Foreword

This chapter is based on lecture notes from coding theory courses taught by Venkatesan Guruswami at University at Washington and CMU; by Atri Rudra at University at Buffalo, SUNY and by Madhu Sudan at MIT.

This version is dated **April 28, 2013**. For the latest version, please go to

http://www.cse.buffalo.edu/ atri/courses/coding-theory/book/

The material in this chapter is supported in part by the National Science Foundation under CAREER grant CCF-0844796. Any opinions, findings and conclusions or recomendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

Chapter 7

Bridging the Gap Between Shannon and Hamming: List Decoding

In Section 6.4, we made a qualitative comparison between Hamming and Shannon's world. We start this chapter by doing a more quantitative comparison between the two threads of coding theory. In Section 7.2 we introduce the notion of list decoding, which potentially allows us to go beyond the (quantitative) results of Hamming and approach those of Shannon's. Then in Section 7.3, we show how list decoding allows us to go beyond half the distance bound for any code. Section 7.4 proves the optimal trade-off between rate and fraction of correctable errors via list decoding. Finally, in Section 7.5, we formalize why list decoding could be a useful primitive in practical communication setups.

7.1 Hamming versus Shannon: part II

Let us compare Hamming and Shannon theories in terms of the asymptotic bounds we have seen so far (recall rate $R = \frac{k}{n}$ and relative distance $\delta = \frac{d}{n}$).

- Hamming theory: Can correct $\leq \frac{\delta}{2}$ fraction of worse case errors for codes of relative distance δ. By the Singleton bound (Theorem 4.3.1),

  $$\delta \leq 1 - R,$$

  which by Proposition 1.4.1 implies that p fraction of errors can be corrected has to satisfy

  $$p \leq \frac{1-R}{2}.$$

  The above can be achieved via efficient decoding algorithms for example for Reed-Solomon codes (we'll see this later in the book).

- Shannon theory: In $q\mathrm{SC}_p$, for $0 \leq p < 1 - 1/q$, we can have reliable communication with $R < 1 - H_q(p)$. It can be shown that

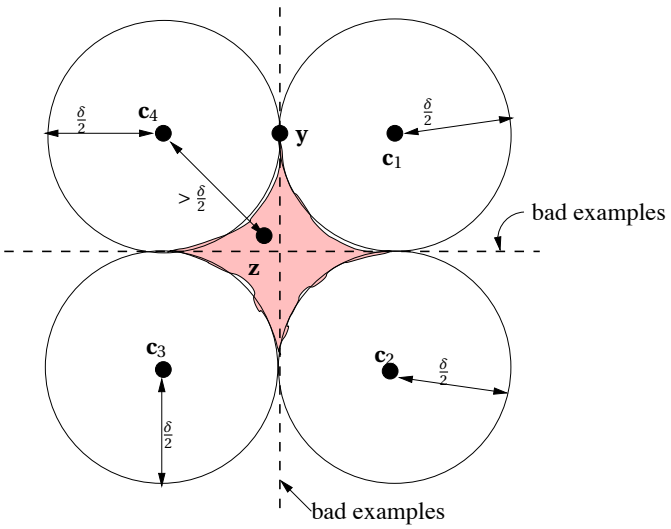

Figure 7.1: In this example vectors are embedded into Euclidean space such that the Euclidean distance between two mapped points is the same as the Hamming distance between vectors. $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4$ are codewords. The dotted lines contain the "bad examples," that is, the received words for which unique decoding is not possible.

1. $1 - H_q(p) \leq 1 - p$ (this is left as an exercise); and
2. $1 - H_q(p) \geq 1 - p - \varepsilon$, for large enough q– in particular, $q = 2^{\Omega(1/\varepsilon)}$ (Proposition 3.3.2).

Thus, we can have reliable communication with $p \sim 1 - R$ on $q\mathrm{SC}_p$ for large enough q.

There is a gap between Shannon and Hamming world: one can correct twice as many errors in Shannon's world. One natural question to ask is whether we can somehow "bridge" this gap. Towards this end, we will now re-visit the the bad example for unique decoding (Figure 1.3) and consider an extension of the bad example as shown in Figure 7.1.

Recall that $\mathbf{y}$ and the codewords $\mathbf{c}_1$ and $\mathbf{c}_2$ form the bad example for unique decoding that we have already seen before. Recall that for this particular received word we can not do error recovery by unique decoding since there are two codewords $\mathbf{c}_1$ and $\mathbf{c}_2$ having the same distance $\frac{\delta}{2}$ from vector $\mathbf{y}$. On the other hand, the received word $\mathbf{z}$ has an unique codeword $\mathbf{c}_1$ with distance $p > \frac{\delta}{2}$. However, unique decoding does not allow for error recovery from $\mathbf{z}$. This is because by definition of unique decoding, the decoder must be able to recover from *every* error pattern (with a given Hamming weight bound). Thus, by Proposition 1.4.1, the decoded codeword cannot have relative Hamming distance larger than $\delta/2$ from the received word. In this example, because of the received word $\mathbf{y}$, unique decoding gives up on the opportunity to decode $\mathbf{z}$.

