The Hamming Codes and Delsarte's Linear Programming Bound

by

Sky McKinley

Under the Astute Tutelage of

Professor John S. Caughman, IV

A thesis submitted in partial fulfillment of the requirements for

the degree of

Master of Science

in

Mathematics

Portland State University

May 2003

Contents

Chapter 1

Introduction

1.1 Summary

Two of the main goals of coding theory are to find and classify useful codes. In this paper we examine the structure of one of the most famous of code families, the Hamming codes. The central result derived in this paper is one of the strongest and most sophisticated of the bounds restricting the sizes of codes, Delsarte's linear programming bound, developed in Delsarte's thesis [3]. This bound is a result in the theory of association schemes, so we include a brief introduction to association schemes. We explore the application of the theory of association schemes to coding theory, in particular to the Hamming codes.

1.2 Motivation for Coding Theory

When data is transmitted over large distances or noisy channels, it becomes necessary to ensure that the data sent is correctly delivered. How can someone on the receiving end of a data channel be assured that the information they receive is the data which was transmitted? And, if data is corrupted, can the correct information be reconstructed using what was received? The field of study known as coding theory attempts to answer these questions. Special constructs called *codes* enable the receiving party to determine if information has been correctly received and, under certain circumstances, allow the receiver to fix information which has been corrupted.

As an example, consider a scenario where a message using words $\{0, 1\}$ is sent across a noisy channel to a receiver. Suppose that there is a 1% chance that, in transmission, an error is introduced that switches a 1 to a 0 or vice-versa. If such a switch occurs, there is no way that the receiver can detect the error, which is clearly undesirable. We can adapt to this by sending each bit twice. The codewords in this new code are the strings of length 2 over the field of two elements giving a code $C_1 = \{00, 11\}$, out of the set of 4 possible words of length two. This code allows the receiver to detect whether or not a single transmission error has occurred. For example, if the word received is 01, which is not a codeword, then the receiver knows that one of the elements has changed and can request

retransmission. The difficulty is not completely resolved, however. Two errors can occur which transform a codeword to another codeword, say 11 to 00, and such an error cannot be detected. The odds of two errors occurring, though, are only .01%, and this is clearly better than the original scheme. Note, however, that even if an error is detected, the receiver cannot be sure of which bit has been switched. The word sent could have been 00 or 11, and so although the receiver can request retransmission, there is no way for the receiver to independently correct the error. On the other hand, if the code used is $C_2 = \{000, 111\}$, then the receiver could use a "majority rules" decoding scheme where 111, 101, 011, and 110 would all be decoded as 1, and 000, 001, 010, and 100 would all be decoded as 0. Notice that any single error can not only be detected, the word which was sent can be recovered. Decoding in this way is called *nearest-neighbor decoding*. The codewords are longer in this coding scheme, and so information will take longer to transmit. One of the main goals of coding theory is to determine a precise relationship between the length of the code words and the ability to detect and correct a fixed number of errors.

So good codes must have special properties: the codewords must be long enough to make detecting errors possible but they should be short enough to ensure efficient transmission. Bounds on the sizes of codes relative to code word length are invaluable, and one of the strongest such bounds is the linear programming bound developed by Phillipe Delsarte in 1973 [3]. The search for "optimal" codes has led

to the development of a family of codes known as the Hamming codes, which have some very beneficial encoding and decoding properties.

In this paper, we introduce the Hamming codes and examine them using the algebraic structure of association schemes. It is in this framework that we prove the linear programming bound, which gives us useful information about the relationship between the size of a code and the length of the codewords needed to achieve the desired error correction capacity. We begin with an overview of the basic ideas of coding theory.

Chapter 2

Hamming Codes

2.1 Foundations of Coding Theory

If X is a finite set, then a non-empty subset C of X is called a *code*. Often X is the set of n-tuples from a finite alphabet K with q elements. The elements of X are called *words* and the elements of C are called *codewords*. When K is a field, X is an n-dimensional vector space over K. In this case, C is called a *linear code* if C is a linear subspace of X. When $K = F_q$, the finite field of q elements, then q will be a prime power and X will be denoted $V(n, q)$.

