Introduction to Coding Theory **CMU: Spring 2010**

Notes 5.1: Fourier Transform, MacWillams identities, and LP bound

February 2010

Lecturer: Venkatesan Guruswami *Scribe: Venkat Guruswami & Srivatsan Narayanan*

We will discuss the last and most sophisticated of our (upper) bounds on rate of codes with certain relative distance, namely the first linear programming bound or the first JPL bound due to McEliece, Rodemich, Rumsey, and Welch, 1977 (henceforth, MRRW). This bound is the best known asymptotic upper bound on the rate of a binary code for a significant range of relative distances (which is roughly $\delta \in (0.273, 1/2)$). We will present a complete and self-contained proof of the this bound. A variant called the second JPL bound gives the best known upper bound for the remainder of the range, and we will mention this bound (without proof) at the end.

The linear programming bound is so-called because it is based on Delsarte's linear programming approach which shows that the distance distribution of a binary code satisfies a family of linear constraints whose coefficients are the evaluations of a certain family of orthogonal polynomials (in this case, the Krawtchouk polynomials). The optimum (maximum) of this linear program gives an upper bound on $A(n, d)$. MRRW constructed good feasible solutions to the dual of linear program using tools from the theory of orthogonal polynomials, and their value gave an upper bound on $A(n, d)$ by weak duality.

In these notes, we will use Fourier analysis of functions defined on the hypercube to derive a relationship between the weight distribution of a linear code and its dual, called the MacWilliams identifies. These give the linear constraints of the above-mentioned linear program.

Instead of the using the linear program or its dual and the theory of orthogonal polynomials (and specifically properties of Krawtchouk polynomials), in the second part of these notes, we will give a *self-contained proof* of the first linear programming bound for binary linear codes using a Fourier analytic approach. This is based on the methods of Friedman and Tillich, which was later extended also to general codes by Navon and Samorodnitsky, that shows that the dual of a linear code of large distance must have small "essential covering radius" (which means that Hamming balls of small radii around the dual codewords will cover a large fraction of the Hamming space $\{0,1\}^n$). This shows that the dual must have large size, and therefore the code itself cannot be too large. The method can be extended to non-linear codes, but we will be content with deriving the linear programming bound for (binary) linear codes.

1 Fourier analysis over the Boolean hypercube

Let $\mathcal{F}_n$ be set of all real-valued functions over the boolean hypercube, *i.e.*, $\mathcal{F}_n = \{f : \{0,1\}^n \to \mathbb{R}\}$. Then the following characterization is straightforward.

Exercise 1 *Show that $\mathcal{F}_n$ forms a vector space with dimension 2^n. In fact, show that $\{e_\alpha : \alpha \in$*

$\{0,1\}^n\}$ *forms a basis for* $\mathcal{F}_n$*, where* $e_\alpha : \{0,1\}^n \to \mathbb{R}$ *is defined by:*

$$e_\alpha(x) = \delta_{x\alpha} = \begin{cases} 1, & \text{if} x = \alpha \\ 0, & \text{otherwise} \end{cases}$$

In fact, the above exercise views a function $f \in \mathcal{F}_n$ as simply a vector of dimension 2^n, indexed by the "coordinates" $\alpha \in \{0,1\}^n$. This motivates us to define an inner product on $\mathcal{F}_n$:

Definition 1 *For* $f, g \in \mathcal{F}_n$*, define the inner product between* f *and* g *to be:*

$$\langle f, g \rangle = \frac{1}{2^n} \sum_x f(x)g(x) = \mathbb{E}_x \left[f(x)g(x) \right]$$

(This is just the standard inner product for vectors over reals, but suitably normalized.)

Now, we will define another basis for $\mathcal{F}_n$, called the *Fourier basis*. This needs the following simple lemma.

Lemma 2 *For every binary linear code* $C \subseteq \{0,1\}^n$*,*

$$\sum_{c \in C} (-1)^{\alpha \cdot c} = \begin{cases} |C|, & \text{if } \alpha \in C^\perp, \\ 0, & \text{otherwise.} \end{cases}$$

where $\cdot$ *denotes the dot product modulo* 2*.*

Proof: If $\alpha \in C^\perp$, then the claim is obvious. Suppose that $\alpha \notin C^\perp$. Then, there exists a $c_0 \in C$ such that $\alpha \cdot c_0 = 1$. Now, for each $c \in C$,

$$(-1)^{\alpha \cdot c} + (-1)^{\alpha \cdot (c + c_0)} = (-1)^{\alpha \cdot c} \left(1 + (-1)^{\alpha \cdot c_0}\right) = 0 \tag{1}$$