Let us consider the example in Figure 7.1 for the binary case. It can be shown that the number of vectors in dotted lines is insignificant compared to volume of shaded area (for large enough block length of the code). The volume of all Hamming balls of radius $\frac{\delta}{2}$ around all the

2^k codewords is roughly equal to:

$$2^k 2^{nH(\frac{\delta}{2})},$$

which implies that the volume of the shaded area (without the dotted lines) is approximately equal to:

$$2^n - 2^k 2^{nH(\frac{\delta}{2})}.$$

In other words, the volume when expressed as a fraction of the volume of the ambient space is roughly:

$$1 - 2^{-n(1-H(\frac{\delta}{2})-R)}, \tag{7.1}$$

where $k = Rn$ and by the Hamming bound (Theorem 1.3) $R \leq 1 - H(\frac{\delta}{2})$. If $R < 1 - H(\frac{\delta}{2})$ then second term of (7.1) is very small. Therefore the number of vectors in shaded area (without the bad examples) is almost all of the ambient space. Note that by the stringent condition on unique decoding none of these received words can be decoded (even though for such received words there is a unique closest codeword). Thus, in order to be able to decode such received vectors, we need to relax the notion of unique decoding. We will consider such a relaxation called *list decoding* next.

7.2 List Decoding

The new notion of decoding that we will discuss is called *list decoding* as the decoder is allowed to output a list of answers. We now formally define (the combinatorial version of) list decoding:

Definition 7.2.1. Given $0 \leq \rho \leq 1, L \geq 1$, a code $C \subseteq \Sigma^n$ is (ρ, L)-list decodable if for every received word $\mathbf{y} \in \Sigma^n$,

$$\left|\{c \in C | \Delta(\mathbf{y}, c) \leq \rho n\}\right| \leq L$$

Given an error parameter ρ, a code C and a received word $\mathbf{y}$, a list-decoding algorithm should output all codewords in C that are within (relative) Hamming distance ρ from $\mathbf{y}$. Note that if the fraction of errors that occurred during transmission is at most ρ then the transmitted codeword is *guaranteed* to be in the output list. Further, note that if C is (ρ, L)-list decodable then the algorithm will always output at most L codewords for any received word. In other words, for efficient list-decoding algorithm, L should be a polynomial in the block length n (as otherwise the algorithm will have to output a super-polynomial number of codewords and hence, cannot have a polynomial running time). Thus, the restriction of L being at most some polynomial in n is an *a priori* requirement enforced by the fact that we are interested in efficient polynomial time decoding algorithms. Another reason for insisting on a bound on L is that otherwise the decoding problem can become trivial: for example, one can output all the codewords in the code. Finally, it is worthwhile to note that one can always have an exponential time list-decoding algorithm: go through all the codewords in the code and pick the ones that are within ρ (relative) Hamming distance of the received word.

Note that in the communication setup, we need to recover the transmitted message. In such a scenario, outputting a list might not be useful. There are two ways to get around this "problem":

1. Declare a decoding error if list size > 1. Note that this generalizes unique decoding (as when the number of errors is at most half the distance of the code then there is a unique codeword and hence, the list size will be at most one). However, the gain over unique decoding would be substantial only if for most error patterns (of weight significantly more than half the distance of the code) the output list size is at most one. Fortunately, it can be show that:

   - For random codes, with high probability, for most error patterns, the list size is at most one. In other words, for *most* codes, we can hope to see a gain over unique decoding. The proof of this fact follows from Shannon's proof for the capacity for qSC: the details are left as an exercise.
   - In Section 7.5, we show that the above behavior is in fact general: i.e. for *any* code (over a large enough alphabet) it is true that with high probability, for most error patterns, the list size is at most one.

   Thus, using this option to deal with multiple answers, we still deal with worse case errors but can correct more error patterns than unique decoding.

2. If the decoder has access to some side information, then it can use that to prune the list. Informally, if the worst-case list size is L, then the amount of extra information one needs is $O(\log L)$. This will effectively decrease[1] the dimension of the code by $O(\log L)$, so if L is small enough, this will have negligible effect on the rate of the code. There are also application (especially in complexity theory) where one does not really care about the rate being the best possible.

Recall that Proposition 1.4.1 implies that $\delta/2$ is the maximum fraction of errors correctable with unique decoding. Since list decoding is a relaxation of unique decoding, it is natural to wonder

Question 7.2.1. *Can we correct more than $\delta/2$ fraction of errors using list decoding?*

and if so

Question 7.2.2. *What is the maximum fraction of errors correctable using list decoding?*

In particular, note that the intuition from Figure 7.1 states that the answer to Question 7.2.1 should be yes.

[1] Note that side information effectively means that not all possible vectors are valid messages.

7.3 Johnson Bound

In this section, we will indeed answer Question 7.2.1 in the affirmative by stating a bound due to Johnson. To setup the context again, recall that Proposition 1.4.1 implies that any code with relative distance δ is $(\delta/2, 1)$-list decodable.

Notice that if we can show a code for some $e > \left\lfloor \frac{d-1}{2} \right\rfloor$ is $(e/n, n^{O(1)})$-list decodable, then (combinatorially) it is possible to list decode that code up to e errors. We'll show by proving the Johnson bound that this is indeed the case for any code.

Theorem 7.3.1 (Johnson Bound)**.** *Let $C \subseteq [q]^n$ be a code of distance d. If $\rho < J_q\left(\frac{d}{n}\right)$, then C is a (ρ, qdn)-list decodable code, where the function $J_q(\delta)$ is defined as*

$$J_q(\delta) = \left(1 - \frac{1}{q}\right)\left(1 - \sqrt{1 - \frac{q\delta}{q-1}}\right).$$

Proof (for $q = 2$). The proof technique that we will use has a name: double counting. The main idea is to count the same quantity in two different ways to get both an upper and lower bound on the same quantity. These bounds then imply an inequality and we will derive our desired bound from this inequality.

