorem 3.3. We first solve the relaxation that ignores all linear constraints, keeping only the ellipsoid $\frac{a^2}{\eta} + \frac{b^2}{1-\eta} \leq \alpha$ and $a, b \geq 0$. By Cauchy-Schwarz, $(a+b)^2 \leq \big(\frac{a^2}{\eta} + \frac{b^2}{1-\eta}\big)(\eta + (1-\eta)) \leq \alpha$, with equality when $a/\eta = b/(1-\eta)$. The relaxed optimum is therefore $a + b = \sqrt{\alpha}$, achieved at $(a, b) = (\eta\sqrt{\alpha}, (1-\eta)\sqrt{\alpha})$.

Regime 1 ($\eta \leq \sqrt{\alpha}$): The relaxed optimum satisfies all linear constraints: $a = \eta\sqrt{\alpha} \leq \alpha$ is equivalent to $\eta \leq \sqrt{\alpha}$, and $a \leq \eta$ and $b \leq 1 - \eta$ hold since $\sqrt{\alpha} \leq 1$. Thus $\beta^* = \sqrt{\alpha}$.

Regime 2 ($\sqrt{\alpha} < \eta \leq \eta_{\text{crit}}$): The relaxed optimum has $a = \eta\sqrt{\alpha} > \alpha$, violating the constraint $a \leq \alpha$. Note that $a \leq \eta$ is not the binding constraint, since $\eta > \sqrt{\alpha} > \alpha$ (using $\alpha < 1$). The optimum of the original problem must therefore have $a = \alpha$, and we maximize b subject to the remaining constraints. The ellipsoid gives $\frac{b^2}{1-\eta} \leq \alpha - \frac{\alpha^2}{\eta} = \frac{\alpha(\eta-\alpha)}{\eta}$, which is positive since $\eta > \alpha$. The largest b satisfying the ellipsoid is $b_{\text{ell}} := \sqrt{(1-\eta) \cdot \frac{\alpha(\eta-\alpha)}{\eta}}$. We also require $b \leq 1 - \eta$, so $b^* = \min(b_{\text{ell}},\ 1 - \eta)$.

We determine when $b_{\text{ell}} > 1 - \eta$.

$$\begin{aligned}(1-\eta)\cdot\frac{\alpha(\eta-\alpha)}{\eta} &> (1-\eta)^2\\ \frac{\alpha(\eta-\alpha)}{\eta} &> 1-\eta\\ \alpha(\eta-\alpha) &> \eta(1-\eta)\\ \eta^2+(\alpha-1)\eta-\alpha^2 &> 0.\end{aligned}$$

The left-hand side is a quadratic in η with unique positive root $\eta_{\text{crit}} = \frac{(1-\alpha)+\sqrt{1-2\alpha+5\alpha^2}}{2}$. For $\eta \le \eta_{\text{crit}}$, we have $b_{\text{ell}} \le 1-\eta$, so $b^* = b_{\text{ell}}$ and $\beta^* = \alpha + \sqrt{(1-\eta)\left(\alpha - \frac{\alpha^2}{\eta}\right)}$.

Regime 3 ($\eta > \eta_{\text{crit}}$): Now $b_{\text{ell}} > 1-\eta$, so the constraint $b \le 1-\eta$ binds before the ellipsoid. Thus $b^* = 1-\eta$ and $\beta^* = \alpha + 1 - \eta$. □

Note that $\sqrt{\alpha}$ is the list-decoding threshold given by the standard Johnson bound (without hints). We interpret the three regimes as follows. In Regime 1 the hint is outside of the standard Johnson radius and is not helpful. As soon as the hint is within the Johnson radius (Regime 2), the hint can be used to improve the list-decoding threshold. As the hint gets even closer to the received word (Regime 3), we simply get the triangle inequality. Regime 2 is the most interesting, and we study the improvement in this regime over the standard Johnson bound.

Theorem 3.4. *Fix $\alpha \in (0,1)$. For $\eta = \sqrt{\alpha} + \epsilon$ with sufficiently small $\epsilon > 0$,*

$$\beta^*(\alpha,\eta) = \sqrt{\alpha} - \frac{\epsilon^2}{2(1-\sqrt{\alpha})} + O_\alpha(\epsilon^3).$$

In particular, a hint with agreement exceeding the Johnson radius by ϵ improves the list-decoding threshold by an additive term $\Theta(\epsilon^2)$.

Proof. For sufficiently small ϵ, we are in the second regime of Theorem 3.3, so

$$\beta^* = \alpha + \sqrt{(1-\eta)\alpha(\eta-\alpha)/\eta}.$$

The result follows by Taylor-expanding around $\eta = \sqrt{\alpha}$. □

Figure 1 illustrates the improvement.

Let $\beta_{\text{Johnson}} = \sqrt{\alpha}$ be the list-decoding threshold given by the Johnson bound. As a corollary, we get a result on the structure of lists showing the tightness of the Johnson bound. In particular, for any exponential-size list L of codewords all with agreement at least $\beta_{\text{Johnson}} - \delta$, all elements of L have agreement at most $\beta_{\text{Johnson}} + O_\alpha(\sqrt{\delta})$. Any element of the list with higher agreement could be used as a hint and, in conjunction with the previous theorems, would imply that the list size must be polynomial.

Corollary 3.5. *Let C be a code with maximum agreement α over a large alphabet and let $r \in \Sigma^n$. For any sufficiently small constant $\delta > 0$, if $L \subseteq C$ is a list of $\exp(\Omega(n))$ codewords, each with agreement at least $\sqrt{\alpha} - \delta$ with r, then every codeword $c \in L$ satisfies $\text{agree}(c,r) \le \sqrt{\alpha} + O_\alpha(\sqrt{\delta})$.*

Proof. Suppose some $c \in L$ has $\text{agree}(c, r) = \sqrt{\alpha} + \epsilon$ for $\epsilon > 0$. Using c as a hint, Theorem 3.4 gives decoding threshold $\beta^* = \sqrt{\alpha} - \frac{\epsilon^2}{2(1-\sqrt{\alpha})} + O_\alpha(\epsilon^3)$. For $\epsilon \geq C\sqrt{\delta}$ with C a sufficiently large constant depending on α, we get $\beta^* < \sqrt{\alpha} - \delta$. The remaining $\exp(\Omega(n)) - 1$ codewords each have agreement at least $\sqrt{\alpha} - \delta > \beta^*$, so by Theorem 3.2, the list has polynomial size, a contradiction. □

Remark 3. For comparison, the triangle inequality gives only the weaker bound $\text{agree}(c, r) \leq \sqrt{\alpha} + (1 - \sqrt{\alpha})^2 + \delta$. Indeed, if $c_1, c_2 \in C$ have agreements β_1, β_2 with r, the positions where both agree with r overlap in at least $(\beta_1 + \beta_2 - 1)n$ coordinates, so $\text{agree}(c_1, c_2) \geq \beta_1 + \beta_2 - 1$. Since $\text{agree}(c_1, c_2) \leq \alpha$, we get $\beta_1 + \beta_2 \leq 1 + \alpha$. Taking $\beta_2 \geq \sqrt{\alpha} - \delta$ yields $\beta_1 \leq 1 + \alpha - \sqrt{\alpha} + \delta = \sqrt{\alpha} + (1 - \sqrt{\alpha})^2 + \delta$. Unlike Corollary 3.5, this allows a constant gap of $(1 - \sqrt{\alpha})^2$ above $\sqrt{\alpha}$ even as $\delta \to 0$.