Given an (un-encoded) source message, an encoded message can be produced by breaking the source message into blocks of length m and encoding these blocks as codewords of length n. Such a code is called a *block code*. Thus for each n

symbols transmitted, only m symbols of the message are transmitted. We define the *transmission rate* ρ of such a code C as

$$\rho(C) = \frac{m}{n}.$$

In the case of a linear code C, m is the dimension of C as a subspace of $V(n, q)$.

Given a set X of words over a finite alphabet K, it is customary to assume that K has at least the structure of an Abelian group. We use additive notation to denote the group operation, so 0 will denote the identity of K. All operations on X will be done entry-wise. The *weight* $w(\mathbf{x})$, also called the *Hamming weight*, of a word $\mathbf{w} \in X$ is the number of non-zero entries in $\mathbf{w}$, and the *distance* $d(\mathbf{w_1}, \mathbf{w_2})$, also called the *Hamming metric*, between two words $\mathbf{w_1}, \mathbf{w_2}$ in X is number of positions in which $\mathbf{w_1}$ and $\mathbf{w_2}$ differ, denoted $d(\mathbf{w_1}, \mathbf{w_2})$. In other words, the distance between two code words $\mathbf{w_1}$ and $\mathbf{w_2}$ will be the weight of $\mathbf{w_1} - \mathbf{w_2}$. The notion of *nearest neighbor decoding* refers to decoding a received word $\mathbf{w} \in X$ as a nearest codeword in C relative to the Hamming metric. The following lemma establishes that the Hamming metric is indeed a metric on X.

2.1.1 Lemma. *The set X of n-tuples from a finite alphabet K along with the Hamming metric is a metric space.*

Proof: To verify that the Hamming metric $d : X \times X \to \mathbb{R}^+$ is a metric, we must show that

1. $d(\mathbf{w}, \mathbf{v}) = 0$ if and only if $\mathbf{w} = \mathbf{v}$,

2. $d(\mathbf{w}, \mathbf{v}) = d(\mathbf{v}, \mathbf{w})$, and

3. $d(\mathbf{u}, \mathbf{w}) \leq d(\mathbf{u}, \mathbf{v}) + d(\mathbf{v}, \mathbf{w})$ for all $\mathbf{u}, \mathbf{v}, \mathbf{w} \in X$.

Clearly, a word $\mathbf{w}$ differs from itself in 0 places, and so $d(\mathbf{w}, \mathbf{w}) = 0$. Also, if $d(\mathbf{w}, \mathbf{v}) = 0$, then each entry in $\mathbf{w}$ is equal to the corresponding entry in $\mathbf{v}$, and so $\mathbf{w} = \mathbf{v}$. It is obvious that $d(\mathbf{w}, \mathbf{v}) = d(\mathbf{v}, \mathbf{w})$ for all $\mathbf{w}$ and $\mathbf{v}$. To see that $d(\mathbf{u}, \mathbf{w}) \leq d(\mathbf{u}, \mathbf{v}) + d(\mathbf{v}, \mathbf{w})$ for all $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ in X, note that for each position where $\mathbf{u}$ differs from $\mathbf{w}$, $\mathbf{v}$ must differ from at least one of $\mathbf{w}$ or $\mathbf{u}$. So for each position where $\mathbf{u}$ and $\mathbf{w}$ differ, and thus a 1 is contributed to $d(\mathbf{u}, \mathbf{w})$, a 1 will be contributed to at least one of $d(\mathbf{u}, \mathbf{v})$ or $d(\mathbf{v}, \mathbf{w})$. Thus the triangle inequality holds, and so X, along with the Hamming metric, is a metric space. **Done.**

Given a code C, the *diameter* Δ of a code is the maximum distance between any two codewords in C. The *minimum distance* $\delta = \delta(C)$ of a code C is the smallest distance between distinct codewords, and the *minimum weight* $\omega = \omega(C)$ is the smallest weight of a nonzero codeword. If C is a linear code with dimension m and minimum distance δ in the space $V(n, q)$, it is called a q-ary (n, m, δ)-code. Sometimes the notation $[n, \delta]$-code is used, as when the dimension is irrelevant or not applicable.