We have to show that for every binary code $C \subseteq \{0,1\}^n$ with distance d (i.e. for every $\mathbf{c}_1 \neq \mathbf{c}_2 \in C$, $\Delta(\mathbf{c}_1, \mathbf{c}_2) \geq d$) and every $\mathbf{y} \in \{0,1\}^n$, $|B(\mathbf{y}, e) \bigcap C| \leq 2dn$.

Fix arbitrary C and $\mathbf{y}$. Let $\mathbf{c}_1, \ldots, \mathbf{c}_M \in B(\mathbf{y}, e)$. We need to show that $M \leq 2dn$. Define $\mathbf{c}'_i = \mathbf{c}_i - \mathbf{y}$ for $1 \leq i \leq M$. Then we have the following:

(i) $wt(\mathbf{c}'_i) \leq e$ for $1 \leq i \leq M$ because $\mathbf{c}_i \in B(\mathbf{y}, e)$.

(ii) $\Delta(\mathbf{c}'_i, \mathbf{c}'_j) \geq d$ for every $i \neq j$ because $\Delta(\mathbf{c}_i, \mathbf{c}_j) \geq d$.

Define

$$S = \sum_{i<j} \Delta(\mathbf{c}'_i, \mathbf{c}'_j).$$

We will prove both an upper and a lower bound on S from which we will extract the required upper bound on M. Then from (ii) we have

$$S \geq \binom{M}{2} d \tag{7.2}$$

Consider the $n \times M$ matrix $(\mathbf{c}'^T_1, \cdots, \mathbf{c}'^T_M)$. Define m_i as the number of 1's in the i-th row for $1 \leq i \leq n$. Then the i-th row of the matrix contributes the value $m_i(M - m_i)$ to S because this is the number of 0-1 pairs in that row. (Note that each such pair contributes one to S.) This implies that

$$S = \sum_{i=1}^{n} m_i(M - m_i). \tag{7.3}$$

Define

$$\bar{e} = \sum_i \frac{m_i}{M}.$$

Note that

$$\sum_{i=1}^{n} m_i = \sum_{j=1}^{M} wt(\mathbf{c}_i) \le eM,$$

where the inequality follows From (i) above. Thus, we have

$$\bar{e} \le e.$$

Using the Cauchy-Schwartz inequality (i.e., $\langle \mathbf{x}, \mathbf{y} \rangle \le ||\mathbf{x}|| \cdot ||\mathbf{y}||$ for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$) by taking $\mathbf{x} = (m_1, \cdots, m_n)$, $\mathbf{y} = (1/n, \cdots, 1/n)$, we have

$$\left(\frac{\sum_{i=1}^{n} m_i}{n} \right)^2 \le \left(\sum_{i=1}^{n} m_i^2 \right) \frac{1}{n}. \tag{7.4}$$

Thus, from (7.3)

$$S = \sum_{i=1}^{n} m_i (M - m_i) = M^2 \bar{e} - \sum_{i=1}^{n} m_i^2 \le M^2 \bar{e} - \frac{(M\bar{e})^2}{n} = M^2 (\bar{e} - \frac{\bar{e}^2}{n}), \tag{7.5}$$

where the inequality follows from (7.4). By (7.2) and (7.5),

$$M^2 \left(\bar{e} - \frac{\bar{e}^2}{n} \right) \ge \frac{M(M-1)}{2} d,$$

which implies that

$$\begin{aligned} M \le \frac{dn}{dn - 2n\bar{e} + 2\bar{e}^2} = \frac{2dn}{2dn - n^2 + n^2 - 4n\bar{e} + 4\bar{e}^2} &= \frac{2dn}{(n - 2\bar{e})^2 - n(n-2d)} \\ &\le \frac{2dn}{(n-2e)^2 - n(n-2d)}, \end{aligned} \tag{7.6}$$

where the last inequality follows from the fact that $\bar{e} \le e$. Then from

$$\frac{e}{n} < \frac{1}{2} \left(1 - \sqrt{1 - \frac{2d}{n}} \right),$$

we get

$$n - 2e > \sqrt{n(n-2d)}.$$

In other words

$$(n-2e)^2 > n(n-2d).$$

Thus, $(n-2e)^2 - n(n-2d) \ge 1$ because n, e are all integers and therefore, from (7.6), we have $M \le 2dn$ as desired. □

Next, we prove the following property of the function $J_q(\cdot)$, which along with the Johnson bound answers Question 7.2.1 in the affirmative.

Lemma 7.3.2. *Let $q \geq 2$ be an integer and let $0 \leq x \leq 1 - \frac{1}{q}$. Then the following inequalities hold:*

$$J_q(x) \geq 1 - \sqrt{1-x} \geq \frac{x}{2},$$

where the second inequality is tight for $x > 0$.

Proof. We start with by proving the inequality

$$\left(1 - \frac{1}{q}\right)\left(1 - \sqrt{1 - \frac{xq}{q-1}}\right) \geq 1 - \sqrt{1-x}.$$

Indeed, both the LHS and RHS of the inequality are zero at $x = 0$. Further, it is easy to check that the derivatives of the LHS and RHS are $\frac{1}{\sqrt{1-\frac{xq}{q-1}}}$ and $\frac{1}{\sqrt{1-x}}$ respectively. The former is always larger than the latter quantity. This implies that the LHS increases more rapidly than the RHS, which in turn proves the required inequality.

