Lecture 9

Lecturer: Madhu Sudan *Scribe: Nitin Thaper*

Today we will talk about:

- MacWilliams identities
- LP bound
- Some perspectives on asymptotics

1 MacWilliams Identities

Recall that our theme in the last lecture was to upper bounds on the rate of *any* family of codes given its relative distance. Bounds of this form, that hold for *every* code or family of codes, are termed universal bounds. Our first topic today are bounds of this form for all *linear* codes. While these are not directly formulated as upper bounds on the rate of codes, given their relative distance, these bounds do play an essential role in some of the strongest known bounds on rate. We will get to that part later, but first we introduce and prove the MacWilliams Identities.

1.1 Linear codes and their duals

Recall that a linear code C is specified by a $k \times n$ generator matrix $\mathbf{G}$. Alternatively, the code can be specified by a $n \times (n-k)$ parity check matrix $\mathbf{H}$. The matrix $\mathbf{G}$ consists of k linearly independent rows while $\mathbf{H}$ consists of $n-k$ linearly independent columns.

The dual of the linear code C, denoted $C^{\perp}$, is the the code *generated* by $\mathbf{H}^T$, the transpose of the parity check matrix of C. It is obvious that $C^{\perp}$ is also a linear code, with block length n and message length $n-k$. Furthermore every pair of codewords $\mathbf{b} \in C$ and $\mathbf{c} \in C^{\perp}$ satisfy $\langle \mathbf{b}, \mathbf{c} \rangle = 0$. A slightly stronger fact is proven in the proposition below.

Proposition 1 *For linear code $C \subseteq \mathbb{F}_q^n$ and vector $\mathbf{x} \in \mathbb{F}_q^n$, the following hold:*

- *If $\mathbf{x} \in C^{\perp}$, then for every $\mathbf{c} \in C$, $\langle \mathbf{c}, \mathbf{x} \rangle = 0$.*
- *If $\mathbf{x} \notin C^{\perp}$, then for every $\alpha, \beta \in \mathbb{F}_q$, the sets $\{\mathbf{c} \in C \mid \langle \mathbf{c}, \mathbf{x} \rangle = \alpha\}$ and $\{\mathbf{c} \in C \mid \langle \mathbf{c}, \mathbf{x} \rangle = \beta\}$ have the same cardinality.*

Proof The first part of the proposition follows from the definition of the dual of a code. For the second part note first that without loss of generality, we may assume $\beta = 0$ and $\alpha \neq 0$. Let $S_\alpha = \{\mathbf{c} \in C \mid \langle \mathbf{c}, \mathbf{x} \rangle = \alpha\}$, and $S_0 = \{\mathbf{c} \in C \mid \langle \mathbf{c}, \mathbf{x} \rangle = 0\}$. Note that $\mathbf{0} \in S_0$ and hence the latter is non-empty. We note next that the former is also non-empty. Since $\mathbf{x} \notin C^{\perp}$ we have that there exists $\mathbf{c} \in C$ such that $\langle \mathbf{c}, \mathbf{x} \rangle = \alpha' \neq 0$. Then we have that $\mathbf{b} = (\alpha')^{-1}\alpha\mathbf{c}$ is an element of C with $\langle \mathbf{b}, \mathbf{x} \rangle = \alpha$, and thus $\mathbf{b} \in S_\alpha$.

For a set $A \subseteq \mathbb{F}_q^n$ and vector $\mathbf{y} \in \mathbb{F}_q^n$, let $\mathbf{y} + A$ denote the set $\{\mathbf{y} + \mathbf{x} \mid \mathbf{x} \in A\}$. Fix $\mathbf{b} \in S_\alpha$. We get $|S_\alpha| = |S_0|$ from the following series of facts:

1. $\mathbf{b} + S_0 \subseteq S_\alpha$: If $\mathbf{a} \in S_0$ then $\langle \mathbf{b} + \mathbf{a}, \mathbf{x} \rangle = \langle \mathbf{b}, \mathbf{x} \rangle + \langle \mathbf{a}, \mathbf{x} \rangle = \alpha + 0 = \alpha$.
2. $|\mathbf{b} + S_0| = |S_0|$: Follows since $\mathbf{b} + \mathbf{a} = \mathbf{b} + \mathbf{a}'$ iff $\mathbf{a} = \mathbf{a}'$.