The minimum distance of a code is of particular interest in coding theory, as

this quantity determines how many errors that code can detect or correct. In particular, we have the following.

2.1.2 Lemma. *Suppose X is the set of n-tuples from a finite alphabet K. Assume that K has the structure of an Abelian group and let $d(\mathbf{x}, \mathbf{y})$ denote the Hamming metric on X. If the minimum distance δ of a code C satisfies $\delta \geq 2r + 1$ for some $r \in \mathbb{N}$, then nearest-neighbor decoding can correct transmission errors involving $t \leq r$ entries.*

Proof: Suppose a codeword $\mathbf{c}$ is transmitted and the word $\mathbf{w}$ is received with $t \leq r$ errors occurring, that is $d(\mathbf{w}, \mathbf{c}) = t \leq r$. To show that C is r-error correcting, we must show that $d(\mathbf{c}', \mathbf{w}) > r$ for all $\mathbf{c}' \neq \mathbf{c}$ in C. The received word $\mathbf{w}$ differs from $\mathbf{c}$ in exactly t places, and so

$$\begin{aligned} d(\mathbf{w}, \mathbf{c}') &\geq d(\mathbf{c}', \mathbf{c}) - d(\mathbf{w}, \mathbf{c}) \\ &\geq 2r + 1 - r \\ &\geq r + 1 \\ &> r \end{aligned}$$

Thus $\mathbf{w}$ is distance at most r from exactly one codeword, namely $\mathbf{c}$, and so nearest-neighbor decoding will correct up to r transmission errors. **Done.**

The advantage of working with linear codes involves computational considerations. If a code C is non-linear, then all $\binom{|C|}{2}$ distances between distinct codewords

of C must be checked to determine the minimum distance $\delta(C)$. If C is a linear code, however, $\delta(C)$ equals the minimum weight $\omega(C)$ of the code. Finding $\omega(C)$ only involves $|C| - 1$ calculations, one for each non-zero codeword, and so is much faster. This is because, when C is linear, the difference between any two codewords is again a codeword, and so the distances $d(\mathbf{c_1}, \mathbf{c_2})$ become weights $w(\mathbf{c_1} - \mathbf{c_2})$, where $\mathbf{c_1} - \mathbf{c_2} \neq \mathbf{0}$.

A code $C \subseteq X$ is said to *cover* X with radius r if for all words $\mathbf{x} \in X$ there is a codeword $\mathbf{c}$ in C with $d(\mathbf{x}, \mathbf{c}) \leq r$. A code C is said to be *r-perfect* if each word in X is distance less than r from exactly one codeword. Such codes are occasionally referred to as *perfect r-codes*. As an example, consider the code $C_2 = \{000, 111\}$ from the introduction as a linear code in $V(3, 2)$. Any other word in $V(3, 2)$ is distance one from exactly one of 000 or 111, and so $C_2 = \{000, 111\}$ is a perfect 1-code.

Given a particular space X, a natural question is how large a code can be defined on X with minimum distance δ? One of the more well known bounds on such a code is the *Hamming bound* or *sphere packing bound*.