The second inequality follows from the subsequent relations. As $x \geq 0$,

$$1 - x + \frac{x^2}{4} \geq 1 - x,$$

which implies that

$$\left(1 - \frac{x}{2}\right)^2 \geq 1 - x,$$

which in turn implies the required inequality. (Note that the two inequalities above are strict for $x > 0$, which implies that $1 - \sqrt{1-x} > x/2$ for every $x > 0$, as desired.) □

Theorem 7.3.1 and Lemma 7.3.2 imply that for *any* code, list decoding can potentially correct strictly more errors than unique decoding in polynomial time, as long as q is at most some polynomial in n (which will be true of all the codes that we discuss in this book). This answers Question 7.2.1 in the affirmative. See Figure 7.2 for an illustration of the gap between the Johnson bound and the unique decoding bound.

Theorem 7.3.1 and Lemma 7.3.2 also implies the following "alphabet-free" version of the Johnson bound.

Theorem 7.3.3 (Alphabet-Free Johnson Bound)**.** *If $e \leq n - \sqrt{n(n-d)}$, then any code with distance d is $(e/n, qnd)$-list decodable for all the q.*

A natural question to ask is the following:

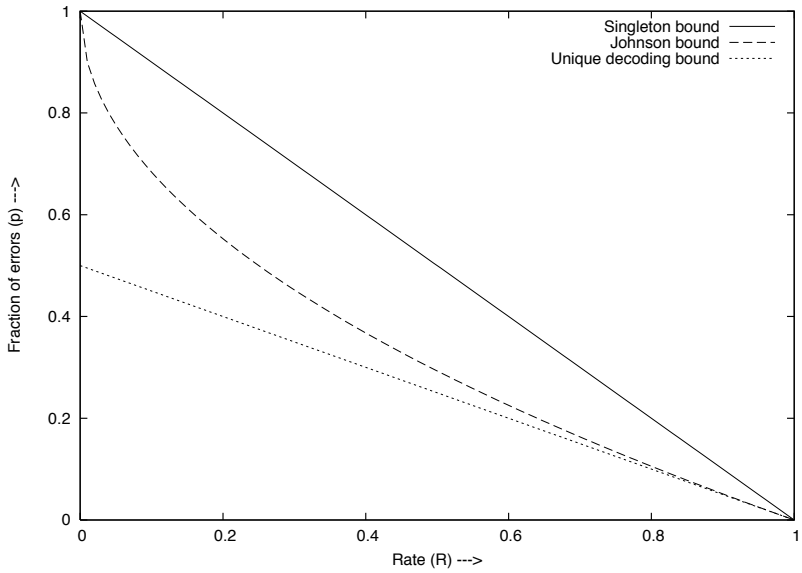

Figure 7.2: The trade-off between rate R and the fraction of errors that can be corrected. $1-\sqrt{R}$ is the trade-off implied by the Johnson bound. The bound for unique decoding is $(1-R)/2$ while $1-R$ is the Singleton bound (and the list decoding capacity for codes over large alphabets).

Question 7.3.1. *Is the Johnson bound tight?*

The answer is yes in the sense that there exist linear codes with relative distance δ such that we can find Hamming ball of radius larger than $J_q(\delta)$ with super-polynomially many codewords. On the other hand, in the next section, we will show that, in some sense, it is not tight.

7.4 List-Decoding Capacity

In the previous section, we saw what can one achieve with list decoding in terms of distance of a code. In this section, let us come back to Question 7.2.2. In particular, we will consider the trade-off between rate and the fraction of errors correctable by list decoding. Unlike the case of unique decoding and like the case of BSC_p, we will be able to prove an optimal trade-off.

Next, we will prove the following result regarding the optimal trade-off between rate of a code and the fraction of errors that can be corrected via list decoding.

Theorem 7.4.1. *Let $q \geq 2$, $0 \leq p < 1 - \frac{1}{q}$, and $\varepsilon > 0$. Then the following holds for codes of large enough block length n:*

(i) If $R \leq 1 - H_q(p) - \varepsilon$, then there exists a $\left(p, O\left(\frac{1}{\varepsilon}\right)\right)$-list decodable code.

(ii) If $R > 1 - H_q(p) + \varepsilon$, every (ρ, L)-list decodable code has $L \geq q^{\Omega(n)}$.

Thus, the *List-decoding capacity*[2] is $1 - H_q(p)$ (where p is the fraction of errors). Further, this fully answers Question 7.2.2. Finally, note that this exactly matches capacity for $q\text{SC}_p$ and hence, list decoding can be seen as a bridge between Shannon's world and Hamming's world. The remarkable aspect of this result is that we bridge the gap between these worlds by allowing the decoder to output at most $O(1/\varepsilon)$ many codewords.

7.4.1 Proof of Theorem 7.4.1

We begin with the basic idea behind the proof of part (i) of the theorem.

As in Shannon's proof for capacity of BSC_p, we will use the probabilistic method (Section 3.2). In particular, we will pick a random code and show that it satisfies the required property with non-zero probability. In fact, we will show that a random code is (ρ, L)-list decodable with high probability as long as:

$$R \leq 1 - H_q(p) - \frac{1}{L}$$

The analysis will proceed by proving that probability of a "bad event" is small. "Bad event" means there exist messages $\mathbf{m}_0, \mathbf{m}_1, \cdots, \mathbf{m}_L \in [q]^{Rn}$ and a received code $\mathbf{y} \in [q]^n$ such that:

$$\Delta\big(C(\mathbf{m}_i), \mathbf{y}\big) \leq \rho n, \text{for every } 0 \leq i \leq L.$$

Note that if a bad event occurs, then the code is not a (ρ, L)-list decodable code. The probability of the occurrence of any bad event will then be calculated by an application of the union bound.

