1 Introduction

We continue the discussion on bounds from the last lecture. To recap, we proved the existence of binary codes with the following asymptotic upper bounds on the rates:

1. The Gilbert-Varshamov bound: This bound shows the existence of binary codes with a bound of $R_{GV} = 1 - h(\delta)$ on the asymptotic rate. Here, h is the binary entropy function, and δ is the relative distance of the code.
2. The Plotkin bound: $R_{Plotkin} = 1 - 2\delta$
3. The Hamming bound: $R_{\text{Hamming}} = 1 - h(\delta/2)$.

Combining all these previous bounds, we see that the upper bound on the rate of a code with relative distance δ satisfies

$$1 - h(\delta) \leq R(\delta) \leq \min(1 - 2\delta, 1 - h(\delta/2))$$

The q-ary versions of these upper bounds are similar: we simply replace $h(\delta)$ with its q-ary version $h_q(\delta)$, and 2δ becomes $\frac{q}{q-1}$ in the Plotkin bound.

$$R_{q\text{-ary Plotkin}} = 1 - \left(\frac{q}{q-1}\right)\delta$$

$$R_{q\text{-ary Hamming}} = 1 - h_q(\delta/2)$$

See Figure 5.1 for a visual comparison of these bounds.

In the figure, codes of rate R and relative distance δ are achievable so long as the point (R, δ) lies below the curve of the GV bound. However, a code with rate R and relative distance δ can only exist if the point (R, δ) is below the Plotkin and the Hamming curves.

Observe that the Hamming code gives better rates than the Plotkin bound for codes with small relative distances. However, the Plotkin bound gives better rates than the Hamming code for codes with larger relative distances. Thus, there is a tradeoff between the δ and R in the Hamming and Plotkin bounds.

In today's lecture, we will introduce some more bounds that improve both Hamming and Plotkin bounds. We will also develop some new classes of error correcting codes.

2 Elias-Bassalygo Bound

2.1 Statement and Motivation

In this section, we will present the EB (Elias-Bassalygo) bound, which outperforms both the Plotkin bound and the Hamming bound.

Theorem 5.1 (Elias-Bassalygo Bound)**.** *For sufficiently large n, a binary code C of relative distance δ and rate R satisfies the bound*

$$R \leq R_{EB}(\delta) \coloneqq 1 - h(J(\delta)),$$

where $J(\delta) = \frac{1-\sqrt{1-2\delta}}{2}$, and is called the Johnson radius. For the q-ary version of the bound,

$$R(\delta) \leq R_{EB}(\delta) = 1 - h_q(J_q(\delta)),$$

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

Before proving the theorem, we state the following useful facts:

Fact 1. *For any $\delta \in (0, 1/2]$, we have that*

$$\frac{1-\sqrt{1-2\delta}}{2} > \delta/2,$$

Fact 2. *for any $x \in (0, 1/2]$, we have that $h(x) \geq 4x(1-x)$.*

Bringing these together, we can show that R_{EB} is a better (tighter) upper bound than $R_{\text{Plotkin}}(\delta) = 1 - 2\delta$ and $R_{\text{Hamming}}(\delta) = 1 - h(\delta/2)$.

2.2 Proof of the Elias-Bassalygo Bound

In this section, we will prove the EB bound. The key idea of this proof is to extend the Hamming ball argument from previous proofs, but to extend it via the idea of list-decoding.

Recall the argument we used to prove the Hamming bound (binary case)– we deduced that,

$$|C| \cdot |B_n(0, (\delta/2) \cdot n)| \leq 2^n.$$

Recall that we used the growth rate of the entropy function to bound the volume of a Hamming ball of radius $\frac{\delta \cdot n}{2}$. This gave us the following bound:

$$|C| \cdot 2^{h(\delta/2)n} \leq 2^n$$

If we increase the volume of the balls, beating the Hamming and Plotkin bounds may be achievable. However, increasing the volume may sabotage the unique error-correction property. That is, if a message is encoded and passed through a channel where the codeword is corrupted, then the correct codeword cannot be uniquely determined. Instead, we get a list L of possible corrected codewords.