2.1.3 Theorem. *(Hamming Bound) Let X be the set of n-tuples over a finite alphabet K with q elements. Assume as above that K has the structure of an Abelian group. If C is a code in X with minimum distance $\delta \geq 2r + 1$ for some*

$r \geq 0$, *then*

$$|C| \sum_{i=0}^{r} \binom{n}{i} (q-1)^i \leq q^n \tag{2.1}$$

Proof: Note that $|X| = q^n$. Now let $\mathbf{c}$ be a codeword in C. Then for a word in X to be at distance i from $\mathbf{c}$ it must differ from $\mathbf{c}$ in exactly i places. There are $\binom{n}{i}$ choices for which i of the n entries are different and for each of these i entries there are $q-1$ letters different from the one in $\mathbf{c}$. Thus for each distance i there are $\binom{n}{i}(q-1)^i$ words at distance i from $\mathbf{c}$. So the sum on the left of (2.1) counts the number of words in a ball $B_r(\mathbf{c})$ of radius r centered at a codeword $\mathbf{c}$. Since $\delta \geq 2r+1$, these balls are disjoint as $\mathbf{c}$ ranges over the code C, so

$$|\cup_{\mathbf{c}\in C} B(\mathbf{c})| = |C| \sum_{i=0}^{r} \binom{n}{i} (q-1)^i. \tag{2.2}$$

But clearly $|\cup_{\mathbf{c}\in C} B_r(\mathbf{c})| \leq |X|$, and so (2.1) follows. **Done.**

2.1.4 Corollary. *Equality holds in Theorem 2.1.3 if and only if C is a perfect $(\delta-1)/2$-code.*

Proof: Let $r := (\delta-1)/2$. Note that by the definition of r, the balls around distinct codewords are disjoint. If C is a perfect r-code, then the balls of radius r centered at the codewords of C partition X, and so equality holds. If C is not perfect, then the balls of radius $(\delta-1)/2$ centered at the words of C do not cover X and so strict inequality must hold. **Done.**

2.2 Generating and Parity Check Matrices

Let C denote a linear code in $V(n,q)$. Let G be a matrix whose rows generate (span) C. The matrix G is called a *generating matrix* of C. The *dual code* of C, denoted $C^{\perp}$, is defined to be the set

$$C^{\perp} = \{\mathbf{x} \in V(n,q) \,:\, \langle \mathbf{x}, \mathbf{c} \rangle = \mathbf{0} \;\forall\; \mathbf{c} \;\in C\} \tag{2.3}$$

where $\langle \mathbf{u}, \mathbf{v} \rangle := u_1v_1 + u_2v_2 + \cdots u_nv_n$. Note that $C^{\perp}$ is clearly also a linear code, and thus has a generating matrix H. By the definition of $C^{\perp}$, it can be seen that

$$C = \{\mathbf{c} \in V(n,q) : \mathbf{c}H^t = 0\}. \tag{2.4}$$

The matrix H is called a *parity check matrix* for C. If a word $\mathbf{w}$ is received, then it can be verified that $\mathbf{w}$ is a codeword simply by checking that $\mathbf{w}H^t = 0$.

2.3 The Binary Hamming Code

In this section we introduce the "classical" Hamming code which is a linear code in $V(n,2)$ for some $n \geq 2$. We refer to such a code as a *binary* Hamming code. Let F_2 denote the field of two elements and let H be the matrix whose columns are all the non-zero vectors of length k over F_2, for some $k \in \mathbb{N}$. Note that there will be $2^k - 1$ of these. Clearly any two columns of H are linearly independent, as different columns will have 1's in different places. We define the binary Hamming

code as follows:

2.3.1 Definition. Fix $k \geq 2$ and let $n = 2^k - 1$. Let H denote the $k \times n$ matrix defined above. The (binary) Hamming code $\mathrm{Ham}_2(n)$ is the linear subspace of $V(n, 2)$ consisting of the set of all vectors orthogonal to all the rows of H. That is,

$$\mathrm{Ham}_2(n) = \{\mathbf{v} \in V(n, 2) \,:\, \mathbf{v}\, H^t = 0\}. \tag{2.5}$$

2.3.2 Proposition. *The binary Hamming code $Ham_2(n)$ with $k \times (2^k - 1)$ parity check matrix H is a $(2^k - 1, 2^k - k - 1, 3)$-code.*