Next, we restate Theorem 7.4.1 and prove a stronger version of part (i). (Note that $L = \lceil \frac{1}{\varepsilon} \rceil$ in Theorem 7.4.2 implies Theorem 7.4.1.)

Theorem 7.4.2 (List-Decoding Capacity)**.** *Let $q \geq 2$ be an integer, and $0 < \rho < 1 - \frac{1}{q}$ be a real number.*

(i) Let $L \geq 1$ be an integer, then there exists an $\big(\rho, L\big)$-list decodable code with rate

$$R \leq 1 - H_q(\rho) - \frac{1}{L}$$

(ii) For every $\big(\rho, L\big)$ code of rate $1 - H_q(\rho) + \varepsilon$, it is necessary that $L \geq 2^{\Omega(\varepsilon n)}$.

Proof. We start with the proof of (i). Pick a code C at random where

$$|C| = q^k, \text{ where } k \leq \left(1 - H_q(\rho) - \frac{1}{L}\right) n.$$

That is, as in Shannon's proof, for every message $\mathbf{m}$, pick $C(\mathbf{m})$ uniformly and independently at random from $[q]^n$.

[2] Actually the phrase should be something like "capacity of worst case noise model under list decoding" as the capacity is a property of the channel. However, in the interest of brevity we will only use the term list-decoding capacity.

Given $\mathbf{y} \in [q]^n$, and $\mathbf{m}_0, \cdots, \mathbf{m}_L \in [q]^k$, the tuple $(\mathbf{y}, \mathbf{m}_0, \cdots, \mathbf{m}_L)$ defines a *bad event* if

$$C(\mathbf{m}_i) \in B(\mathbf{y}, \rho n), 0 \leq i \leq L.$$

Note that a code is (ρ, L)-list decodable if and only if there does not exist any bad event.

Fix $\mathbf{y} \in [q]^n$ and $\mathbf{m}_0, \cdots, \mathbf{m}_L \in [q]^k$.
Note that for fixed i, by the choice of C, we have:

$$\Pr[C(\mathbf{m}_i) \in B(\mathbf{y}, \rho n)] = \frac{Vol_q(\rho n, n)}{q^n} \leq q^{-n(1-H_q(\rho))}, \tag{7.7}$$

where the inequality follows from the upper bound on the volume of a Hamming ball (Proposition 3.3.1). Now the probability of a bad event given $(\mathbf{y}, \mathbf{m}_0, \cdots, \mathbf{m}_L)$ is

$$\Pr\left[\bigwedge_{i=0}^{L} C(\mathbf{m}_i) \in B(\mathbf{y}, \rho n)\right] = \prod_{0}^{L} \Pr[C(\mathbf{m}_i) \in B(\mathbf{y}, \rho n)] \leq q^{-n(L+1)(1-H_q(\rho))}, \tag{7.8}$$

where the equality follows from the fact that the random choices of codewords for distinct messages are independent and the inequality follows from (7.7). Then,

$$\begin{aligned}
\Pr[\text{ There is a bad event}] &\leq q^n \binom{q^k}{L+1} q^{-n(L+1)(1-H_q(\rho))} && (7.9)\\
&\leq q^n q^{Rn(L+1)} q^{-n(L+1)(1-H_q(\rho))} && (7.10)\\
&= q^{-n(L+1)[1-H_q(\rho)-\frac{1}{L+1}-R]} \\
&\leq q^{-n(L+1)[1-H_q(\rho)-\frac{1}{L+1}-1+H_q(\rho)+\frac{1}{L}]} && (7.11)\\
&= q^{-\frac{n}{L}} \\
&< 1
\end{aligned}$$

In the above, (7.9) follows by the union bound (Lemma 3.1.3) with (7.8) and by counting the number of $\mathbf{y}$'s (which is q^n), and the number of $L+1$ tuples (which is $\binom{q^k}{L+1}$). (7.10) follows from the fact that $\binom{a}{b} \leq a^b$ and $k = Rn$. (7.11) follows by assumption $R \leq 1 - H_q(\rho) - \frac{1}{L}$. The rest of the steps follow from rearranging and canceling the terms. Therefore, by the probabilistic method, there exists C such that it is (ρ, L)-list decodable.

Now we turn to the proof of part (ii). For this part, we need to show the existence of a $\mathbf{y} \in [q]^n$ such that $|C \cap B(\mathbf{y}, \rho n)|$ is exponentially large for every C of rate $R \geq 1 - H_q(\rho) + \varepsilon$. We will again use the probabilistic method to prove this result.

Pick $\mathbf{y} \in [q]^n$ uniformly at random. Fix $\mathbf{c} \in C$. Then

$$\begin{aligned}
\Pr[\mathbf{c} \in B(\mathbf{y}, \rho n)] &= \Pr[\mathbf{y} \in B(\mathbf{c}, \rho n)] \\
&= \frac{Vol_q(\rho n, n)}{q^n} && (7.12)\\
&\geq q^{-n(1-H_q(\rho))-o(n)}, && (7.13)
\end{aligned}$$

where (7.12) follows from the fact that $\mathbf{y}$ is chosen uniformly at random from $[q]^n$ and (7.13) follows by the lower bound on the volume of the Hamming ball (Proposition 3.3.1).