3. $|S_0| \leq |S_\alpha|$: Follows from Parts (1) and (2) above.
4. $|S_\alpha| \leq |S_0|$: Similar to Parts (1)-(3) above, except that here we work with the set $(-\mathbf{b}) + S_\alpha$.

■

1.2 Weight distributions and the weight generating function

Definition 2 (Weight distribution & generating function) *Given an $[n,k,d]_q$ code C, and index $i \in \{0,\ldots,n\}$, the ith weight enumerator of C is the number of codewords in C of Hamming weight exactly i. The weight distribution of C is the sequence $\langle A_0,\ldots,A_n\rangle$, where A_i is the ith weight enumerator of C. The weight generating function of C is the formal polynomial $A_C(x) = \sum_{i=0}^n A_i x^i$, where $\langle A_0,\ldots,A_n\rangle$ is the weight distribution of C.*

The MacWilliams Identities relate the weight distribution of a code C with that of the code $C^\perp$. We will do so by studying the *weight generating function*, $A_C(y) = \sum_{i=0} A_i y^i$, of the code C, and relating A_C to $A_{C^\perp}$. As is by now usual, we will prove the identity first for the binary case, and then move to the q-ary case later.

1.3 The extended generating function

We will start by defining an elaborate generating function of a code C, called its *extended generating function*. It will be quite evident that this generating function preserves all information about the code C (unlike the weight generating function which does not tell us exactly what the code is). We will then relate this elaborate generating function of the code to that of its dual.

We will introduce the extended generating function gently. We will start by defining the extended generating function of a single bit(!), and then define it for a word in $\{0,1\}^n$. and then define the extended generating function of a code.

Definition 3 (Extended generating function of a bit) *The weight generating function of C is the formal polynomial $A(x) = \sum_{i=0}^n A_i x^i$. For a bit $b \in \{0,1\}$, the extended generating function $W_b(x,y)$, is a polynomial in two variables x and y defined as:*

$$\begin{aligned} W_b(x,y) &= x \qquad \textit{if } b = 0 \\ &= y \qquad \textit{if } b = 1 \end{aligned}$$

Definition 4 (Extended generating function of a word) *For a vector $\mathbf{b} = \langle b_1,\ldots,b_n\rangle \in \{0,1\}^n$, the extended generating function $W_{\mathbf{b}}(\mathbf{x},\mathbf{y})$, is a polynomial in $2n$ variables, $\mathbf{x} = \langle x_1,\ldots,x_n\rangle$ and $\mathbf{y} = \langle y_1,\ldots,y_n\rangle$, defined as: $W_{\mathbf{b}}(\mathbf{x},\mathbf{x}) = \prod_{i=1}^n W_{b_i}(x_i,y_i)$.*

Finally we define the extended generating function of a code C.

Definition 5 (Extended generating function of a code) *For a code $C \subseteq \{0,1\}^n$, the extended generating function $W_C(\mathbf{x},\mathbf{y})$, is a polynomial in $2n$ variables, $\mathbf{x} = \langle x_1,\ldots,x_n\rangle$ and $\mathbf{y} = \langle y_1,\ldots,y_n\rangle$, defined as: $W_C(\mathbf{x},\mathbf{x}) = \sum_{\mathbf{b}\in C} W_{\mathbf{b}}(\mathbf{x},\mathbf{y})$.*

To make sense of the definitions above, it would help to see an example. Consider the code $C = \{000, 101, 110, 011\}$. The extended generating function for this code is:

$$W_C(\mathbf{x}, \mathbf{y}) = x_1x_2x_3 + y_1x_2y_3 + y_1y_2x_3 + y_1y_2x_3 + x_1y_2y_3.$$

It should be immediately clear that the extended generating function carries all the information about a code and does not do so in any especially clever way. The extended generating function is not intended to be a way of compressing information about the code, but rather an elaborate way of representing the code, which will come in useful in studying the combinatorics of codes. At the outset we are not hoping to use them to glean any information about codes in a computationally efficient manner.