Summing Equation 1 for all $c \in C$, we get:

$$0 = \sum_{c \in C} \left((-1)^{\alpha \cdot c} + (-1)^{\alpha \cdot (c + c_0)} \right) = \sum_{c \in C} (-1)^{\alpha \cdot c} + \sum_{c \in C} (-1)^{\alpha \cdot (c + c_0)} = 2 \sum_{c \in C} (-1)^{\alpha \cdot c},$$

giving the claim. □

Corollary 3 *We have*

$$\sum_{c \in \{0,1\}^n} (-1)^{\alpha \cdot c} = \begin{cases} 2^n, & \text{if } \alpha = 0, \\ 0, & \text{otherwise.} \end{cases}$$

Proof: In Lemma 2, take C to be the whole vector space $\{0,1\}^n$, so that $C^\perp = \{0\}$. □

Remark 4 *In this lecture, the notation* 0 *is typically overloaded to mean either a single alphabet symbol, or the zero vector* (0^n) *of the vector space. However, the right definition should be clear from the context.*

For each $\alpha \in \{0,1\}^n$, define $\chi_\alpha : \{0,1\}^n \to \mathbb{R}$ by $\chi_\alpha(x) = (-1)^{\alpha \cdot x}$ (where $\cdot$ refers to the inner product between vectors, taken modulo 2). The function χ_α is often called a *character function*. We show that the set of all character functions also forms an orthonormal basis for $\mathcal{F}_n$.

Lemma 5 $\langle \chi_\alpha, \chi_\beta \rangle = \delta_{\alpha\beta}$

Proof:

$$\langle \chi_\alpha, \chi_\beta \rangle = \mathbb{E}_x \left[(-1)^{\alpha \cdot x} (-1)^{\beta \cdot x} \right] = \frac{1}{2^n} \sum_x (-1)^{(\alpha - \beta) \cdot x} = \begin{cases} 1, & \text{if } \alpha - \beta = 0, \\ 0, & \text{otherwise} \end{cases}$$

using Corollary 3. The claim follows from the definition of $\delta_{\alpha\beta}$. □

Corollary 6 *Let* B *be the set of character functions, i.e.,* $B = \{\chi_\alpha : \alpha \in \{0,1\}^n\}$. *Then,* B *is an orthonormal basis for* $\mathcal{F}_n$, *called its Fourier basis.*

Proof: From Lemma 5, it follows that B is a linearly independent set. Also the cardinality of B is 2^n, which equals the dimension of the whole space $\mathcal{F}_n$. Therefore, B must be a basis. The orthonormality of B is directly implied again by Lemma 5. □

By the definition of a basis, any function $f \in \mathcal{F}_n$ can be expressed uniquely as a linear combination of the character functions. That is, there exist $\hat{f}(\alpha) \in \mathbb{R}$ such that

$$f = \sum_\alpha \hat{f}(\alpha) \chi_\alpha \ .$$

(We use the notation $\hat{f}(\alpha)$, instead of the conventional c_α to remind us that the coefficients depend on f.) Note that this is equivalent to saying

$$f(x) = \sum_\alpha \hat{f}(\alpha) \chi_\alpha(x)$$

for all $x \in \{0,1\}^n$.

The following are some immediate consequences of this fact.

Lemma 7 *Let* $f, g \in \mathcal{F}_n$. *Then the following hold.*

1. $\langle f, \chi_\alpha \rangle = \hat{f}(\alpha)$
2. *(Parseval's identity)* $\langle f, g \rangle = \sum_\alpha \hat{f}(\alpha) \hat{g}(\alpha)$
3. $\hat{f}(0) = \mathbb{E}_x f(x)$

PROOF: Each of the above claims can be shown by a straightforward calculation.

1. $\langle f, \chi_\alpha \rangle = \langle \sum_\beta \hat{f}(\beta)\chi_\beta, \chi_\alpha \rangle = \sum_\beta \hat{f}(\beta)\langle \chi_\beta, \chi_\alpha \rangle = \sum_\beta \hat{f}(\beta)\delta_{\alpha\beta} = \hat{f}(\alpha)$
2. $\langle f, g \rangle = \langle f, \sum_\alpha \hat{g}(\alpha)\chi_\alpha \rangle = \sum_\alpha \hat{g}(\alpha)\langle f, \chi_\alpha \rangle = \sum_\alpha \hat{f}(\alpha)\hat{g}(\alpha)$
3. $\hat{f}(0) = \langle f, \chi_0 \rangle = \mathbb{E}_x \left[f(x)(-1)^{0 \cdot x} \right] = \mathbb{E}_x f(x)$