We have

$$\begin{aligned} E[|C \cap B(\mathbf{y}, \rho n)|] &= \sum_{\mathbf{c} \in C} E[\mathbb{1}_{\mathbf{c} \in B(\mathbf{y}, \rho n)}] && (7.14) \\ &= \sum_{\mathbf{c} \in C} \Pr[\mathbf{c} \in B(\mathbf{y}, \rho n)] \\ &\geq \sum_{\mathbf{c} \in C} q^{-n(1-H_q(\rho)+o(n))} && (7.15) \\ &= q^{n[R-1+H_q(\rho)-o(1)]} \\ &\geq q^{\Omega(\varepsilon n)} \\ & && (7.16) \end{aligned}$$

In the above, (7.14) follows by the linearity of expectation (Proposition 3.1.2), (7.15) follows from (7.13), and (7.16) follows by choice of R. Hence, by the probabilistic method, there exists $\mathbf{y}$ such that $|B(\mathbf{y}, \rho n) \cap C|$ is $q^{\Omega(n)}$, as desired. □

The above proof can be modified to work for random linear codes. In particular, one can show that with high probability, a random linear code is (ρ, L)-list decodable code as long as

$$R \leq 1 - H_q(\rho) - \frac{1}{\lceil \log_q(L+1) \rceil}. \tag{7.17}$$

The details are left as an exercise.

We now return to Question 7.3.1. Note that by the Singleton bound, the Johnson bound implies that for any code one can hope to list decode from about $p \leq 1 - \sqrt{R}$ fraction of errors. However, this trade-off between p and R is not tight. Note that Lemma 3.3.2 along with Theorem 7.4.1 implies that for large q, the list decoding capacity is $1 - R > 1 - \sqrt{R}$. Figure 7.2 plots and compares the relevant trade-offs.

Finally, we have shown that the list decoding capacity is $1 - H_q(p)$. However, we showed the existence of a code that achieves the capacity by the probabilistic method. This then raises the following question:

Question 7.4.1. *Do there exist explicit codes that achieve list decoding capacity?*

Also the only list decoding algorithm that we have seen so far is the brute force algorithm that checks every codeword to see if they need to be output. This also leads to the follow-up question

Question 7.4.2. *Can we achieve list decoding capacity with efficient list decoding algorithms?*

A more modest goal related to the above would be the following:

Question 7.4.3. *Can we design an efficient list decoding algorithm that can achieve the Johnson bound? In particular, can we efficiently list decode a code of rate R from $1-\sqrt{R}$ fraction of errors?*

7.5 List Decoding from Random Errors

In this section, we formalize the intuition we developed from Figure 7.1. In particular, recall that we had informally argued that for most error patterns we can correct beyond the $\delta/2$ bound for unique decoding (Proposition 1.4.1). Johnson bound (Theorem 7.3.1) tells us that one can indeed correct beyond $\delta/2$ fraction of errors. However, there are two shortcomings. The first is that the Johnson bounds tells us that the output list size is qdn but it does not necessarily imply that for most error patterns, there is unique by closest codewords (i.e. one can uniquely recover the transmitted codeword). In other words, Johnson bound is a "true" list decoding result and tells us nothing about the behavior of codes on the "average." The second aspect is that the Johnson bound holds for up to $1-\sqrt{1-\delta}$ fraction of errors. Even though it is more than $\delta/2$ for every $\delta > 0$, the bound e.g. is not say twice the unique decoding bound for every $\delta > 0$.

Next we show that for *any* code with relative distance δ (over a large enough alphabet size) for most error patterns, the output of a list decoder for any fraction of errors arbitrarily close to δ will have size one. In fact, the result is somewhat stronger: it show that even if one fixes the error locations *arbitrarily*, for most error patterns the output list size is one.

Theorem 7.5.1. *Let $\varepsilon > 0$ be a real and $q \geq 2^{\Omega(1/\varepsilon)}$ be an integer. Then the following is true for any $0 < \delta < 1 - 1/q$ and large enough n. Let $C \subseteq \{0,1,\ldots q-1\}^n$ be a code with relative distance δ and let $S \subseteq [n]$ such that $|S| = (1-\rho)n$, where $(0 < \rho \leq \delta - \varepsilon)$.*

Then, for all $\mathbf{c} \in C$ and all but a $q^{-\Omega(\varepsilon n)}$ fraction of error patterns, $\mathbf{e} \in \{0,1\ldots q-1\}^n$ such that

$$\mathbf{e}_S = \mathbf{0} \text{ and } wt(\mathbf{e}) = \rho n \tag{7.18}$$

the only codeword within Hamming distance ρn of $\mathbf{c}+\mathbf{e}$ is $\mathbf{c}$ itself.

For illustration of the kinds of error pattern we will deal with, see Figure 7.3.