Given this elaborate representation of a code, it should not come as a surprise that given the extended generating function of a code C we can derive the extended generating function of its dual. After all. all the information of C is embedded into $W_C(\mathbf{x}, \mathbf{y})$, so C can be derived from W_C, $C^\perp$ can be derived from C, and finally $W_{C^\perp}$ can be derived from $C^\perp$. The generalized MacWilliams Identity just gives an explicit form of this rather obvious statement. The strength of the result lies in the fact that the relationship takes an especially simple closed form - one that will turn out to be especially amenable to manipulations later on.

The crux of the identity is some sort of a "Discrete Fourier Transform" (or Hadamard transform or Walsh transform, depending on your loyalties). Instead of focusing on the function $W_C(\mathbf{x}, \mathbf{y})$ we will focus on the function $W_C(\mathbf{x}+\mathbf{y}, \mathbf{x}-\mathbf{y})$. Before saying what this transform does to the generating function of a code, we will describe what the transform does to the generating function of a single vector $\mathbf{b}$.

Lemma 6

$$W_{\mathbf{b}}(\mathbf{x}+\mathbf{y}, \mathbf{x}-\mathbf{y}) = \sum_{\mathbf{c}\in\{0,1\}^n} (-1)^{\langle \mathbf{b},\mathbf{c}\rangle} W_{\mathbf{c}}(\mathbf{x}, \mathbf{y}).$$

Proof Note by definition of $W_{\mathbf{b}}$ that

$$W_{\mathbf{b}}(\mathbf{x}+\mathbf{y}, \mathbf{x}-\mathbf{y}) = \prod_{i=1}^{n} W_{b_i}(x_i+y_i, x_i-y_i) = \prod_{i=1}^{n} \left(x_i + (-1)^{b_i} y_i\right).$$

Expanding the right hand side above we get a sum of 2^n terms of the form $\pm z_1 \cdots z_n$, where $z_i \in \{x_i, y_i\}$ and the sign is odd iff there is an odd number of indices i such that where $z_i = y_i$ and $b_i = 1$. Letting $\mathbf{c} \in \{0,1\}^n$ denote the index of this sum of 2^n terms, and setting $z_i = y_i$ if $c_i = 1$, we see that $z_i = W_{c_i}(x_i, y_i)$ and the sign in front is odd iff $\langle \mathbf{b}, \mathbf{c}\rangle = 1$. Thus we have

$$\prod_{i=1}^{n} \left(x_i + (-1)^{b_i} y_i\right) = \sum_{\mathbf{c}\in\{0,1\}^n} (-1)^{\langle \mathbf{b},\mathbf{c}\rangle} \prod_{i=1}^{n} W_{c_i}(x_i, y_i) = \sum_{\mathbf{c}\in\{0,1\}^n} (-1)^{\langle \mathbf{b},\mathbf{c}\rangle} W_{\mathbf{c}}(\mathbf{x}, \mathbf{y}).$$

This yields the lemma. ■

Theorem 7 (Generalized MacWilliams Identity [5]) *The extended generating function of a code C and its dual $C^\perp$ satify the identity:*

$$W_{C^\perp}(\mathbf{x}, \mathbf{y}) = \frac{1}{|C|} W_C(\mathbf{x}+\mathbf{y}, \mathbf{x}-\mathbf{y}).$$