□

2 Dual codes, Fourier analysis, and MacWilliams identities

Let us introduce the following notation: for any $S \subseteq \{0,1\}^n$, define $1_S : \{0,1\}^n \to \mathbb{R}$, called the *characteristic function of* S, by

$$1_S(x) = \begin{cases} 1, & \text{if } x \in S, \\ 0, & \text{otherwise.} \end{cases}$$

We will now show that Fourier transform of the characteristic function of a code is *essentially the same* (up to a constant scaling factor) as the characteristic function of its dual. This is useful because the Fourier transform can be viewed as a notion of duality for functions. Fortunately, there is a natural correspondence between the two notions (dual codes and Fourier transforms).

Lemma 8 *For any linear code* $C \subseteq \{0,1\}^n$,

$$\widehat{1_C} = \frac{|C|}{2^n} 1_{C^\perp}$$

PROOF: For every $\alpha \in \{0,1\}^n$,

$$\widehat{1_C}(\alpha) = \langle 1_C, \chi_\alpha \rangle = \frac{1}{2^n} \sum_x 1_C(x)\chi_\alpha(x) = \frac{1}{2^n} \sum_{x \in C} (-1)^{\alpha \cdot x} = \begin{cases} \frac{1}{2^n}|C|, & \text{if } \alpha \in C^\perp, \\ 0, & \text{otherwise} \end{cases}$$

using Lemma 2. Therefore,

$$\widehat{1_C}(\alpha) = \frac{|C|}{2^n} 1_{C^\perp}(\alpha),$$

for all $\alpha \in \{0,1\}^n$, giving the claim. □

Definition 9 *For any* $S \subseteq \{0,1\}^n$, *let*

$$W_i^S = \#\{x \in S : \mathrm{wt}(x) = i\},$$

that is, W_i^S *denotes the number of points in* S *of weight* i. *Further, by weight distribution of* S, *we denote the* $(n+1)$*-tuple* $W^S = \langle W_0^S, W_1^S, \ldots, W_n^S \rangle$.

Now, our goal is to relate the "weight distribution" of a code C to that of its dual $C^\perp$. Let $\ell \in \{0, 1, \ldots, n\}$. Then,

$$\begin{aligned}
W_\ell^{C^\perp} &= \sum_{\alpha:\mathrm{wt}(\alpha)=\ell} 1_{C^\perp}(\alpha) \\
&= \frac{2^n}{|C|} \sum_{\alpha:\mathrm{wt}(\alpha)=\ell} \widehat{1_C}(\alpha) \\
&= \frac{2^n}{|C|} \sum_{\alpha:\mathrm{wt}(\alpha)=\ell} \mathbb{E}_x \left[1_C(x)(-1)^{\alpha\cdot x}\right] \\
&= \frac{2^n}{|C|} \mathbb{E}_x \left[\sum_{\alpha:\mathrm{wt}(\alpha)=\ell} 1_C(x)(-1)^{\alpha\cdot x}\right] \\
&= \frac{2^n}{|C|} \mathbb{E}_x \left[1_C(x) \sum_{\alpha:\mathrm{wt}(\alpha)=\ell} (-1)^{\alpha\cdot x}\right]
\end{aligned}$$

For completeness, we calculate the sum $\sum_{\alpha:\mathrm{wt}(\alpha)=\ell}(-1)^{\alpha\cdot x}$ in the following lemma. The exact sum is not of any significance for our purposes in this course. We will however use the fact that this sum depends *only* on the weight of x.

Lemma 10 *For any $x \in \{0,1\}^n$ with $\mathrm{wt}(x) = i$,*

$$\sum_{\alpha:\mathrm{wt}(\alpha)=\ell} (-1)^{\alpha\cdot x} = \sum_{j=0}^{\ell} (-1)^j \binom{i}{j}\binom{n-i}{\ell-j} .$$

The latter quantity will be denoted as $K_\ell(i)$ — the value of the Krawtchouk polynomial at i.