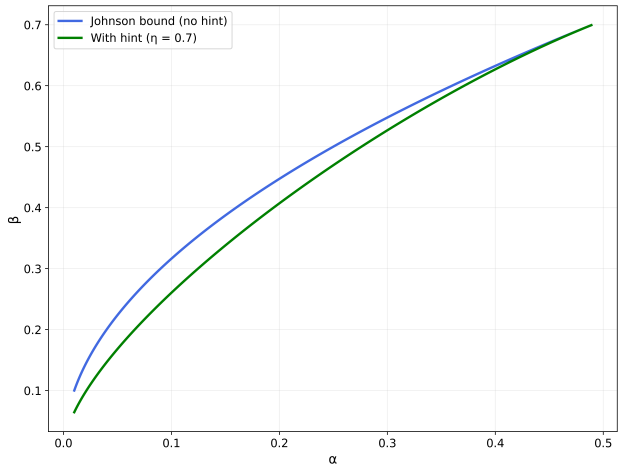


Figure 1: List decoding with a hint of agreement $\eta = 0.7$. The x-axis is α and the y-axis is the decoding threshold β. The blue curve is the Johnson bound without a hint and the green curve is the Johnson bound with a hint of agreement η.

4 Johnson Bound with a hint, q small

The proof of the lower bound breaks down when q is small because the agreement between codewords in the list on symbols that are not equal to r and h is no longer negligible. As an extreme example, consider the case of $q = 2$. Take r to be the all-zeros string, and let h be the string consisting of ηn 0s followed by $(1 - \eta)n$ 1s. Let a be the normalized agreement with r on the first ηn coordinates, i.e., $a = \frac{1}{n} \sum_{i=1}^{\eta n} p_{i,0}$. Since $q = 2$, the remaining mass on these coordinates must all go to symbol 1, contributing $\eta - a$ in total. By Cauchy-Schwarz, the pairwise agreement between codewords on these coordinates is at least $\frac{a^2}{\eta} + \frac{(\eta - a)^2}{\eta}$, whereas for large q the second term is $O(1/q)$ and can be absorbed into the slack. Similarly, on the last $(1-\eta)n$ coordinates, the agreement with r (symbol 0) and with h (symbol 1) together determine the full distribution, and both contribute non-negligibly to pairwise agreement.

To handle the small-alphabet case, we adapt the proof by partitioning all of the $p_{i,\sigma}$ (not just those with $\sigma = r_i$). In particular, we partition $[n] \times \Sigma$ into five parts as follows.

- U: coordinates where $r_i = h_i$ and symbol $\sigma = r_i$.

- V: coordinates where $r_i = h_i$ and symbol $\sigma \neq r_i$.
- W: coordinates where $r_i \neq h_i$ and symbol $\sigma = r_i$.
- Z: coordinates where $r_i \neq h_i$ and symbol $\sigma = h_i$.
- Y: coordinates where $r_i \neq h_i$ and symbol $\sigma \neq r_i$ and $\sigma \neq h_i$.

We define macroscopic variables u, w, z as the normalized sums over parts U, W, and Z:

$$u \coloneqq \frac{1}{n} \sum_{(i,\sigma)\in U} p_{i,\sigma}, \qquad w \coloneqq \frac{1}{n} \sum_{(i,\sigma)\in W} p_{i,\sigma}, \qquad z \coloneqq \frac{1}{n} \sum_{(i,\sigma)\in Z} p_{i,\sigma}.$$

Then the normalized sums over the remaining parts V and Y are $\eta - u$ and $1-\eta-w-z$, respectively. Applying Cauchy-Schwarz to each of the five groups, the sum $\frac{1}{n}\sum_{i,\sigma} p_{i,\sigma}^2$ is lower bounded by the quadratic expression in u, w, z appearing as the ellipsoid constraint below. Together with the linear constraints from the hint agreement and the box constraints, this yields the following optimization for the exact list-decoding threshold.

Optimization 4.1.

$$\begin{aligned} \max \quad & u + w \\ s.t. \quad & \frac{u^2 + \frac{(\eta-u)^2}{q-1}}{\eta} + \frac{w^2 + z^2 + \frac{(1-\eta-w-z)^2}{q-2}}{1-\eta} \leq \alpha \\ & u + z \leq \alpha \\ & u \leq \eta \\ & w + z \leq 1 - \eta \\ & u, w, z \in [0, 1] \end{aligned}$$

Note the following convention for $q = 2$. On a coordinate where $r_i \neq h_i$, the two available symbols are precisely r_i and h_i. Accordingly, in the binary case $w + z = 1 - \eta$, and we omit the term involving $q - 2$ in the quadratic constraint, which avoids dividing by zero.

For parameters (α, η, q), let $\beta^*(\alpha, \eta, q)$ be the optimum of Optimization 4.1.

Theorem 4.2. *Consider the list-decoding problem with a single hint over an alphabet of size $q \geq 2$. Let $\eta \in [0, 1]$ be the agreement between the hint and the received word, and let $\alpha \in (0, 1)$ be the maximum agreement of the code, satisfying $\alpha > 1/q$. Let $\beta^*(\alpha, \eta, q)$ be the optimum of Optimization 4.1. Then $\beta^*(\alpha, \eta, q)$ is the average list-decoding threshold for the one-hint, small-alphabet list-decoding problem.*

We omit the proof since it is almost identical to that of Theorem 3.2 and the key difference is described above. The other difference is in the Slater point used. Instead of taking the all-zeros point, we take a point corresponding to the uniform distribution on the alphabet. In particular, we use the Slater point $(\eta/q, (1-\eta)/q, (1-\eta)/q)$. This idea is also present in the most general case, Theorem 6.4, for which we give a full proof.

5 Johnson Bound with multiple hints, q large

When there are multiple hints $h_1, \ldots, h_m$, the received word may agree with multiple hints simultaneously. In the single-hint case, each coordinate belongs to one of two groups depending on whether

the hint agrees with the received word. With m hints, each coordinate i is characterized by the subset $S \subseteq [m]$ of hints that agree with r at that coordinate, yielding up to 2^m groups.

Formally, for each $S \subseteq [m]$, define $U_S = \{i \in [n] : \{j : h_{j,i} = r_i\} = S\}$, and let $\eta_S = |U_S|/n$. The vector $\eta \in [0,1]^{\mathcal{P}([m])}$ with $\sum_{S \subseteq [m]} \eta_S = 1$ records which hints agree with the received word.

The macroscopic variables and constraints are defined as before, using this finer partition of coordinates. This yields the following optimization problem.

Optimization 5.1. *The parameters are $\alpha \in [0,1]$, $m \in \mathbb{N}$, and $\eta \in [0,1]^{\mathcal{P}([m])}$. The decision variables are $(a_S)_{S \subseteq [m]}$.*

$$\begin{aligned} \max \quad & \sum_{S \subseteq [m]} a_S \\ s.t. \quad & \sum_{S \subseteq [m]} \frac{a_S^2}{\eta_S} \leq \alpha \\ & \sum_{S \ni j} a_S \leq \alpha && \forall j \in [m] \\ & 0 \leq a_S \leq \eta_S && \forall S \subseteq [m] \end{aligned}$$

Let $\beta^*(\alpha, m, \eta)$ be the optimum of Optimization 5.1.