More formally, we will increase the radius of the balls, such that they include each codeword, say at most L times. We can then introduce a factor of L on one side of the equation to account for this. We will show that when we set the radius of the balls to be ρn, each point will only be counted $L = O(n)$ times. This will yield us the bound

$$|C| \cdot 2^{h(\rho)n} \leq L \cdot 2^n$$

Thus our goal is to show that for any code $C \subseteq \{0,1\}^n$ and $y \in \{0,1\}^n$ such that $\text{dist}(C) \geq \delta$, we have that

$$|\text{Ball}(y, \rho n) \cap C| \leq O(n).$$

Equivalently stated, we can list decode $\rho(n)$ errors with list size $O(n)$. Setting $\rho = J(\delta)$, our claim is that

$$|\text{Ball}(y, J(\delta)n) \cap C| \leq L \tag{5.1}$$

With this intuition, we will now provide a more complete proof sketch.

Proof. Without loss of generality, by translating the code, we can consider the center of the ball to be $y = 0^n$

Now, consider the map $V : \{0,1\}^n \to \mathbb{R}^n$ defined as

$$V : c \mapsto v_c \coloneqq \frac{1}{\sqrt{n}}((-1)^{c_1}, (-1)^{c_2}, \dots (-1)^{c_n}) \ .$$

In order to prove our bound, we will first need the following geometric lemma.

Lemma 1. *Let $u_1, \dots u_m$ be nonzero vectors in $\mathbb{R}^n$. If $\langle u_i, u_j \rangle \leq 0$ for all $1 \leq i < j \leq m$, then $m \leq 2n$.*

Using this lemma, we will show that the size of the list L is $\mathcal{O}(n)$.

Note that $V(y) = [1, 1, \dots, 1]$ as $y = 0^n$. Now, let $c_1, \dots c_m$ be codewords that satisfy the following conditions:

1. $\Delta(c_i, c_j) \geq \delta n$, which follows by the definition of the minimum distance of the code.
2. $\Delta(y, c_i) \leq \rho \cdot n$, i.e. these codewords are within the ball of radius ρn around y.

Next, we pick a parameter α (that we will optimize) to ensure that

$$\langle v_i - \alpha v_y, v_j - \alpha v_y \rangle \leq 0,$$

which implies that $m \leq 2n = \mathcal{O}(n)$.

Simplifying the left side,

$$\begin{aligned}
\langle v_i - \alpha v_y, v_j - \alpha v_y \rangle &= \langle v_i, v_j \rangle - \alpha \left(\langle v_y, v_i \rangle + \langle v_y, v_j \rangle \right) + \alpha^2 n \\
&\leq n - 2\delta n + \alpha^2 n - 2\alpha(1 - 2\rho)n \\
&= n(1 - 2\delta + \alpha^2 - 2\alpha + 4\alpha\rho) \\
&\leq 0
\end{aligned}$$

This means

$$4\rho \leq 2 - \left(\frac{1 - 2\alpha}{\alpha} + \alpha \right).$$

Taking $\alpha = \sqrt{1 - 2\delta}$, we then get that

$$4\rho \leq 2 - 2\sqrt{1 - 2\delta} \iff \rho \leq 1 - \frac{\sqrt{1 - 2\delta}}{2} \ .$$

Thus $\rho = J(\delta)$. By Plugging in $\rho = J(\delta)$ and $\alpha = \sqrt{1 - 2\delta}$ and applying Lemma 1 on the vectors $v_{c_i} - \alpha v_y]$ proves Inequality (5.1). By the discussion preceding Inequality 5.1, this therefore finishes the proof of Theorem 5.1. □