Proof The proof follows easily from Proposition 1 and Lemma 6. Note that

$$\begin{aligned} W_C(\mathbf{x}+\mathbf{y}, \mathbf{x}-\mathbf{y}) &= \sum_{\mathbf{b}\in C} W_{\mathbf{b}}(\mathbf{x}+\mathbf{y}, \mathbf{x}-\mathbf{y}) \\ &= \sum_{\mathbf{b}\in C} \sum_{\mathbf{c}\in\{0,1\}^n} (-1)^{\langle \mathbf{b},\mathbf{c}\rangle} W_{\mathbf{c}}(\mathbf{x},\mathbf{y}) \quad \text{(By Lemma 6)} \\ &= \sum_{\mathbf{b}\in C} \sum_{\mathbf{c}\in C^\perp} W_{\mathbf{c}}(\mathbf{x},\mathbf{y}) + \sum_{\mathbf{b}\in C} \sum_{\mathbf{c}\notin C^\perp} (-1)^{\langle \mathbf{b},\mathbf{c}\rangle} W_{\mathbf{c}}(\mathbf{x},\mathbf{y}) \\ &= |C| W_C(\mathbf{x},\mathbf{y}) + 0, \end{aligned}$$

where the first part of the last equation is by definition of the extended generating function and the second part applies Proposition 1 to every $\mathbf{c} \notin C^\perp$. The theorem follows. ■

1.4 The MacWilliams Identities

We now return to the goal of studying the weight distribution of a code C. The following, simple proposition shows that we have made some progress already!

Proposition 8 *For every linear code C, we have $x^n A_C(y/x) = W_C(x, \ldots, x, y, \ldots, y)$.*

Proof The proposal follows easily by inspecting the right hand side. Under the substitution $x_i = x$ and $y_i = y$, we have

$$\begin{aligned} W_C(x, \ldots, x, y, \ldots, y) &= \sum_{\mathbf{b}\in C} x^{n-\mathrm{wt}(\mathbf{b})} y^{\mathrm{wt}(\mathbf{b})} \\ &= x^n \sum_{\mathbf{b}\in C} (y/x)^{\mathrm{wt}(\mathbf{b})} \\ &= x^n \sum_{i=0}^{n} \sum_{\mathbf{b}\in C \ \mid \ \mathrm{wt}(\mathbf{b})=i} (y/x)^i \\ &= x^n \sum_{i=0}^{n} A_i (y/x)^i \\ &= x^n A_C(y/x). \end{aligned}$$

■

The MacWilliams Identity now follows easily:

Theorem 9 (MacWilliams Identity [5])

$$A_{C^\perp}(y) = \frac{1}{|C|}(1+y)^n A_C\left(\frac{1-y}{1+y}\right).$$

Proof The proposition we just proved, Proposition 8, tells us that $A_{C^\perp}(y) = 1^n W_{C^\perp}(1, \ldots, 1, y, \ldots, y)$. Now, applying the generalized MacWilliams identity (Theorem 7), we get

$$W_{C^\perp}(1, \ldots, 1, y, \ldots, y) = \frac{1}{|C|} W_C(1+y, \ldots, 1+y, 1-y, \ldots, 1-y).$$

Finally applying Proposition 8 again, we have

$$W_C)(1+y,\ldots,1+y,1-y,\ldots,1-y) = (1+y)^n A_C((1-y)/(1+y)).$$

Putting the above together, we get the theorem. ■

In the appendix to this lecture, we extend the bound above to the q-ary case. We state the q-ary version so that we can discuss the more general result and its implications.

Theorem 10 (q-ary MacWilliams Identity) *For a q-ary linear code C of block length n, we have*

$$A_{C^\perp}(y) = \frac{(1+(q-1)y)^n}{|C|} A_C\left(\frac{1-y}{1+(q-1)y}\right).$$

The MacWilliams identity shows that the weight distribution of a linear code can be computed from the weight distribution of its dual! Note, this is totally non-trivial - we can see no obvious reason why specifying the weight distribution of C, should fix the weight distribution of $C^\perp$. In a sense the identity suggests that any two linear codes of a given weight distribution are essentially the same, motivating the following question (the answer to which is not obvious to Madhu).