Theorem 5.2. *Consider the list-decoding problem with m hints over a large alphabet. Let $\eta \in [0,1]^{\mathcal{P}([m])}$ be an α-realizable agreement profile between the hints and the received word, with η_S denoting the fraction of coordinates where exactly the hints in S agree with the received word, and let $\alpha \in (0,1)$ be the maximum agreement of the code. Let $\beta^*(\alpha, m, \eta)$ be the optimum of Optimization 5.1. Then $\beta^*(\alpha, m, \eta)$ is the average list-decoding threshold for the multiple-hint, large-alphabet list-decoding problem.*

Again, the proof is almost identical to that of Theorem 3.2, and the main difference is simply taking a finer partition of the coordinates based on which hints agree with the received word. For the lower bound, take r to be constant and assign a distinct non-r symbol to each hint wherever it disagrees with r. Then $\text{agree}(h_j, h_k) = \sum_{S \supseteq \{j,k\}} \eta_S \leq \alpha$ by realizability. We omit the details.

6 Johnson Bound with multiple hints, general case

In the previous section, we partitioned the coordinates based on which hints agree with the received word. This suffices in the large-alphabet regime because on coordinates where a hint disagrees with r, the hint symbol is distinct from the received symbol and contributes negligibly to agreement.

When the alphabet is small, the situation is more subtle. Two hints that both disagree with r at a given coordinate may place the *same* non-received symbol there, creating additional structure. More generally, the agreement between codewords, the received word, and the hints depends not just on which hints match r, but on the full pattern of symbol equalities among all of them.

Consider $q = 3$ with $\Sigma = \{0,1,2\}$, and two hints h_1, h_2. Suppose $r_i = 0$, $h_{1,i} = 1$, and $h_{2,i} = 1$. Both hints disagree with r, so in the partition from the previous section, this coordinate belongs to the group $S = \emptyset$. However, the two hints place the *same* symbol at this coordinate. A codeword

placing symbol 1 simultaneously agrees with both hints and disagrees with r, which is a stronger constraint than if the two hints placed distinct non-received symbols (say $h_{1,i} = 1$ and $h_{2,i} = 2$). The general case must distinguish these two configurations.

Thus, this most general case uses an even finer partition of coordinates based on the full pattern of symbol equalities among r and all hints. At each coordinate i, we record which of $r, h_1, \ldots, h_m$ place the same symbol, forming a partition of $\{0, 1, \ldots, m\}$ (where 0 represents r). Coordinates sharing the same partition type are grouped together. We now set up the notation.

Let $r = h_0$ and $h_1, \ldots, h_m \in C$. For each $i \in [n]$, let π_i be the partition of $\{0, 1, \ldots, m\}$ induced by the symbols appearing at coordinate i. Formally, for each coordinate $i \in [n]$, define the equivalence relation $\sim_i$ on $\{0, 1, \ldots, m\}$ such that $j \sim_i j'$ if and only if $h_{j,i} = h_{j',i}$. π_i is then the partition on $\{0, 1, \ldots, m\}$ induced by this equivalence relation. For each $i \in [n]$ and each part $S \in \pi_i$, define the symbol $\sigma_{i,S}$ to be the symbol that the members of S take at coordinate i. Let Π be the set of all partitions of $\{0, 1, \ldots, m\}$. For each $\pi \in \Pi$, let $U_\pi = \{i \in [n] : \pi_i = \pi\}$. The parameter $\eta \in [0, 1]^\Pi$ records the frequency of each partition type: $\eta_\pi = |U_\pi|/n$ is the fraction of coordinates $i \in [n]$ with $\pi_i = \pi$. We write s_π for the number of parts in π.

Optimization 6.1 (Multiple hints)**.** *The parameters are $\alpha \in [0, 1]$, an alphabet Σ of size q, $m \in \mathbb{N}$, and $\eta \in [0, 1]^\Pi$. The decision variables are $(a_{\pi,S})_{\pi\in\Pi, S\in\pi}$ and $(b_\pi)_{\pi\in\Pi}$.*

$$
\begin{aligned}
\max \quad & \sum_{\substack{\pi\in\Pi \\ S\in\pi \\ S\ni 0}} a_{\pi,S} \\
s.t. \quad & \sum_{\substack{\pi\in\Pi \\ S\in\pi}} \frac{a_{\pi,S}^2}{\eta_\pi} + \sum_{\substack{\pi\in\Pi \\ q>s_\pi}} \frac{b_\pi^2}{(q-s_\pi)\eta_\pi} \le \alpha \\
& \sum_{\substack{\pi\in\Pi, \ S\in\pi \\ S\ni j}} a_{\pi,S} \le \alpha && \forall j \in [m] \\
& b_\pi + \sum_{S\in\pi} a_{\pi,S} = \eta_\pi, && \forall \pi \in \Pi \\
& a_{\pi,S}, b_\pi \in [0, 1] && \forall \pi \in \Pi, S \in \pi
\end{aligned}
$$

Let $\beta^*(\alpha, q, m, \eta)$ denote the optimum of Optimization 6.1.

Before proving that β^* is a list-decoding threshold, we record some facts about Optimization 6.1. The first lemma establishes a Slater point, a strictly feasible solution, which guarantees strong duality and shows $\beta^* > 1/q$. The Slater point is based on the uniform distribution. The second lemma is a lower bound on the directional derivative at the optimal solution in the direction of a point x that obtains a higher objective value and satisfies the linear constraints but not the ellipsoid constraint. This will be a key ingredient for the upper bound on list size and the decoding algorithm analysis. Note that the assumption $\alpha > 1/q$ below is natural: by the Plotkin bound, any code over a q-ary alphabet with maximum agreement at most $1/q$ has only polynomially many codewords, so the list-decoding problem is only interesting when $\alpha > 1/q$.

Lemma 6.2 (Slater point for Optimization 6.1)**.** *Suppose $\alpha > 1/q$. Define z by $a_{\pi,S} = \eta_\pi/q$ for all $\pi \in \Pi, S \in \pi$ and $b_\pi = (q - s_\pi)\eta_\pi/q$ for all $\pi \in \Pi$. Then z is feasible for Optimization 6.1 with $f(z) = 1/q < \alpha$ and objective $c^T z = 1/q$. Furthermore, $\beta^*(\alpha, q, m, \eta) > 1/q$.*

Proof. The equality constraints hold: $\sum_{S\in\pi} a_{\pi,S} + b_\pi = s_\pi\eta_\pi/q + (q-s_\pi)\eta_\pi/q = \eta_\pi$. The capacity constraints give $\sum_{\pi,\, S\ni j} a_{\pi,S} = \sum_\pi \eta_\pi/q = 1/q \le \alpha$, and the box constraints are easily verified. The ellipsoid value is

$$f(z) \;=\; \sum_{\pi,S} \frac{\eta_\pi}{q^2} + \sum_{\pi:\, q>s_\pi} \frac{(q-s_\pi)\eta_\pi}{q^2} \;=\; \sum_\pi \frac{s_\pi\eta_\pi + (q-s_\pi)\eta_\pi}{q^2} \;=\; \frac{1}{q} \;<\; \alpha.$$