Before we present the proof, we present certain corollaries (the proofs of which we leave as exercises). First the result above implies a similar result of the output list size being one for the following two random noise models: (i) uniform distribution over *all* error patterns of weight ρn and (ii) $q\mathrm{SC}_p$. In fact, we claim that the result also implies that any code with distance at

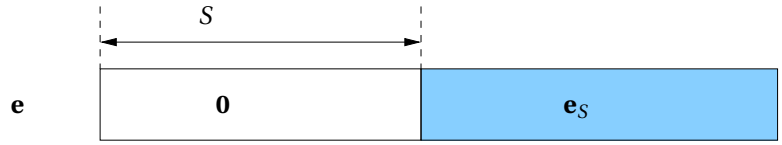

Figure 7.3: Illustration of the kind of error patterns we are trying to count.

least $p+\varepsilon$ allows for reliable communication over $q\mathrm{SC}_p$. (Contrast the $2p+\varepsilon$ distance that was needed for a similar result that was implied by Proposition 6.4.1.)

Finally, we present a lemma (the proof is left as an exercise) that will be crucial to the proof of Theorem 7.5.1.

Lemma 7.5.2. *Let be C be an $(n,k,d)_q$ code. If we fix the values in $n-d+1$ out of the n positions in a possible codeword, then at most one codeword in C can agree with the fixed values.*

Proof of Theorem 7.5.1. For the rest of the proof, fix a $\mathbf{c} \in C$. For notational convenience define $\mathscr{E}_S$ to be the set of all error patterns $\mathbf{e}$ such that $\mathbf{e}_S = 0$ and $wt(\mathbf{e}) = \rho n$. Note that as every error position has $(q-1)$ non-zero choices and there are ρn such positions in $[n] \setminus S$, we have

$$|\mathscr{E}_s| = (q-1)^{\rho n}. \tag{7.19}$$

Call an error pattern $\mathbf{e} \in \mathscr{E}_s$ as *bad* if there exists another codeword $\mathbf{c}' \neq \mathbf{c}$ such that

$$\triangle(\mathbf{c}', \mathbf{c}+\mathbf{e}) \leq \rho n.$$

Now, we need to show that the number of bad error patterns is at most

$$q^{-\Omega(\varepsilon n)}|\mathscr{E}_s|.$$

We will prove this by a somewhat careful counting argument.

We begin with a definition.

Definition 7.5.1. Every error pattern $\mathbf{e}$ is associated with a codeword $c(\mathbf{e})$, which is the closest codeword which lies within Hamming distance ρn from it.

For a bad error pattern we insist on having $c(\mathbf{e}) \neq \mathbf{c}$– note that for a bad error pattern such a codeword always exists. Let A be the set of positions where $c(\mathbf{e})$ agrees with $\mathbf{c}+e$.

The rest of the argument will proceed as follows. For each possible A, we count how many bad patterns $\mathbf{e}$ are associated with it (i.e. $\mathbf{c}+\mathbf{e}$ and $c(\mathbf{e})$ agree exactly in the positions in A). To bound this count non-trivially, we will use Lemma 7.5.2.

Define a real number α such that $|A| = \alpha n$. Note that since $c(\mathbf{e})$ and $\mathbf{c}+\mathbf{e}$ agree in at least $1-\rho$ positions,

$$\alpha \geq 1-\rho \geq 1-\delta+\varepsilon. \tag{7.20}$$

For now let us fix A with $|A| = \alpha n$ and to expedite the counting of the number of bad error patterns, let us define two more sets:

$$A_1 = A \cap S,$$

and

$$A_2 = A \setminus A_1.$$

See Figure 7.4 for an illustration of the notation that we have fixed so far.

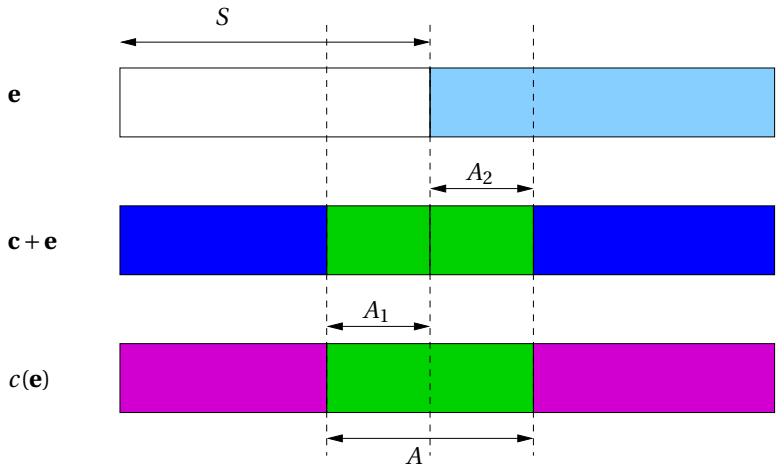

Figure 7.4: Illustration of notation used in the proof of Theorem 7.5.1. Positions in two different vectors that agree have the same color.

.

Define β such that

$$|A_1| = \beta n. \tag{7.21}$$

Note that this implies that

$$|A_2| = (\alpha - \beta) n. \tag{7.22}$$

Further, since $A_1 \subseteq A$, we have

$$\beta \leq \alpha.$$

To recap, we have argued that every bad error pattern $\mathbf{e}$ corresponds to a codeword $c(\mathbf{e}) \neq \mathbf{c}$ and is associated with a pair of subsets (A_1, A_2). So, we fix (A_1, A_2) and then count the number of bad $\mathbf{e}$'s that map to (A_1, A_2). (Later on we will aggregate this count over all possible choices of (A_1, A_2).)