Question: **True or False:** For any pair of q-ary codes C_1 and C_2 of block length n that have the same weight distribution, there exists a permutation $\pi : [n] \to [n]$ such that $\mathbf{c} = \langle c_1, \ldots, c_n \rangle \in C_1$ iff such that $\mathbf{c}' = \langle c_{\pi(i)}, \ldots, c_{\pi(n)} \rangle \in C_2$.

We now examine the exact form of the relationship obtained between the weight distribution of a code and its dual. As in the case of the relationship between the extended (elaborate?) generating functions of a code and its dual, the exact form of the relationship turns out to be simple and quite useful. In particular, the weight distribution of the dual code is just a linear function of the weight distribution of the primal code, as we note below.

Corollary 11 *For every q and n there exists an $(n+1) \times (n+1)$ rational matrix $\mathbf{M}$ such that the following holds: Let $\mathbf{a} = \langle A_0, \ldots, A_n \rangle$ be the weight distribution of a q-ary linear code C of block length n, and let $\mathbf{b} = \langle B_0, \ldots, B_n \rangle$ be the weight distribution of $C^\perp$. Then*

$$\mathbf{b} = \frac{1}{A_0 + \cdots + A_n} \mathbf{a}\mathbf{M}.$$

Furthermore, the matrix $\mathbf{M} = \{m_{ij}\}$ is given by $m_{ij} = \sum_{\ell=0}^{i} \binom{n-j}{i-\ell}\binom{j}{\ell}(-1)^\ell (q-1)^{i-\ell}$.

Proof Since B_i is the coefficient of y^i in $A_{C^\perp}(y)$, we need to examine the coefficient of y^i in $\frac{(1+(q-1)y)^n}{|C|} A_C\left(\frac{1-y}{1+(q-1)y}\right)$. Write this quantity as $\frac{1}{|C|}\sum j = 0^n A_j (1+(q-1)y)^{n-j}(1-y)^j$. The coefficient of y^i in this sum is obtained as the coefficient of y^ℓ in the expansion of $(1-y)^j$ times the coefficient of $y^{i-\ell}$ in the expansion of $(1+(q-1)y)^{n-j}$, summed up over $\ell \in [i]$. We thus get:

$$\begin{aligned} B_i &= \frac{1}{|C|}\sum_{j=0}^{n} A_j \sum_{\ell=0}^{i} \binom{n-j}{i-\ell}\binom{j}{\ell}(q-1)^{i-\ell}(-1)^\ell \\ &= \frac{1}{|C|}\sum_{j=0}^{n} A_j m_{ij} \end{aligned}$$

The corollary follows from the fact that $|C| = A_0 + \cdots + A_n$. ■

In the rest of this lecture we will describe (without proof) two consequences of this explicit form of the MacWilliams Identity.

2 Weight distribution of MDS codes

Recall the notion of an MDS (Maximum Distance Separable) code. These code are codes that meet the Singleton bound, i.e., have a minimum distance equal to $n-k+1$, where n is the block length and k is the message length of the code. In this section, we will describe the weight distribution function of linear MDS codes exactly, for every n and k! This ought to be surprising, since this implies that all MDS codes have the same weight distribution and we have no prior reason to believe so! The first step in our proof is a simple but surprising fact.