Since $f(z) < \alpha$, we can increase the objective beyond $1/q$ while remaining feasible. Pick any π with $\eta_\pi > 0$ and any $S \in \pi$ with $0 \in S$. If $q > s_\pi$ (so that $b_\pi = (q-s_\pi)\eta_\pi/q > 0$), increase $a_{\pi,S}$ by a small $\mu > 0$ and decrease b_π by μ. If $q = s_\pi$ (so that $b_\pi = 0$), pick any $S' \in \pi$ with $0 \notin S'$ (which exists since $s_\pi = q \ge 2$), and increase $a_{\pi,S}$ by μ while decreasing $a_{\pi,S'}$ by μ. In both cases, the equality constraint is preserved and the objective increases by μ. The capacity constraints remain satisfied for small μ, since at z each has value $1/q < \alpha$. The ellipsoid and box constraints are likewise preserved for μ small enough. Thus $\beta^* > c^T z = 1/q$. □

Lemma 6.3. *Consider the optimization problem*

$$\max_{x\in\mathcal{X}} c^T x \quad \textit{subject to} \quad f(x) \le \alpha,$$

where f is convex and differentiable and $\mathcal{X}$ is a convex set defined by linear constraints. Let x^* be an optimal solution with value $\beta^* = c^T x^*$. If there exists a Slater point $z \in \mathcal{X}$ with $f(z) < \alpha$, and a point $x \in \mathcal{X}$ with $c^T(x - x^*) = \epsilon > 0$, then

$$\langle \nabla f(x^*),\, x - x^* \rangle \;\ge\; \frac{\epsilon\,(\alpha - f(z))}{\beta^* - c^T z}.$$

Furthermore, $f(x^*) = \alpha$.

Proof. Since $z \in \mathcal{X}$ satisfies $f(z) < \alpha$, Slater's condition holds for the convex constraint $f(x) \le \alpha$. Combined with the linearity of the constraints defining $\mathcal{X}$, strong duality applies and there exists a KKT multiplier $\lambda \ge 0$ such that x^* maximizes the Lagrangian

$$L(x) \;=\; c^T x - \lambda\big(f(x) - \alpha\big)$$

over $x \in \mathcal{X}$, with complementary slackness $\lambda(f(x^*) - \alpha) = 0$.

We first show $\lambda > 0$. If $\lambda = 0$, then x^* maximizes $c^T x$ over $\mathcal{X}$ unconditionally, so $c^T x \le \beta^*$ for all $x \in \mathcal{X}$, contradicting $c^T(x - x^*) = \epsilon > 0$. So $\lambda > 0$. Then, by complementary slackness, we have $f(x^*) = \alpha$, establishing the "furthermore" part of the lemma.

Since x^* maximizes L over the convex set $\mathcal{X}$, the first-order necessary condition for optimality gives $\langle \nabla L(x^*),\, x - x^* \rangle \le 0$. Expanding $\nabla L(x^*) = c - \lambda \nabla f(x^*)$ yields

$$\epsilon \;=\; c^T(x - x^*) \;\le\; \lambda\,\langle \nabla f(x^*),\, x - x^* \rangle.$$

We now upper bound λ. Since x^* maximizes L over $\mathcal{X}$ and $z \in \mathcal{X}$, we have $L(z) \le L(x^*)$. Using $f(x^*) = \alpha$, this gives

$$c^T z - \lambda\big(f(z) - \alpha\big) \;\le\; \beta^*,$$

so $\lambda(\alpha - f(z)) \le \beta^* - c^T z$, and therefore

$$\lambda \;\le\; \frac{\beta^* - c^T z}{\alpha - f(z)}.$$

Combining these inequalities, we obtain

$$\langle \nabla f(x^*),\, x - x^* \rangle \;\geq\; \frac{\epsilon}{\lambda} \;\geq\; \frac{\epsilon\,(\alpha - f(z))}{\beta^* - c^T z}. \qquad \square$$

Theorem 6.4 (Johnson bound with multiple hints)**.** *Consider the list-decoding problem with m hints over an alphabet of size q. Let $\eta \in [0,1]^{\Pi}$ be a (q,α)-realizable partition profile between the received word and the hints, and let $\alpha \in (0,1)$ be the maximum agreement of the code. Suppose $\alpha > 1/q$. Let $\beta^*(\alpha, q, m, \eta)$ be the optimum of Optimization 6.1. Then $\beta^*(\alpha, q, m, \eta)$ is the average list-decoding threshold for the list-decoding problem with parameters (α, q, m, η).*

Proof of the upper bound ($\beta > \beta^*$). Let $(C, r, h_1, \ldots, h_m)$ be an instance of the list-decoding problem with multiple hints and parameters (α, q, m, η). Let $L \subseteq C \setminus \{h_1, \ldots, h_m\}$ be such that $\mathbb{E}_{c\sim L}[\mathrm{agree}(c, r)] \geq \beta$ where $\beta = \beta^* + \epsilon$ for $\epsilon > 0$.

For $i \in [n]$ and $\sigma \in \Sigma$, set $p_{i,\sigma} \coloneqq \Pr_{x\sim L}[x_i = \sigma]$. By Lemma 2.3,

$$\frac{1}{n} \sum_{i\in[n]} \sum_{\sigma\in\Sigma} p_{i,\sigma}^2 \;\leq\; \alpha + \frac{1}{|L|}.$$

Define

$$a_{\pi,S} \;\coloneqq\; \frac{1}{n} \sum_{i\in U_\pi} p_{i,\sigma_{i,S}}, \qquad b_\pi \;\coloneqq\; \frac{1}{n} \sum_{i\in U_\pi} \sum_{\substack{\sigma\in\Sigma \\ \sigma\neq\sigma_{i,S}\;\forall S}} p_{i,\sigma}.$$

Note that (a, b) defined in this way satisfy all of the linear and box constraints. The equality constraints hold since $\sum_{S\in\pi} a_{\pi,S} + b_\pi = \frac{1}{n}\sum_{i\in U_\pi}\sum_\sigma p_{i,\sigma} = \frac{|U_\pi|}{n} = \eta_\pi$. The objective is $\sum_{\pi,\, S\ni 0} a_{\pi,S} = \frac{1}{n}\sum_i p_{i,r_i} = \mathbb{E}_{c\sim L}[\mathrm{agree}(c, r)] \geq \beta$. The capacity constraints hold: for each $j \in [m]$, $\sum_{\pi,\, S\ni j} a_{\pi,S} = \frac{1}{n}\sum_i p_{i,h_{j,i}} \leq \alpha$ since every $c \in L \subseteq C$ satisfies $\mathrm{agree}(c, h_j) \leq \alpha$. For the ellipsoid constraint, Cauchy-Schwarz applied to each group gives

$$\frac{a_{\pi,S}^2}{\eta_\pi} \;\leq\; \frac{1}{n} \sum_{i\in U_\pi} p_{i,\sigma_{i,S}}^2, \qquad \frac{b_\pi^2}{(q - s_\pi)\eta_\pi} \;\leq\; \frac{1}{n} \sum_{i\in U_\pi} \sum_{\sigma\neq\sigma_{i,S}\;\forall S} p_{i,\sigma}^2.$$

Summing over all π and S gives

$$f(a, b) \;\coloneqq\; \sum_{\pi\in\Pi} \sum_{S\in\pi} \frac{a_{\pi,S}^2}{\eta_\pi} + \sum_{\substack{\pi\in\Pi \\ q > s_\pi}} \frac{b_\pi^2}{(q - s_\pi)\eta_\pi} \;\leq\; \frac{1}{n} \sum_{i\in[n]} \sum_{\sigma\in\Sigma} p_{i,\sigma}^2 \;\leq\; \alpha + \frac{1}{|L|}. \tag{5}$$

Let $x = (a, b)$, and $x^* = (a^*, b^*)$. To establish an upper bound of $\frac{1}{\epsilon(\alpha - 1/q)}$ on $|L|$, it suffices to show that $f(x) \geq \alpha + \epsilon(\alpha - 1/q)$. The argument proceeds similarly to that in the single-hint case.