Proof: That the length of the vectors in $\mathrm{Ham}_2(n)$ is $2^k - 1$ is clear. The code $\mathrm{Ham}_2(n)$ is defined to be the subspace of $V(n, 2)$ orthogonal to the rowspace of H, which has dimension k, and so the dimension of $\mathrm{Ham}_2(n)$ will be $2^k - k - 1$ by the rank-nullity theorem. By definition, no two columns of H are dependent, but since the vectors $\mathbf{u} = \langle 1, 0, \ldots, 0\rangle^t$, $\mathbf{v} = \langle 0, 1, 0, \ldots, 0\rangle^t$, and $\mathbf{w} = \langle 1, 1, 0, \ldots, 0\rangle^t$ are among the columns of H, there exist three columns in H which are linearly dependent. This implies that the code generated will have minimum distance 3. To see this, recall that for a linear code, the minimum distance is equivalent to the minimum weight of a codeword. Suppose columns i, j, and k of H are linearly dependent. Then some linear combination of those three columns with non-zero coefficients will equal zero, and since the vectors are taken over F_2, the coefficients

must be 1. So the vector with 1's in the i, j, and k position is in $\mathrm{Ham}_2(n)$, and so the minimum weight of the code is at most 3. It cannot be less than 3, or else some linear combination of two columns of H would be zero, which we have ruled out. Thus H will be the parity check matrix for a $(2^k - 1, 2^k - k - 1, 3)$-code. **Done.**

From the definition of transmission rate, we have the following corollary:

2.3.3 Corollary. *Let $n = 2^k - 1$ for some $k \geq 2$. Then the binary Hamming code $Ham_2(n)$ is a perfect 1-code and has transmission rate*

$$\rho(Ham_2(n)) = \frac{2^k - k - 1}{2^k - 1} \tag{2.6}$$

Proof: $\mathrm{Ham}_2(n)$ is a perfect 1-code by Corollary 2.1.4, and the transmission rate is immediate from the definition of $\mathrm{Ham}_2(n)$. **Done.**

2.4 The q-ary Hamming Code

In this section we consider the natural generalization of the binary Hamming code to an arbitrary finite field. For any positive integer m and prime p, let F_q denote the finite field with $q = p^m$.

In the previous section we constructed the binary Hamming code in $V(n, 2)$ using a matrix whose columns were all the non-zero vectors in $V(n, 2)$. Because the

field of scalars was F_2, this implied that the columns were pairwise independent. To construct the Hamming code over an arbitrary finite field F_q, we must modify this construction, since the non-zero vectors over F_q are not pairwise linearly independent. We begin with the following lemma.

2.4.1 Lemma. *Let $k \in \mathbb{N}$ with $k \geq 2$, and let q be a prime power. Consider the vector space $V(k,q)$ over the finite field F_q. Let $n = \frac{q^k-1}{q-1}$. Then the following hold:*

1. *The number of 1-dimensional subspaces of $V(k,q)$ is n.*
2. *There exists a $k \times n$ matrix over F_q whose columns are pairwise linearly independent.*

Proof: To show (1), we note that the number of non-zero vectors in $V(k,q)$ is $q^k - 1$. However, for each non-zero vector $\mathbf{v}$, any of the $q-1$ non-zero multiples of $\mathbf{v}$ is also a non-zero vector and lies in the same one-dimensional subspace as $\mathbf{v}$. Thus $\frac{q^k-1}{q-1}$ counts each one-dimensional subspace of $V(k,q)$. By selecting one non-zero vector from each of the 1-dimensional subspaces of $V(n,q)$ and taking these vectors to be the columns of $k \times n$ matrix, we see that claim (2) follows immediately from claim (1). **Done.**

We construct the q-ary Hamming code as follows: For $k \geq 2$ and prime power q, let $n = (q^k - 1)/(q-1)$. We construct the q-ary Hamming code by finding a

$k \times n$ matrix H with entries in F_q whose columns are pairwise linearly independent.