Proposition 12 *If C is a linear MDS code, then so is its dual $C^{\perp}$.*

Proof The proof is a slight variation of the proof of the Singleton bound. Let C be an $[n,k,n-k+1]$-code. We know that $C^{\perp}$ is an $[n,n-k,d]$-code for some $d \leq k+1$ (using the Singleton bound), and we need to show $d \geq k+1$. Assume otherwise, and suppose there exists a vector $\mathbf{c} \in C^{\perp}$ with $\mathrm{wt}(\mathbf{c}) \leq k$. Without loss of generality assume that the first $i \leq k$ coordinates of $\mathbf{c}$ are non-zero and the remaining coordinates are zero. Then for every codeword $\mathbf{b} \in C$, we have $\sum_{j=1}^{k} b_j c_j = 0$. Project the codewords of C to their first k coordinates. We claim that the projection of C is a subspace of F_q^k of dimension $k-1$ or less (since $\sum_{j=1}^{k} b_j c_j = 0$ for all vectors in the projection). In such a case there exist two vectors $\mathbf{c}_1$ and $\mathbf{c}_2$ in C, whose projection on to the first k coordinates is the same. But then $\Delta(\mathbf{c}_1,\mathbf{c}_2) \leq n-k$ contradicting the hypothesis that C has minimum distance $n-k+1$. ■

Combining the proposition above with MacWilliams Identities, one gets the following surprising result — explicit expressions for the weight distribution for all MDS codes.

Theorem 13 *If C is an $[n,k,n-k+1]_q$-MDS code, then the weight distribution $\langle A_0,\ldots,A_n\rangle$ of C is given by $A_i = \binom{n}{i}\sum_{j=0}^{i+k-n}(-1)^j\binom{i}{j}((q^{i+k-n-j}-1)$.*

We won't prove the theorem here — but lets see why it should not be too surprising if we put together what we know. Consider the $2n+2$ variables corresponding to the weight distribution of C and $C^{\perp}$. The MacWilliams Identities gives us $n+1$ linear conditions relating these variables once the dimension of the primal code is fixed. Using these conditions in the obvious way, one can see that one can compute the dual weights given the primal distribution. By duality, one can also compute the primal weights given the dual weight distribution. Presumably there are other subsets of $(n+1)$ of the weights that would lead to the rest being specified as well! In the case of MDS codes we know $A_0 = B_0 = 1$ and $A_1 = \cdots = A_{n-k} = B_1 = \cdots = B_k = 0$ - which turns to be $n+2$ variables, that fortunately are of full rank. So the linear system can be solved and yields the theorem above!

Exercise: Determine which subsets of the $2n+2$ variables in the MacWilliams Identities are linearly independent. (Answer not known to Madhu.)

Notice a simple corollary to the above theorem: There are exactly A_i polynomials in $\mathbb{F}_q[x]$ of degree $< k$ that have exactly $n-i$ zeroes in any given subset $S \subseteq \mathbb{F}_q$ of cardinality n.

On the one hand the theorem above is quite impressive in that it gives exact numbers. However since the final expressions are not closed forms it is hard to get a sense of the growth of the expressions. A coarse approximation that works out reasonably is $A_i = \Theta(\binom{n}{i}q^{k+i-n})$ for MDS codes.

3 The Linear Programming bound

We now describe the most powerful application of the MacWilliams Identities, namely the Linear Programming (LP) bound on the rate of a code.

Let C be a $[n, ?, d]_q$ code with weight distribution $A_0, A_1, \ldots, A_n$. The number of codewords for the code is $\sum_i A_i$ and hence the rate $R = \log_q(\sum A_i)/n$. As mentioned earlier our goal is to get an asymptotic upper bound on the rate R. This reduces to deriving an upper bound on $\sum A_i$ subject to the restrictions that

$$A_0 = 1, A_1 = 0, A_2 = 0, \ldots, A_{d-1} = 0$$
$$B_0 = 1, B_1, B_2, \ldots, B_n \geq 0$$

As noted earlier, B_i's are essentially linear functions of A_i's. Precisely, $B_i \cdot \left(\sum_{j=0}^{n} A_j\right)$ is given by a linear function of the A_i's.