By the first-order condition for convexity, $f(x) \geq f(x^*) + \langle \nabla f(x^*), x - x^* \rangle$. Let $\mathcal{X}$ denote the polytope defined by the linear, equality, and box constraints of Optimization 6.1. Note that the objective is linear, so we can write it as $c^T x$ for some c. By Lemma 6.3 with the Slater point z from Lemma 6.2 (satisfying $f(z) = 1/q$ and $c^T z = 1/q$) and $x \in \mathcal{X}$ with $c^T(x - x^*) \geq \epsilon$, we have

$$\langle \nabla f(x^*),\, x - x^* \rangle \;\geq\; \frac{\epsilon\,(\alpha - 1/q)}{\beta^* - 1/q}, \text{ and } \quad f(x^*) = \alpha.$$

Hence, $f(a,b) \geq \alpha + \frac{\epsilon\,(\alpha-1/q)}{\beta^*-1/q} \geq \alpha + \epsilon(\alpha - 1/q)$, as required.

□

Remark 4 (List size)**.** The proof gives the quantitative bound

$$|L| \leq \frac{\beta^* - 1/q}{(\alpha - 1/q)(\beta - \beta^*)}.$$

Proof of the lower bound. Fix $\epsilon > 0$ and suppose $\beta = \beta^* - \epsilon$. It suffices to consider $\beta > 1/q$. Indeed, if $\beta \leq 1/q$, choose any $\widetilde{\beta} \in (1/q, \beta^*)$ and apply the construction below with $\widetilde{\beta}$ in place of β; since every resulting codeword has agreement at least $\widetilde{\beta} > \beta$, the same instance witnesses the claimed lower bound at β. We may therefore assume $\beta > 1/q$, so $0 < \epsilon < \beta^* - 1/q$. Let (a^*, b^*) be an optimal solution to Optimization 6.1 with $f(a^*, b^*) \leq \alpha$. Let z be the Slater point from Lemma 6.2, with $f(z) = 1/q < \alpha$, objective $c^T z = 1/q$. We define a new feasible point (a, b) with objective $\beta^* - \epsilon$ by taking a convex combination of (a^*, b^*) and z. Namely, let $t \coloneqq \epsilon/(\beta^* - 1/q)$ and define $(a, b) \coloneqq (1-t)(a^*, b^*) + t\,z$. Since the equality constraints are affine, (a, b) is feasible with objective $(1-t)\beta^* + t/q = \beta^* - \epsilon = \beta$. For the ellipsoid, convexity of f gives

$$f(a,b) \leq (1-t)\,f(a^*,b^*) + t\,f(z) \leq (1-t)\alpha + \frac{t}{q} = \alpha - t\left(\alpha - \frac{1}{q}\right) = \alpha - \delta,$$

where $\delta \coloneqq \frac{\epsilon\,(\alpha-1/q)}{\beta^*-1/q} = O(\epsilon) > 0$.

Let $r, h_1, \ldots, h_m$ be a witness to the (q,α)-realizability of η. Since only the coordinatewise equality patterns matter, choose the witness in canonical form so that, for each $\pi \in \Pi$ and $S \in \pi$, the symbol $\sigma_{i,S}$ is independent of $i \in U_\pi$; denote it by $\sigma_{\pi,S}$. We will construct a code L of size $\exp(\Omega(n))$ such that the average agreement between r and L is at least β, and $L \cup \{h_1, \ldots, h_m\}$ is a code with maximum agreement at most α. Assume $|U_\pi| = \Theta(n)$ for all $\pi \in \Pi$; otherwise, we can ignore the coordinates in blocks of size $o(n)$ since they contribute only $o(1)$ to the normalized agreements.

Fix $\pi \in \Pi$. Define a type t_π over Σ by

$$t_\pi(\sigma) \coloneqq \begin{cases} a_{\pi,S} n & \text{if } \sigma = \sigma_{\pi,S} \text{ for some } S \in \pi, \\ \frac{b_\pi}{q - s_\pi} n & \text{otherwise.} \end{cases}$$

The constraint $b_\pi + \sum_{S\in\pi} a_{\pi,S} = \eta_\pi$ ensures $\sum_{\sigma\in\Sigma} t_\pi(\sigma) = |U_\pi|$. (If $q = s_\pi$, we must have $b_\pi = 0$, and we omit all terms dividing by $q - s_\pi$.) By Lemma 2.2, there exists a code $L_\pi \subseteq \Sigma^{U_\pi}$ of size $\exp(\Omega(|U_\pi|))$, all of whose codewords have type t_π and whose maximum agreement (normalized by $|U_\pi|$) is at most

$$\sum_{\sigma\in\Sigma} \left(\frac{t_\pi(\sigma)}{|U_\pi|}\right)^2 + \delta/2 = \sum_{S\in\pi} \left(\frac{a_{\pi,S}}{\eta_\pi}\right)^2 + (q - s_\pi)\left(\frac{b_\pi/(q-s_\pi)}{\eta_\pi}\right)^2 + \delta/2 = \sum_{S\in\pi} \frac{a_{\pi,S}^2}{\eta_\pi^2} + \frac{b_\pi^2}{(q-s_\pi)\,\eta_\pi^2} + \delta/2.$$

Let $N \coloneqq \min_\pi |L_\pi| = \exp(\Omega(n))$, and fix an arbitrary enumeration of each L_π. Define $L \coloneqq \{(c_\pi^{(j)})_{\pi\in\Pi} : j \in [N]\}$, where $c_\pi^{(j)}$ is the j-th element of L_π. Since each L_π has distinct elements, distinct codewords in L differ on every block.

To conclude the proof, we show that L satisfies the required properties. First, we show that distinct codewords in L have agreement at most α. For distinct $c, c' \in L$, since they differ on every block, the agreement normalized by n decomposes as

$$\begin{aligned}\operatorname{agree}(c,c') &\le \sum_{\pi\in\Pi}\eta_\pi\Big(\sum_{S\in\pi}\frac{a_{\pi,S}^2}{\eta_\pi^2}+\frac{b_\pi^2}{(q-s_\pi)\,\eta_\pi^2}+\delta/2\Big)\\ &= \sum_{\pi\in\Pi}\sum_{S\in\pi}\frac{a_{\pi,S}^2}{\eta_\pi} \;+\; \sum_{\pi\in\Pi}\frac{b_\pi^2}{(q-s_\pi)\,\eta_\pi} \;+\; \delta/2\\ &= f(a,b)+\delta/2 \;\le\; (\alpha-\delta)+\delta/2 \;=\; \alpha-\delta/2.\end{aligned}$$

Then, for any $j \in [m]$ and $c \in L$,

$$\operatorname{agree}(c,h_j) = \frac{1}{n}\sum_{\pi\in\Pi} t_\pi(\sigma_{\pi,S(j)}) = \sum_{\substack{\pi\in\Pi,\ S\in\pi\\ S\ni j}} a_{\pi,S} \;\le\; \alpha,$$

where $S(j)$ denotes the part of π (a subset of $\{0,1,\dots,m\}$) containing j. Likewise,

$$\operatorname{agree}(c,r) = \sum_{\substack{\pi\in\Pi,\ S\in\pi\\ S\ni 0}} a_{\pi,S} \;\ge\; \beta.$$

□

7 Decoding RS codes with hints

For Reed-Solomon codes, we can make our bounds algorithmic. Namely, we can decode to the new list-decoding thresholds given in Theorems 3.2 and 5.2 (note that Reed-Solomon codes fall into the large-alphabet case). As mentioned in the introduction, hints arise naturally as part of list decoding. Any codeword found in a first pass by a list-decoding algorithm can then be interpreted as a hint.