2.4.2 Definition. Let $n = (q^k - 1)/(q - 1)$ for some $k \geq 2$ and a prime power q. The *q-ary Hamming code $Ham_q(n)$* with parity check matrix H is the set of vectors in $V(n, q)$ orthogonal to H. That is,

$$\mathrm{Ham}_q(n) = \{\mathbf{v} \in V(n, q) : \mathbf{v}\, H^t = 0\} \tag{2.7}$$

The following theorem characterizes the q-ary Hamming codes.

2.4.3 Theorem. *Fix any positive integer $k \geq 2$ and prime power q. Let $n = \frac{q^k - 1}{q - 1}$. The q-ary Hamming code $\mathrm{Ham}_q(n)$ with $k \times n$ parity check matrix H is an $(n, n - k, 3)$-code.*

Proof: By defining n as we have above, we have included in the columns of H a representative from each one-dimensional subspace of $V(n, q)$, by Lemma 2.4. So by picking the columns of H to be n pairwise linearly independent vectors, we have chosen a maximal set of pairwise linearly independent vectors. In particular, if we take the sum of any two vectors, say $\mathbf{v}$ and $\mathbf{w}$, which are columns of H, their sum must be in the span of one of the columns of H. Otherwise the maximality of the set of column vectors of H would be violated. Thus we can find three column vectors in H which are linearly dependent, and so the minimum distance of this code must be 3. Among the columns of H we must include a representative from

the one-dimensional space spanned by each of the usual basis vectors $\langle 1, 0, \ldots, 0\rangle$, $\langle 0, 1, \ldots, 0\rangle$, $\langle 0, 0, \ldots, 1\rangle$, and so the rows of H will be linearly independent. Thus the rowspace of H has dimension k, and so $\mathrm{Ham}_q(n)$ will have dimension $n - k$. **Done.**

We also have the following corollary, analogous to Corollary 2.3.3, for q-ary Hamming codes.

2.4.4 Corollary. *The q-ary Hamming code $Ham_q(n)$ with $k \times n$ parity check matrix H will be a perfect 1-code with transmission rate*

$$\rho(Ham_q(n)) = \frac{q^k - k - 1}{q^k - 1}. \tag{2.8}$$

Proof: That $\mathrm{Ham}_q(n)$ is a perfect 1-code is obvious from Corollary 2.1.4, and the rate of $\mathrm{Ham}_q(n)$ is immediate from the definition of $\mathrm{Ham}_q(n)$. **Done.**

Note that the transmission rate of a q-ary Hamming code approaches 1 as $k \to \infty$. Thus for large message blocks, the Hamming codes are economical in their encoding of the source message. Note also that as $q \to \infty$, the rate of the code increases. So using larger fields results in a larger transmission rate. The minimum distance, however, remains 3 for any k and q.

2.4.5 Example. *Ternary Hamming code of length 4.* For a ternary code, $q = 3$, and we pick $k = 2$ so that $n = \frac{3^2-1}{3-1} = 4$. We may take the following as a parity

check matrix

$$H = \begin{pmatrix} 2 & 1 & 1 & 0 \\ 2 & 2 & 0 & 1 \end{pmatrix} \tag{2.9}$$

and so we see that

$$\begin{aligned} \mathrm{Ham}_3(4) =\{&(0,0,0,0),(0,1,1,2),(0,2,2,1),(1,0,1,1),\\ &(1,1,2,0),(1,2,0,2),(2,0,2,2),(2,1,0,1),(2,2,1,0)\} \end{aligned}$$

is a (4,2,3)-code in $V(4,3)$. Since the minimum distance of $\mathrm{Ham}_3(4)$ is 3, this code can correct single errors. The transmission rate of $\mathrm{Ham}_3(4)$ is $\frac{2^2-2-1}{2^2-1} = \frac{5}{7}$.