PROOF: Notice that summation is taken over all α of a given weight ℓ. So, by symmetry, it depends only the number of 1's in x, and not on their positions. Hence, without any loss in generality, assume that $x = 1^i 0^{n-i}$. A vector α of weight ℓ must have j 1's in the first i positions, and $\ell - j$ in the last $n-i$ positions, for some $j \in \{0, 1, \ldots, \ell\}$, and in this case $(-1)^{x\cdot\alpha} = (-1)^j$. The number of α's satisfying this condition for any particular $j \in \{0, 1, \ldots, \ell\}$ equals $\binom{i}{j}\binom{n-i}{\ell-j}$. The claim thus follows. □

Remark 11 (Krawtchouk polynomial) *The quantity $\sum_{j=0}^{\ell}(-1)^j\binom{i}{j}\binom{n-i}{\ell-j}$, denoted $K_\ell(i)$, can be regarded as the evaluation of a polynomial K_ℓ at $\mathrm{wt}(x) = i$. K_ℓ is usually called the ℓ^{th} Krawtchouk polynomial and is defined as*

$$K_\ell(X) = \sum_{j=0}^{\ell} (-1)^j \binom{X}{j}\binom{n-X}{\ell-j} .$$

(The function K_ℓ also depends on n, but we supress this dependence for notational convenience.) Note that K_ℓ is a polynomial of degree ℓ and $K_0(X) = 1$ and $K_1(X) = n - 2X$, etc.

Now, we will complete the calculation of $W_\ell^{C^\perp}$.

$$\begin{aligned} W_\ell^{C^\perp} &= \frac{2^n}{|C|}\frac{1}{2^n}\sum_x \left[1_C(x)\sum_{\alpha:\mathrm{wt}(\alpha)=\ell}(-1)^{\alpha\cdot x}\right] \\ &= \frac{1}{|C|}\sum_x 1_C(x)K_\ell(\mathrm{wt}(x)) \\ &= \frac{1}{|C|}\sum_{x\in C} K_\ell(\mathrm{wt}(x)) \\ &= \frac{1}{|C|}\sum_{i=0}^{n}\sum_{x\in C,\mathrm{wt}(x)=i} K_\ell(i) \end{aligned}$$

giving

$$W_\ell^{C^\perp} = \frac{1}{|C|}\sum_{i=0}^{n} W_i^C K_\ell(i) \tag{2}$$

for every $\ell = 0, 1, 2, \ldots, n$.

Equation 2, called the MacWilliams identity, tells us that the weight distribution of the dual code $C^\perp$ is completely determined once we are given the weight distribution of the code C.

Remark 12 *We can write the MacWilliams identitities (2) equivalently as:*

$$W_\ell^{C^\perp} = \mathbb{E}_{x\in C}\left[K_\ell(\mathrm{wt}(x))\right] \ ,$$

or as a functional equation

$$\sum_{\ell=0}^{n} W_\ell^{C^\perp} z^\ell = \frac{1}{|C|}\sum_{i=0}^{n} W_i^C (1-z)^i (1+z)^{n-i} \ .$$

Exercise 2 *Extend the MacWilliams identities to linear codes over any finite field $\mathbb{F}_q$. Specifically, if C is a q-ary linear code of block length n, and as before W_i^C (resp. $W_i^{C^\perp}$) denote the number of codewords of C (resp. $C^\perp$) of Hamming weight i, then*

$$W_\ell^{C^\perp} = \frac{1}{|C|}\sum_{i=0}^{n} W_i^C K_\ell^{(q)}(i)$$

where the q-ary Krawtchouk polynomial is defined as

$$K_\ell^{(q)}(X) = \sum_{j=0}^{\ell}(-1)^j (q-1)^{\ell-j}\binom{X}{j}\binom{n-X}{\ell-j} \ .$$

[[**Hint***: When the field size q equals a prime p, replace $(-1)^{x\cdot y}$ in the proof for the binary case by $\zeta_p^{x\cdot y}$ where $\zeta_p = e^{2\pi i/p}$ is a primitive p'th root of unity and $x\cdot y$ is, as usual, computed over the underlying field $\mathbb{F}_q$.*

When $q = p^t$ for a prime p, the role of $(-1)^{x \cdot y}$ can be played by $\zeta_p^{\mathrm{Tr}(x \cdot y)}$ where Tr *is the trace map from $\mathbb{F}_q$ to $\mathbb{F}_p = \{0, 1, 2 \ldots, p-1\}$: $\mathrm{Tr}(z) = z + z^p + \cdots + z^{p^{t-1}}$.]]*

Exercise 3 *Using the above, compute the weight distribution of the $[q^m - 1, q^m - 1 - m, 3]_q$ Hamming code.*

3 A linear program bounding $A(n, d)$

In this section, we will use the MacWilliams identity to derive a linear program that bounds the size of every code with a given minimum distance d, and thus bounds $A(n, d)$. (Recall that $A(n, d)$ is the maximum size of any binary code with block length n and minimum distance d.)