By setting weights on the coordinates based on the optimal solutions to our optimization problems, we reduce the problem of decoding with hints to a single call to the weighted Guruswami-Sudan algorithm, the performance guarantee of which we state below.

Theorem 7.1 ([Gur04])**.** *Let C be a Reed-Solomon code of length n and maximum agreement α. Let $w \in \mathbb{R}^n_{\ge 0}$ be nonnegative, and let r be a received word. Let $\epsilon > 0$. Then there is an algorithm that runs in time* $\operatorname{poly}(n, q, 1/\epsilon)$ *and outputs all codewords c such that*

$$\sum_{i:c_i=r_i} w_i > \sqrt{\alpha n \sum_{i=1}^{n} w_i^2} + \epsilon w_{\max} \tag{6}$$

Note that for Reed-Solomon codes, $q \ge n$, so we are in the large-alphabet regime.

Theorem 7.2 (Hint-Aware Decoding)**.** *Let C be a Reed-Solomon code of length n and maximum agreement α. Let $(r, h_1, \dots, h_m)$ be an instance of the list-decoding problem with multiple hints and parameters α, q, m, and $\eta \in [0,1]^{\mathcal{P}([m])}$. Suppose $\beta > \beta^*$, where β^* is the optimum of Optimization 5.1. Then there exists an algorithm that finds all codewords with agreement at least β with r and runs in time* $\operatorname{poly}(n, q, 1/\delta)$*, where $\delta := \beta - \beta^*$.*

Proof. Let $H_\beta := \{h_j : \text{agree}(h_j, r) \geq \beta\}$, and include these supplied hints in the output. It remains to recover every codeword $c \in C \setminus \{h_1, \ldots, h_m\}$ with $\text{agree}(c, r) \geq \beta$; for each such c, we have $\text{agree}(c, h_j) \leq \alpha$ for every $j \in [m]$. If β^* is determined by the linear constraints alone (i.e., $\sum_S a_S \leq \beta^*$ is forced by the capacity and box constraints without the ellipsoid being tight), then no such non-hint codeword can have agreement exceeding β^*, so the algorithm returns H_β.

Otherwise, the optimum is strictly constrained by the ellipsoid, so the optimal solution $a^* = (a^*_S)_{S \subseteq [m]}$ satisfies $f(a^*) := \sum_{S \subseteq [m]} \frac{(a^*_S)^2}{\eta_S} = \alpha$.

For each $S \subseteq [m]$, define the coordinate block $A_S := \{i \in [n] : \{j \in [m] : h_{j,i} = r_i\} = S\}$, so $|A_S| = \eta_S n$. Define the weights

$$w_i := \frac{a^*_S}{\eta_S}, \qquad i \in A_S.$$

Run the weighted Guruswami-Sudan algorithm (Theorem 7.1) with these weights. We claim that every codeword $c \in C \setminus \{h_1, \ldots, h_m\}$ with $\text{agree}(c, r) \geq \beta$ satisfies the decoding condition (6).

First, the sum of squared weights is

$$\sum_{i=1}^{n} w_i^2 = \sum_{S \subseteq [m]} \eta_S n \cdot \frac{(a^*_S)^2}{\eta_S^2} = \sum_{S \subseteq [m]} \frac{(a^*_S)^2 n}{\eta_S} = \alpha n,$$

so the RHS of (6) is αn. For a codeword c, define the fractional agreements on each block by

$$a_S := \frac{1}{n} \text{Agree}_S(c, r) = \frac{1}{n} \sum_{i \in A_S} \mathbf{1}_{c_i = r_i}.$$

Then $\sum_S a_S = \text{agree}(c, r) \geq \beta > \beta^*$. For each $j \in [m]$, on coordinates $i \in A_S$ with $j \in S$ we have $h_{j,i} = r_i$, so $c_i = r_i$ implies $c_i = h_{j,i}$. Thus $\sum_{S \ni j} a_S \leq \text{agree}(c, h_j) \leq \alpha$. The box constraints $0 \leq a_S \leq \eta_S$ hold by definition. So a satisfies all of the linear constraints of Optimization 5.1 with $c^T a \geq \beta > \beta^* = c^T a^*$.

The weighted agreement of c with r is

$$\sum_{i:\, c_i = r_i} w_i = n \sum_{S \subseteq [m]} a_S \cdot \frac{a^*_S}{\eta_S} = \frac{n}{2} \langle \nabla f(a^*), a \rangle = n\alpha + \frac{n}{2} \langle \nabla f(a^*), a - a^* \rangle,$$

where we used the fact that $\nabla f(a^*) = (2a^*_S/\eta_S)_{S \subseteq [m]}$, and $\frac{1}{2}\langle \nabla f(a^*), a^* \rangle = f(a^*) = \alpha$.

Let $\delta_0 := \beta - \beta^* > 0$ be the decoding gap. Since $c^T(a - a^*) \geq \delta_0$, Lemma 6.3, applied with the Slater point $z = 0$, gives

$$\langle \nabla f(a^*), a - a^* \rangle \geq \frac{\delta_0 \alpha}{\beta^*}.$$

Therefore,

$$\sum_{i:\, c_i = r_i} w_i \geq n\alpha + \frac{n\delta_0 \alpha}{2\beta^*}.$$

Set the interpolation accuracy to

$$\epsilon_{\text{alg}} := \frac{n\delta_0 \alpha}{4\beta^*}.$$

Since $w_{\max} \leq 1$, the weighted agreement exceeds $n\alpha + \epsilon_{\text{alg}} w_{\max}$, as required by (6). The algorithm runs in time $\text{poly}(n, q, 1/\epsilon_{\text{alg}}) = \text{poly}(n, q, 1/\delta_0) = \text{poly}(n, q, 1/\delta)$. Finally, add H_β to the output of the weighted Guruswami–Sudan algorithm and remove duplicates. □

References

[AGL24] Omar Alrabiah, Venkatesan Guruswami, and Ray Li. "Randomly Punctured Reed-Solomon Codes Achieve List-Decoding Capacity over Linear-Sized Fields". In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC)*. 2024, pp. 1458–1469. DOI: 10.1145/3618260.3649634.

[BGM23] Joshua Brakensiek, Sivakanth Gopi, and Visu Makam. "Generic Reed-Solomon Codes Achieve List-Decoding Capacity". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC)*. 2023, pp. 1488–1501. DOI: 10.1145/3564246.3585128.

[BKR10] Eli Ben-Sasson, Swastik Kopparty, and Jaikumar Radhakrishnan. "Subspace Polynomials and Limits to List Decoding of Reed-Solomon Codes". In: *IEEE Transactions on Information Theory* 56.1 (2010), pp. 113–120. DOI: 10.1109/TIT.2009.2034780.