To explore the relative strengths and weaknesses of the Hamming codes, we must place them in a broader context. In next chapter we introduce the theory of association schemes and how they relate to the Hamming codes. For more information on the Hamming codes, see [5] or [7].

Chapter 3

Association Schemes

3.1 Introduction to Association Schemes

While we can say much about codes without talking about association schemes, the algebraic structure they provide gives us a more complete picture of how codes interact with the set they are in. Delsarte used this structure to give what is one of the strongest bounds on code size to date. What is remarkable about this bound is that is applies not only to linear codes, but to any code, and indeed to any subset of an association scheme.

In its most general sense, an association scheme is defined as follows:

3.1.1 Definition. An **association scheme** $\mathcal{A} = \{X, \mathcal{R}\}$ is finite set X and a set

of relations $\mathcal{R} = \{R_0, R_1, \ldots, R_d\}$ on X such that the R_i satisfy

1. $R_0 = \{(x, x) : x \in X\}$
2. For each $R_i \in \mathcal{R}$, the relation $R_i^{-1} = \{(y, x) : (x, y) \in R_i\}$ is also in $\mathcal{R}$.
3. $\mathcal{R}$ partitions $X \times X$.
4. For any $h, i, j \in [0, d]$ there exists a nonnegative integer $p_{i,j}^h$ such that for all (x, y) in R_h

$$|\{z \in X : (x, z) \in R_i \text{ and } (z, y) \in R_j\}| = p_{i,j}^h. \tag{3.1}$$

5. $p_{i,j}^h = p_{j,i}^h$ for all $h, i, j \in [0, d]$.

The numbers $p_{i,j}^h$ are sometimes called the *parameters* of the scheme and d is called the *diameter* of the scheme. If condition 2 is strengthened to require that $R_i = R_i^{-1}$ for all i, then the association scheme is called *symmetric*. In this case, property 5 can be omitted, as it follows from the other conditions.

The particular scheme that we are interested in is the Hamming scheme, which is defined as follows:

3.1.2 Definition. Let K be a finite Abelian group with q elements and X be the set of all n-tuples with entries from K. Define a set $\mathcal{R}$ of relations R_i on X by $(\mathbf{x}, \mathbf{y}) \in R_i$ whenever $d(\mathbf{x}, \mathbf{y}) = i$, where d denotes the Hamming metric. Then

$\mathcal{A} = (X, \mathcal{R})$ is a symmetric association scheme referred to as the *Hamming scheme*. The Hamming scheme is denoted $H(n, q)$.

To see that $H(n, q)$ is indeed a symmetric association scheme, observe that the Hamming metric is symmetric, and so the Hamming scheme will be symmetric. For any v in $V(n, q)$, $d(v, v) = 0$ and so $R_0 = \{(v, v)\}$ as required. The relations R_i clearly partition $V(n, q) \times V(n, q)$. Determining the $p^h_{i,j}$'s is not difficult, but it can be time consuming. It is made easier by the fact that if one of h, i, or j is larger than the sum of the other two, then $p^h_{i,j} = 0$. This is a direct result of the triangle inequality. The following combinatorial expression for the $p^h_{i,j}$ of the Hamming scheme is $H(n, q)$ derived in [1].

$$p^h_{i,j} = \sum_{\delta=0}^{\lfloor i+j-h/2 \rfloor} (q-2)^{i+j-h-2\delta} \binom{h}{j-\delta} \binom{j-\delta}{h-i+\delta} \binom{n-h}{(i+j-w)/2}. \tag{3.2}$$

We illustrate the Hamming schemes with an example.

3.1.3 Example. *The Hamming scheme H(3,2)*

Consider the set X of 3-tuples with entries in $F_2 = \{0, 1\}$. The scheme $H(3, 2)$ can be represented as a graph with vertex set X and an edge between two vertices if and only if they differ by exactly one entry (see figure 3.1). The distance between vertices, i.e.- the length of the shortest edge path connecting them, will then indicate which relation they are contained in.