For the moment, we will focus on linear codes C. Consider the linear program:

$$\begin{aligned}
\text{Maximize} \quad & \sum_{i=0}^{n} A_i & & & \\
\text{s.t.} \quad & A_0 & = & 1 & \\
& A_i & \geq & 0, & \quad i = 1, \ldots, n \\
& A_i & = & 0, & \quad i = 1, \ldots, d-1 \\
& \sum_{i=0}^{n} K_\ell(i) A_i & \geq & 0, & \quad \ell = 1, \ldots, n
\end{aligned}$$

We claim that for any *linear* code C of distance at least d, the assignment $A_i = W_i^C$ is a feasible solution. Indeed, the first two constraints are satisfied trivially. The constraint $A_i = 0$ for $1 \leq i < d$ enforces that the minimum distance of the code (that is, the minimum Hamming weight of any nonzero code word) is at least d. The last set of constraints follow from the MacWilliams identities for any $\ell \in \{1, 2, \ldots, n\}$,

$$\sum_{i=0}^{n} W_i^C K_\ell(i) = W_\ell^{C^\perp} \geq 0$$

For this assignment, the objective function takes the value

$$\sum_{i=0}^{n} W_i^C = |C|$$

Therefore, the optimum of the linear program upper bounds the size of any linear code C of distance at least d.

Now, we consider general codes C, and prove that they satisfy the same bound. Without loss of generality, assume that $0^n \in C$. Define:

$$A_i^C = \frac{\#\{(x, y) \in C^2 \mid \Delta(x, y) = i\}}{|C|}$$

We claim that A_i^C is a feasible solution to the linear program. The first three sets of constraints are trivially satisfied as before, whereas the last set of constraints can be verified in a straightforward manner:

$$\begin{aligned}
\sum_{i=0}^{n} A_i^C K_\ell(i) &= \frac{1}{|C|}\sum_{i=0}^{n}\sum_{(x,y)\in C^2:\Delta(x,y)=i} K_\ell(i) \\
&= \frac{1}{|C|}\sum_{i=0}^{n}\left(\sum_{(x,y)\in C^2:\Delta(x,y)=i}\left(\sum_{z:\mathrm{wt}(z)=\ell}(-1)^{(x-y)\cdot z}\right)\right) \\
&= \frac{1}{|C|}\sum_{(x,y)\in C^2}\left(\sum_{z:\mathrm{wt}(z)=\ell}(-1)^{(x-y)\cdot z}\right) \\
&= \frac{1}{|C|}\sum_{z:\mathrm{wt}(z)=\ell}\left(\sum_{(x,y)\in C^2}(-1)^{x\cdot z}(-1)^{y\cdot z}\right) \\
&= \frac{1}{|C|}\sum_{z:\mathrm{wt}(z)=\ell}\left(\sum_{x\in C}(-1)^{x\cdot z}\right)\left(\sum_{y\in C}(-1)^{y\cdot z}\right) \\
&= \frac{1}{|C|}\sum_{z:\mathrm{wt}(z)=\ell}\left(\sum_{x\in C}(-1)^{x\cdot z}\right)^2 \\
&\geq 0
\end{aligned}$$

The value of the objective function is:

$$\sum_{i=0}^{n} A_i^C = \frac{1}{|C|}\sum_{i=0}^{n}\left(\sum_{(x,y)\in C^2:\Delta(x,y)=i} 1\right) = \frac{1}{|C|}\sum_{(x,y)\in C^2} 1 = |C|$$

Therefore, the optimum value of the linear program upper bounds the size of any code with minimum distance at least d.

3.1 Dual program and the MRRW bound

Consider the dual program for the above linear program. The dual program has variables $\beta_1, \beta_2, \ldots, \beta_n$ (where $\beta_i \geq 0$). Define $\beta(X)$ to be the polynomial

$$\beta(X) = 1 + \sum_{\ell=0}^{n} \beta_\ell K_\ell(X) \ .$$

Then the dual program is given by:

$$\begin{aligned}
\text{Minimize} \quad & \beta(0) \\
\text{s.t.} \quad \beta_i &\geq 0, \qquad i = 1, 2, \ldots, n \\
\beta(j) &\leq 0, \qquad j = d, \ldots, n
\end{aligned}$$

By the weak duality theorem, the value of *any* feasible solution to the dual program upper bounds the optimum value of the linear program, and hence also upper bounds $A(n, d)$. Hence, in order to upper bound the size of the code, it suffices to exhibit a dual feasible solution with a small objective function. This was, in fact, the approach followed by MRRW, leading to the first linear programming bound. However, this involves studying several properties of Krawtchouk polynomials. In the second installment of these notes, we will prove the same bound by following a different approach based on Fourier analysis.