[Bra+26] Joshua Brakensiek et al. *Algorithmic List Decoding of Reed–Solomon Codes up to Capacity*. 2026. arXiv: 2609.08005 [cs.IT]. URL: https://arxiv.org/abs/2609.08005.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[CZ25] Yeyuan Chen and Zihan Zhang. "Explicit Folded Reed-Solomon and Multiplicity Codes Achieve Relaxed Generalized Singleton Bounds". In: *Proceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC)*. 2025.

[DP09] Devdatt P. Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. 1st ed. Cambridge University Press, June 2009. ISBN: 978-0-521-88427-3. DOI: 10.1017/CBO9780511581274. URL: https://www.cambridge.org/core/product/identifier/9780511581274/type/book (visited on 08/25/2025).

[Dvi+13] Zeev Dvir et al. "Extensions to the Method of Multiplicities, with Applications to Kakeya Sets and Mergers". In: *SIAM Journal on Computing* 42.6 (2013), pp. 2305–2328. DOI: 10.1137/100783704.

[Eli57] Peter Elias. "List Decoding for Noisy Channels". In: *IRE WESCON Convention Record, Part 2*. 1957, pp. 94–104.

[GR08] Venkatesan Guruswami and Atri Rudra. "Explicit Codes Achieving List Decoding Capacity: Error-Correction with Optimal Redundancy". In: *IEEE Transactions on Information Theory* 54.1 (2008), pp. 135–150. DOI: 10.1109/TIT.2007.912263.

[GS02a] Venkatesan Guruswami and Igor Shparlinski. "Unconditional Proof of Tightness of Johnson Bound". In: (2002). DOI: 10.5555/644108.644232.

[GS02b] Venkatesan Guruswami and Madhu Sudan. "Decoding Concatenated Codes using Soft Information". In: *Proceedings of the 17th IEEE Annual Conference on Computational Complexity (CCC)*. 2002, pp. 148–157. DOI: 10.1109/CCC.2002.1004350.

[GS99] Venkatesan Guruswami and Madhu Sudan. "Improved Decoding of Reed-Solomon and Algebraic-Geometry Codes". In: *IEEE Transactions on Information Theory* 45.6 (1999), pp. 1757–1767. DOI: 10.1109/18.782097.

[Gur02] Venkatesan Guruswami. "Limits to List Decodability of Linear Codes". In: *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*. 2002, pp. 802–811. DOI: 10.1145/509907.510022.

[Gur04] Venkatesan Guruswami. *List Decoding of Error-Correcting Codes (Winning Thesis of the 2002 ACM Doctoral Dissertation Competition)*. 2004. ISBN: 978-3-540-24051-8. DOI: 10.1007/b104335.

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. "Unbalanced Expanders and Randomness Extractors from Parvaresh-Vardy Codes". In: *Journal of the ACM* 56.4 (2009), 20:1–20:34. DOI: 10.1145/1538902.1538904.

[GZ23] Zeyu Guo and Zihan Zhang. "Randomly Punctured Reed-Solomon Codes Achieve the List Decoding Capacity over Polynomial-Size Alphabets". In: *Proceedings of the 64th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 2023, pp. 164–176. DOI: 10.1109/FOCS57990.2023.00019.

[HKS26] Prahladh Harsha, Mrinal Kumar, and Ramprasad Saptharishi. *Reed–Solomon Codes at Capacity: Algorithmic List-Decoding and Proximity Gaps*. 2026. arXiv: 2610.08610 [cs.CC]. URL: https://arxiv.org/abs/2610.08610.

[Joh62] S. Johnson. "A New Upper Bound for Error-Correcting Codes". In: *IRE Transactions on Information Theory* 8.3 (Apr. 1962), pp. 203–207. ISSN: 2168-2712. DOI: 10.1109/TIT.1962.1057714.

[KL95] Gil Kalai and Nathan Linial. "On the Distance Distribution of Codes". In: *IEEE Transactions on Information Theory* 41.5 (1995), pp. 1467–1472. DOI: 10.1109/18.412711.

[Kop+23] Swastik Kopparty et al. "Improved Decoding of Folded Reed-Solomon and Multiplicity Codes". In: *SIAM Journal on Computing* 52.2 (2023), pp. 440–478. DOI: 10.1137/20M1370215.

[PV05] Farzad Parvaresh and Alexander Vardy. "Correcting Errors Beyond the Guruswami-Sudan Bound in Polynomial Time". In: *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 2005, pp. 285–294. DOI: 10.1109/FOCS.2005.29.

[Sri25] Shashank Srivastava. "Improved list size for folded reed-solomon codes". In: *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM. 2025, pp. 2040–2050.

[STV01] Madhu Sudan, Luca Trevisan, and Salil Vadhan. "Pseudorandom Generators without the XOR Lemma". In: *Journal of Computer and System Sciences* 62.2 (2001), pp. 236–266. DOI: 10.1006/jcss.2000.1730.

[SU05] Ronen Shaltiel and Christopher Umans. "Simple Extractors for All Min-Entropies and a New Pseudorandom Generator". In: *Journal of the ACM* 52.2 (2005), pp. 172–216. DOI: 10.1145/1059513.1059516.

[Sud97] Madhu Sudan. "Decoding of Reed-Solomon Codes beyond the Error-Correction Bound". In: *Journal of Complexity* 13.1 (1997), pp. 180–193. DOI: 10.1006/jcom.1997.0439.

[TU12] Amnon Ta-Shma and Christopher Umans. "Better Condensers and New Extractors from Parvaresh-Vardy Codes". In: *Proceedings of the 27th Conference on Computational Complexity (CCC)*. 2012, pp. 309–315. DOI: 10.1109/CCC.2012.25.

[Uma03] Christopher Umans. "Pseudo-random Generators for All Hardnesses". In: *Journal of Computer and System Sciences* 67.2 (2003), pp. 419–440. DOI: 10.1016/S0022-0000(03)00046-1.

A Alternate Decoding via Factored Interpolation

In this section, we present an alternate algorithm for list decoding Reed-Solomon codes with hints. While Section 7 uses the weighted Guruswami-Sudan algorithm, this approach instead factors the hint polynomial directly into the interpolation polynomial.

A.1 Decoding from one hint

Recall the setup from Section 7: let C be a Reed–Solomon code of length n over $\mathbb{F}_q$ ($q \geq n$) with maximum agreement α (so the degree bound k satisfies $k-1 = \alpha n$). Given a received word $r = (r_1, \ldots, r_n)$ and a single hint $h \in C$ with $\mathrm{agree}(h, r) = \eta$, we wish to find all codewords with agreement at least β with r.

We partition coordinates into

$$U = \{i \in [n] : h(x_i) = r_i\}, \qquad V = [n] \setminus U,$$

so $|U| = \eta n$ and $|V| = (1-\eta)n$. The main insight behind this version of the decoding algorithm is that the hint polynomial $h(X)$ can appear explicitly as a factor of the interpolation polynomial in the standard Guruswami–Sudan algorithm, providing free vanishing at coordinates in U.

Algorithm. Fix $c \in [0, 1]$ to be optimized later. Let M be a positive integer, and assume $cM \in \mathbb{Z}$.