This motivates the following linear program:

$$\begin{array}{ll} \text{Maximize} & K_n = \sum_{i=0}^{n} A_i \\ \text{subject to} & \\ & \mathbf{aM} \geq \mathbf{0} \\ & A_0 = 1 \\ & A_1 = \cdots = A_{d-1} = 0 \\ & A_d, \ldots, A_n \geq 0 \end{array}$$

where $\mathbf{a} = \langle A_0, \ldots, A_n \rangle$, $\mathbf{M}$ is the matrix described in Corollary 11, and the notation $\mathbf{x} \geq \mathbf{0}$, implies every coordinate of $\mathbf{x}$ is non-negative.

The linear program above bounds the number of codewords that any linear code of distance d can have over a q-ary alphabet, and one can hope that linear programming and duality could help anaylze the quantity K_n above. The quantity $R_{\text{LP}} \stackrel{\text{def}}{=} \log_q(K_n)/n$ is called the LP (upper) bound on the rate of a linear code. Somewhat surprisingly, even though the linear program was motivated only for linear codes, the LP bound holds also for non-linear codes, as implied by the following theorem.

Theorem 14 *If C is an $(n, k, d)_q$ code with $A_i \stackrel{\text{def}}{=} \frac{1}{|C|} \sum_{\mathbf{c} \in C} |\{\mathbf{b} \in C \mid \Delta(\mathbf{b}, \mathbf{c}) = i\}|$. Let $\mathbf{a} = \langle A_0, \ldots, A_n \rangle$ and let $\mathbf{M}$ be as in Corollary 11. Then $\mathbf{aM} \geq \mathbf{0}$.*

Thus the quantity R_{LP} is the LP bound on the rate of *all* codes. The LP upper bound was discovered by Delsarte [2] who showed how to derive several known upper bounds using this framework. Performing a tight asymptotic analysis of the LP bound is non-trivial and took a while after the work of Delsarte. The best asymptotic analysis of the LP bound is due to McEliece, Rodemich, Rumsey and Welch [6] who give two upper bounds on the LP bound. We state their bounds for the binary case:

Theorem 15 (MRRW bound [6]) *If $\mathcal{C}$ is a family of binary codes (not necessarily linear) of rate R and relative distance δ then*

$$R \leq H\left(\frac{1}{2} - \sqrt{\delta(1-\delta)}\right),$$

$$\textit{and} \quad R \leq \min_{u \in [0, 1-2\delta]} \{1 + g(u^2) - g(u^2 + 2\delta u + 2\delta)\} \quad \textit{where} \quad g(x) = H_2\left(\frac{1 - \sqrt{1-x}}{2}\right).$$

Proving these bounds is way out of scope of this lecture(r)! The interested reader is pointed to thesis of Samorodnitsky [7], or the article by Levenshtein [4] for further details.

4 Perspectives on Asymptotics

We earlier saw the Gilbert-Varshanov bound, namely $R(\delta, q) \geq 1 - H_q(\delta)$ and saw that codes based on algebraic geometry out-performed this bound. However for the binary alphabet this is the best known lower bound on the rate of a code. Natural questions to ask at this stage are:

1. Is the GV bound tight (or are there codes that do better than the GV bound)?
2. Is the LP bound tight (or do there exist codes that meet the LP bound)?
3. Is the MRRW bound tight (or do there exist codes that meet the MRRW bound)?

Both the GV bound and MRRW bound are known functions and we know that they are not equal. So it is not possible to believe that both bounds are tight. However till recently none of the remaining possibilities even considered pairwize could be ruled out. I.e., it was conceivable that the LP bound and GV bound were both tight, or that the LP bound and MRRW bound were both tight. Recently Samorodnitsky [7] managed to rule out one of these possibilities — he showed that for every δ, the LP bound is at least the arithmetic mean of the GV bound and the MRRW bound. So if the GV bound is tight, then the LP bound can not prove this fact!