Towards this end, first we overestimate the number of error patterns $\mathbf{e}$ that map to (A_1, A_2) by allowing such $\mathbf{e}$ to have arbitrary values in $[n] \setminus (S \cup A_2)$. Note that all such values have to be non-zero (because of (7.18). This implies that the number of possible distinct $\mathbf{e}_{[n]\setminus(S\cup A_2)}$ is at most

$$(q-1)^{n-|S|-|A_2|} = q^{n-(1-\rho)n-(\alpha-\beta)n}, \tag{7.23}$$

where the equality follows from the given size of S and (7.22). Next fix a non-zero $\mathbf{x}$ and let us only consider error patterns $\mathbf{e}$ such that

$$\mathbf{e}_{[n]\setminus(S\cup A_2)} = \mathbf{x}.$$

Note that at this stage we have an error pattern $\mathbf{e}$ as depicted in Figure 7.5.

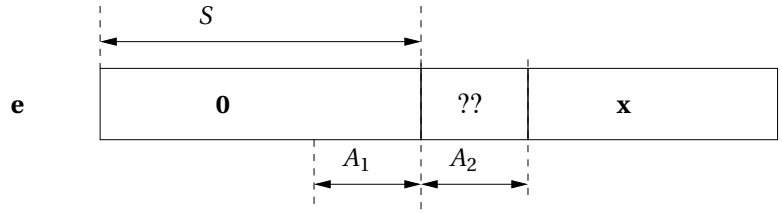

Figure 7.5: Illustration of the kind of error patterns we are trying to count now. The ? denote values that have not been fixed yet.

Now note that if we fix $c(\mathbf{e})_{A_2}$, then we would also fix $\mathbf{e}_{A_2}$ (as $(\mathbf{c}+\mathbf{e})_{A_2} = (c(\mathbf{e}))_{A_2}$). Recall that $\mathbf{c}$ is already fixed and hence, this would fix $\mathbf{e}$ as well. Further, note that

$$c(\mathbf{e})_{A_1} = (\mathbf{c}+\mathbf{e})_{A_1} = \mathbf{c}_{A_1}.$$

This implies that $c(\mathbf{e})_{A_1}$ is already fixed and hence, by Lemma 7.5.2 we would fix $c(\mathbf{e})$ if we fix (say the first) $(1-\delta)n+1-|A_1|$ positions in $c(\mathbf{e})_{A_2}$. Or in other words, by fixing the first $(1-\delta)n+1-|A_1|$ positions in $\mathbf{e}_{A_2}$, $\mathbf{e}$ would be completely determined. Thus, the number of choices for $\mathbf{e}$ that have the pattern in Figure 7.5 is upper bounded by

$$q^{(1-\delta)n+1-|A_1|} = (q-1)^{(1-\delta)n+1-\beta n}, \tag{7.24}$$

where the equality follows from (7.21).

Thus, by (7.23) and (7.24) the number of possible bad error patterns $\mathbf{e}$ that map to (A_1, A_2) is upper bounded by

$$(q-1)^{n-(1-\rho)n-\alpha n+\beta n+(1-\delta)n+1-\beta n} \leq (q-1)^{\rho n-\varepsilon n+1} = (q-1)^{-\varepsilon n+1}|\mathscr{E}_s|,$$

where the inequality follows from (7.20) and the equality follows from (7.19).

Finally, summing up over all choices of $A = (A_1, A_2)$ (of which there are at most 2^n), we get that the total number of bad patterns is upper bounded by

$$2^n \cdot (q-1)^{-\varepsilon n+1} \cdot |\mathscr{E}_S| \leq q^{\frac{n}{\log_2 q}-\frac{\varepsilon n}{2}+\frac{1}{2}} \cdot |\mathscr{E}_A| \leq q^{-\varepsilon n/4} \cdot |\mathscr{E}_S|,$$

where the first inequality follows from $q-1 \geq \sqrt{q}$ (which in turn is true for $q \geq 3$) while the last inequality follows from the fact that for $q \geq \Omega(1/\varepsilon)$ and large enough n, $\frac{n+1/2}{\log_2 q} < \frac{\varepsilon n}{4}$. This completes the proof. □

It can be shown that Theorem 7.5.1 is not true for $q = 2^{o(1/\varepsilon)}$. The proof is left as an exercise.

7.6 Bibliographic Notes

List decoding was defined by Elias [11] and Wozencraft [59].

The result showing that for random error patterns, the list size with high probability is one for the special case of Reed-Solomon codes was shown by McEliece [41]. The result for all codes was proved by Rudra and Uurtamo [47]

In applications of list decoding in complexity theory (see for example [54],[18, Chap. 12]), side information is used crucially to prune the output of a list decoding algorithm to compute a unique answer.

Guruswami [17] showed that the answer to Question 7.3.1 is yes in the sense that there exist linear codes with relative distance δ such that we can find Hamming ball of radius larger than $J_q(\delta)$ with super-polynomially many codewords. This result was proven under a number-theoretic assumption, which was later removed by [27].

(7.17) implies that there exist linear codes with rate $1 - H_q(\rho) - \varepsilon$ that are $\left(\rho, q^{O(1/\varepsilon)}\right)$-list decodable. (This is also true for most linear codes with the appropriate parameters.) However, for a while just for $q = 2$, we knew the *existence* of $\left(\rho, O(1/\varepsilon)\right)$-list decodable codes [21] (though it was not a high probability result). Guruswami, Håstad and Kopparty resolved this "gap" by showing that random linear codes of rate $1 - H_q(\rho) - \varepsilon$ are $(\rho, O(1/\varepsilon))$-list decodable (with high probability) [20].