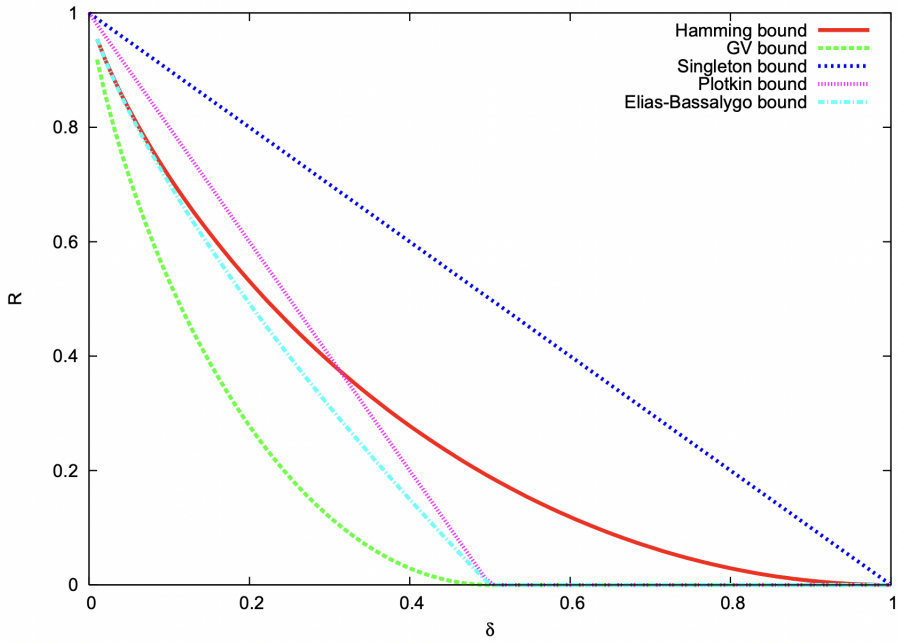

Figure 5.1: Above we can see the comparison of the bounds that we have developed for rates versus relative distances. This plot is from https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/web-coding-book.pdf

3 Best Bounds

The best bounds we can achieve for the rate are known as the MRRW bounds. These bounds are based on a linear programming approach, but are still not as tight as the GV bound. In their first result, the MRRW bound was

$$R_{\mathrm{MRRW1}}(\delta) = h(1/2 - \sqrt{\delta(1-\delta)}).$$

Note that this bound is in fact weaker than R_{EB} in the regime when $\delta \leq 0.15$. This bound was then refined in subsequent work to yield

$$R_{\mathrm{MRRW2}}(\delta) = \min_{\delta/2 \leq \rho \leq 1/2} \{1 - h(\rho) + R(\rho\delta)\} \ .$$

This bound is better for every $\delta \in [0, 1/2]$.

These methods are derived by using a linear programming bound. We consider a graph $G = (V, E)$, where $V = \{0,1\}^n$. We then draw an edge $(u, v) \in E$ if and only if $\Delta(u, v) < d$. For any code $C \subseteq \{0,1\}^n$, it then follows that $\mathrm{dist}(C) \geq d$ if and only if C is an independent set in G.

4 BCH Codes

In this section, we will introduce the study of (binary) BCH (Bose-Chaudhuri-Hocquenghem) codes. For very small δ ($\delta < 1/\log n$), this family of codes approaches the Hamming bound. The distance of the BCH code is $2t+1$ and has dimension $n - t\lceil \log n \rceil - O(1)$. Formally, we adopt the following definition.

Definition 1 (BCH Codes)**.** For a given length $n = 2^m - 1$, a desired distance D, and primitive element $\alpha \in \mathbb{F}_{2^m}^*$. We define the binary BCH code as

$$\{(c_1, c_2, \ldots, c_n) \in \mathbb{F}_{2^n} | c(X) = c_0 + c_1 X \cdots + c_{n-1} X^{n-1} \text{ satisfies } c(\alpha) = c(\alpha^2) = \ldots c(\alpha^{D-1}) = 0\}.$$

From the definition of BCH codes, we immediately see that the parity-check matrix of the BCH code is

$$\begin{bmatrix} 1 & 1 & \ldots & 1 \\ \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \alpha_1^3 & \alpha_2^3 & \ldots & \alpha_n^3 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{2t-1} & \alpha_2^{2t-1} & \ldots & \alpha_n^{2t-1} \end{bmatrix}.$$

As we have seen in a previous lecture, the even-power parity-checks follow from the odd-power parity-checks. Note that when $t = 1$, the BCH code is exactly the Hamming code (whose distance is 3).

5 Reed-Solomon Codes

Here, we introduce the notion of Reed-Solomon codes.