Is the GV bound tight for binary codes? Obviously this question can not be answered at the moment. Yet one could ask what the evidence so far suggests. On the one hand no better code has been found, and we could take this as evidence that the GV bound is tight. On the other hand we could ask if there are examples that suggest, without proving, that the GV bound does not appear to be tight. We have already seen one such example — namely the Reed-Solomon codes were doing "better than random" but were using alphabet sizes that were growing with the block length. Turning this intuition into a refutation of the q-ary GV-bound required all the heavy-duty machinery of AG-codes, but it did pay off. Two significant lessons could be learned from this success — (1) we need some example of an anamolous behavior as exhibited by the RS code, and (2) we need an asymptotic direction to exploit in our investigation. Are there other examples of anamolous behaviour? Turns out every example we have seen shows some anamoly.

Hadamard codes If the best we could do with code is to assume that the spheres of radius d around them do not overlap, then a binary code with relative distance $n/2$ should have only two codewords. The Hadamard codes with $\Omega(n)$ codewords are doing significantly better.

Hamming codes & BCH codes If we fix a distance d and consider $n - k$ for $[n, k, d]$-codes then the GV bound requires $n - k \geq d \log n$, while the BCH and Hamming codes only need $n - k \approx (d/2) \log n$. Thus once again, these codes outperform the GV principle.

So the next question ought to be: Is there an asymptotic sense in which one should try to exploit these examples. We describe two approaches below.

Relative Distance close to $\frac{1}{2}$ The MRRW bound, together with the GV bound, implies that the binary code of relative distance $\frac{1}{2} - \epsilon$ with highest possible rate has rate $\Theta(\epsilon^2)$. So in a certain asymptotic sense, the MRRW bound and the GV bound are already tight at this extreme. However the GV bound is non-constructive and constructive results are still lagging behind the GV/MRRW bound in this case. The best known constructive results yield codes of rate $O(\epsilon^3)$ with distance $\frac{1}{2} - \epsilon$. Some constructions that achieve this bound are:

- Reed-Solomon code concatenated with random linear code. (The reader should work this out as an exercise.)
- Algebraic Geometry code concatenated with Hadamard code. Such a construction was given by Katsman, Tsfasman and Vladuts [3] and could be attempted by the reader as an exercise.
- Codes derived from expander graphs due to Alon, Bruck, Naor, Naor, and Roth [1].

Improving these constructions would be a significant step towards achieving the GV bound constructively.

Rate close to 1 At rate close to 1, or as $\delta \to 0$, the GV bound says that $R \geq 1 - \delta \log(1/\delta)$. All bounds upto the Elias bound only show $R \geq 1 - (\delta/2)\log(2/\delta)$. In this regime the gap between the GV bound and the upper bounds seems maximal. This regime may offer one of the best options for "beating the GV bound" if this is possible at all.

References

[1] Noga Alon, Jehoshua Bruck, Joseph Naor, Moni Naor, and Ronny Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Transactions on Information Theory*, 38:509–516, 1992.

[2] Phillipe Delsarte. An algebraic approach to the association schemes of coding theory. *Philips Research Reports*, Suppl. 10, 1973.

[3] G. L. Katsman, Michael A. Tsfasman, and Serge G. Vlădut. Modular curves and codes with a polynomial construction. *IEEE Transactions on Information Theory*, 30:353–355, 1984.

[4] V. I. Levenshtein. *Universal bounds for codes and designs*, pages 499–648. 1998.

[5] Florence Jessie MacWilliams. A theorem on the distribution of weights in a systematic code. *Bell Systems Technical Journal*, 42:79–94, January 1963.

[6] Robert J. McEliece, Eugene R. Rodemich, Howard Rumsey Jr., and Lloyd R. Welch. New upper bounds on the rate of a code via the Delsarte-MacWilliams inequalities. *IEEE Transactions on Information Theory*, 23:157–166, 1977.

[7] Alex Samorodnitsky. *Ph.D. Thesis.* PhD thesis, Hebrew University, 1998.