1. **Interpolate.** Find a nonzero bivariate polynomial $Q(X, Y)$ of $(1, k-1)$-weighted degree at most D_Q vanishing with multiplicity M at each (x_i, r_i) for $i \in V$, and multiplicity $M(1-c)$ at each (x_i, r_i) for $i \in U$.
2. **Root-find.** Output all polynomials $f(X)$ of degree $< k$ with $(Y - f(X)) \mid Q(X, Y)$ and $\mathrm{agree}(f, r) \geq \beta$.

Theorem A.1. *Every codeword $f \neq h$ with $\mathrm{agree}(f, r) \geq \beta$ is output, provided*

$$\beta \;>\; c\alpha + \sqrt{\alpha\big((1-\eta) + (1-c)^2\eta\big)}.$$

Proof. Define $P(X, Y) = (Y - h(X))^{cM} \cdot Q(X, Y)$.

The proof is essentially identical to the proof of the original Guruswami–Sudan algorithm, with the only new ingredient being the different vanishing multiplicities on U and V.

Multiplicity. For $i \in V$, the polynomial Q vanishes to order M, so P does as well. For $i \in U$, we have $h(x_i) = r_i$, so $(Y - h(X))^{cM}$ vanishes to order cM at (x_i, r_i), and Q vanishes to order $M(1-c)$. The product vanishes to order $cM + M(1-c) = M$.

Degree. Since $h(X)$ has degree $\leq k-1$, the polynomial $Y - h(X)$ has $(1, k-1)$-weighted degree $k-1$. Thus, P has weighted degree at most $cM(k-1) + D_Q$.

For a codeword f with $\mathrm{agree}(f, r) \geq \beta$, the univariate polynomial

$$R(X) = (f(X) - h(X))^{cM} \cdot Q(X, f(X))$$

has degree at most $cM(k-1)+D_Q$ and vanishes to order $\geq M$ at the $\geq \beta n$ points where $f(x_i) = r_i$.

The interpolation cost for Q is approximately $\frac{M^2 n}{2}((1-\eta) + (1-c)^2\eta)$, and the monomial count at weighted degree D_Q is approximately $D_Q^2/(2(k-1))$. Setting these equal gives

$$D_Q = M\sqrt{n(k-1)((1-\eta) + (1-c)^2\eta)}.$$

Thus, $R \equiv 0$ whenever $M\beta n > cM(k-1)+D_Q$, which simplifies to $\beta > c\alpha+\sqrt{\alpha((1-\eta) + (1-c)^2\eta)}$ (using $k-1 = \alpha n$).

Since $f \neq h$, the factor $(f(X)-h(X))^{cM}$ is a nonzero polynomial. Thus, $R \equiv 0$ implies $Q(X, f(X)) \equiv 0$, i.e., $(Y - f(X)) \mid Q(X, Y)$. □

We now find the threshold β^*_{fact} by minimizing $c\alpha + \sqrt{\alpha((1-\eta)+(1-c)^2\eta)}$ over $c \in [0,1]$.

We have

$$\beta'(c) = \alpha - \frac{\sqrt{\alpha}\,\eta(1-c)}{\sqrt{(1-\eta)+(1-c)^2\eta}}.$$

Since

$$\beta''(c) = \frac{\sqrt{\alpha}\,\eta(1-\eta)}{\big((1-\eta)+(1-c)^2\eta\big)^{3/2}} > 0,$$

$\beta(c)$ is strictly convex, so its minimum is either at $c = 0$ or at the unique interior critical point.

If $\eta \le \sqrt{\alpha}$, then

$$\beta'(0) = \alpha - \eta\sqrt{\alpha} = \sqrt{\alpha}(\sqrt{\alpha} - \eta) \ge 0,$$

so the minimum is attained at $c = 0$, giving $\beta^* = \sqrt{\alpha}$.

If $\eta > \sqrt{\alpha}$, then $\beta'(0) < 0$, so the minimum occurs at the unique critical point satisfying

$$(1-c)^2 = \frac{\alpha(1-\eta)}{\eta(\eta-\alpha)}.$$

Substituting this value yields

$$\beta^* = \alpha + \sqrt{\frac{\alpha(1-\eta)(\eta-\alpha)}{\eta}}.$$

This matches the first two regimes of Theorem 3.3. For the third regime, note that the triangle inequality implies that when $\beta > 1 - \eta + \alpha$, the only codeword with agreement $\ge \beta$ is h itself. Hence, list-decoding to agreement at least β is trivial. To recover the third regime, we simply compare the expression for β^* in the $\eta > \sqrt{\alpha}$ regime with $1 - \eta + \alpha$, taking the smaller of the two. We find that $1 - \eta + \alpha$ is smaller so long as $\eta > \eta_{\text{crit}}$, where η_{crit} is as defined in Theorem 3.3.

A.2 Generalization to Multiple Hints

It is natural to ask whether this factored interpolation approach generalizes to multiple hints. For two hints h_1, h_2, we could introduce multipliers $c_1, c_2 \ge 0$ (with $c_1 + c_2 \le 1$) and explicitly factor $(Y - h_1(X))^{c_1 M}(Y - h_2(X))^{c_2 M}$ out of the interpolant. We partition the coordinates based on which hints agree with the received word r: let $\eta_\emptyset$ be the fraction where neither agrees, η_1 where only h_1 agrees, η_2 where only h_2 agrees, and η_{12} where both agree.

The resulting threshold from optimizing the multipliers c_1, c_2 would be

$$\begin{aligned} \beta^*_{\text{fact}} = \min_{c_1,c_2} \quad & \alpha(c_1+c_2) + \sqrt{\alpha\left[\eta_\emptyset + \eta_1(1-c_1)^2 + \eta_2(1-c_2)^2 + \eta_{12}(1-c_1-c_2)^2\right]} \\ \text{subject to} \quad & c_1 \ge 0, \quad c_2 \ge 0, \\ & c_1 + c_2 \le 1. \end{aligned}$$

This minimization is reminiscent of the Lagrangian dual of Optimization 5.1: the variables c_1 and c_2 act like multipliers for the two capacity constraints. However, this optimization does not dualize the box constraints $a_S \le \eta_S$, which leads to a gap between β^*_{fact} and the original (primal) optimum, β^*.

As an example, consider an instance with two hints where $\alpha = 0.5$ and

$$\eta_\emptyset = 0, \qquad \eta_1 = 0.60, \qquad \eta_2 = 0.01, \qquad \eta_{12} = 0.39.$$

Solving the primal formulation (maximizing over variables a_S) yields a primal optimum of $\beta^* = 0.51$, achieved at $a_1^* = 0.11, a_2^* = 0.01, a_{12}^* = 0.39, a_\emptyset^* = 0$. Note that in this solution, the box constraint for the second hint ($a_2 \le \eta_2$) is tight. However, solving for β_{fact}^* yields a global optimum at $c_1^* \approx 0.8984$ and $c_2^* = 0$, giving $\beta_{\text{fact}}^* \approx 0.5497$. The gap arises because the factored formulation dualizes the capacity constraints (via multipliers c_1, c_2) but not the box constraints. More fundamentally, a global factor such as $Y - h_j(X)$ cannot isolate just the coordinates where hint h_j is uniquely correct: any such factor acts on every coordinate where h_j agrees, regardless of whether other hints also agree there. This lack of coordinate-level selectivity is exactly what the weighted Guruswami–Sudan algorithm provides, and it is why a purely factored approach cannot close the gap for multiple hints.