Definition 2 (Reed-Solomon Codes)**.** For $k \leq n \leq q$, the $[n, k]_q$ Reed-Solomon code is defined by a set of distinct evaluation points $\alpha_1, \ldots \alpha_n$ over the finite field $\mathbf{F}_q$.The encoding function for the corresponding Reed-Solomon code is $RS : \mathbf{F}_q^k \to \mathbf{F}_q^n$. which will be defined as follows: given a message $m \in \mathbf{F}_q^k$, we define its corresponding degree $k-1$ polynomial by assigning the i-th component of m to the coefficient of x^i. Explicitly, the polynomial $f_m(X)$ of degree at most $k-1$ associated to m is

$$f_m(X) = \sum_{i=0}^{k-1} m_i X^i \,.$$

The function RS is then defined to be the evaluations of the polynomial $f_m(X)$ over the evaluation points $\alpha_1, \ldots, \alpha_n$. That is,

$$RS(m) = (f_m(\alpha_1), f_m(\alpha_2), \ldots, f_m(\alpha_n)) \,.$$

The most frequent setting of parameters is when $n = q - 1$ and the set of evaluation points being all but the 0 element.

To understand why Reed-Solomon codes are powerful, we will introduce a lemma regarding these polynomials.

Lemma 2. *For any distinct $a_1, \ldots, a_k \in \mathbf{F}_q$, and corresponding $b_1, \ldots, b_k \in \mathbf{F}_q$, there exists a unique polynomial of degree $\leq k-1$ such that $f(a_i) = b_i$ for all $i = 1, \ldots, k$.*

Proof. First, we prove the existence of such a polynomial. We can construct a polynomial satisfying these conditions using Lagrangian interpolation. We note that the functions $I_j(X) = \prod_{i \neq j}(X - a_i)$ will return polynomials that are zero everywhere except at the point a_j. Furthermore, they are of degree $k-1$. After subsequent scaling by some value (say α_j), we can enforce that each polynomial $\alpha_j I_j(X)$ is zero everywhere except a_i, where it will take value b_i. Then, by adding together such polynomials for each a_j, we will receive a polynomial of degree $\leq k-1$, such that for each a_i, it evaluates to b_i.

Next, we prove uniqueness. We suppose for the sake of contradiction that there exist two polynomials f, g such that $\forall i, f(a_i) = b(i), g(a_i) = b_i$. But then the difference of the polynomials $f - g$ has k zeros, despite being degree less than k. Since a nonzero polynomial cannot have more roots than its degree, this means it must be the zero polynomial, so f and g are indeed identical. □

Now, we will use this lemma to prove a bound on the distance of our Reed-Solomon codewords.

Claim 1. *The $[n, k]_q$ Reed-Solomon code has distance at least $n - k + 1$.*

Proof. Suppose we are given two distinct messages $m_1, m_2 \in \mathbf{F}_q^k$. The corresponding polynomials are then $f_{m_1}(x)$ and $f_{m_2}(x)$. Because $m_1 \neq m_2$, we therefore have $f_{m_1}(x) \neq f_{m_2}(x)$. By our previous lemma, this means that the polynomials can only agree on at most $k - 1$ points, as otherwise they are defining the same unique polynomial. Because the evaluation set is of size n, this means that there are $\geq n - k + 1$ evaluation points where the polynomials evaluate to different values. Hence, the minimum distance is at least $n - k + 1$. □

As an aside, we can see that the above distance matches the Singleton bound. This therefore means that Reed-Solomon codes attain the best rate/distance tradeoff over any alphabet size. Note that $q \geq n$, which means that while Reed-Solomon codes have an optimal rate/distance tradeoff, they suffer from a relatively large alphabet size.

Finally, we will try to make a generator matrix for this code. The simplest natural basis with which we can generate polynomials of degree up to $k - 1$ is the set of monomials $1, X, \ldots, X^{k-1}$. Evaluating these points at $\alpha_1, \ldots, \alpha_n$, thus yields the matrix:

$$\begin{bmatrix} 1 & \alpha_1 & \ldots & \alpha_1^{k-1} \\ 1 & \alpha_2 & \ldots & \alpha_2^{k-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha_n & \ldots & \alpha_n^{k-1} \end{bmatrix}.$$

Upon right multiplication by a vector $m = [m_0, \ldots, m_{k-1}]$, this will yield exactly the codeword $[f_m(\alpha_1), \ldots f_m(\alpha_n)]$.