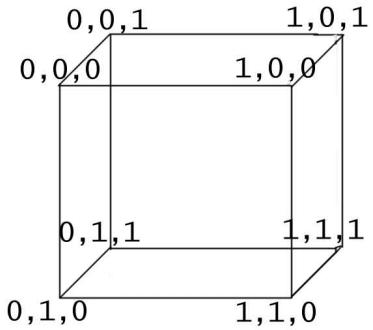

Figure 3.1: The 3-dimensional binary Hamming cube.

The Hamming code $\text{Ham}_2(3)$ is the code $C_2 = \{000, 111\}$ given in the introduction. Notice that the codewords 000 and 111 are at distance 3 from each other in the above graph, and that every other vertex in the graph is distance 1 from exactly one of 000 or 111. So $\text{Ham}_2(3)$ is a perfect 1-code. It follows that $\text{Ham}_2(3)$ has the property that it can correct exactly 1 transmission error. For more information on graph theory and it's applications to coding theory, see [4]

3.2 Associate Matrices and the Bose Mesner Algebra

Assume $\mathcal{A} = \{X, \mathcal{R}\}$ is a symmetric association scheme. There is a useful representation for each R_i as a symmetric zero-one matrix A_i in $Mat_X(\mathbb{C})$. For each i,

$0 \leq i \leq d$, the matrix A_i is defined as

$$(A_i)_{u,v} = \begin{cases} 1 & (u,v) \in R_i \\ 0 & (u,v) \notin R_i. \end{cases} \tag{3.3}$$

The A_i's are known as the *associate matrices*, and they have the following well-known properties, which follow directly from the axioms of a symmetric association scheme above

1. $A_0 = I$
2. $\sum_{i=0}^{d} A_i = J$
3. $A_j A_i = \sum_{h=0}^{d} p_{i,j}^h A_h \ \forall \ 0 \leq i, j \leq d$

Since the associate matrices are zero-one matrices, Property 2 above implies that the matrices A_i are linearly independent. Property 3 above shows that the product of matrices in the span of the A_i's is again in the span of the A_i's, and so the set $\{A_0, A_1, \ldots, A_d\}$ forms a basis for an algebra $\mathcal{M} \subseteq Mat_X(\mathbb{C})$, known as the *Bose-Mesner Algebra*. Since the A_i's are all symmetric, $\mathcal{M}$ is commutative.

For Example 3.1.3, the associate matrices are as follows:

$$A_0 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad A_1 = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad A_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The diameter of the corresponding graph, Figure (3.1) above, is 3, and so we should have 4 relations, and hence 4 associate matrices. Their sum should be the all-ones matrix, and so a 1 should occur in the (i, j) position in exactly one matrix. Both these properties are satisfied by inspection of the above matrices.

3.3 Orthogonal Idempotents and Properties

The Bose-Mesner algebra of a symmetric association scheme has, as a second basis, a set of mutually orthogonal primitive idempotent matrices. To see this, note that the matrices A_i have real entries and are symmetric (because our scheme is symmetric) and so satisfy $A_i = \overline{A_i^t}$. Therefore by the spectral theorem of linear algebra there exist symmetric matrices $E_0, E_1, \ldots, E_d$ in $\mathcal{M}$ satisfying the following properties:

1. $E_i E_j = \begin{cases} \mathbf{0} & i \neq j \\ E_i & i = j \end{cases} \qquad \forall \qquad 0 \leq i, j \leq d$,
2. $A = \sum_{i=0}^{d} \lambda_i E_i$,
3. $\sum_{i=0}^{d} E_i = I$,
4. $AE_i = \lambda_i E_i \qquad \forall \qquad 0 \leq i, j \leq d$,

where the $\{\lambda_i\}_{i=0}^{d}$ are the $d+1$ distinct eigenvalues of A. Each E_i is constructible as follows: For each A_i, let $u_0^{(i)}, u_1^{(i)}, \ldots, u_k^{(i)}$ be the k eigenvectors of A_i. Then

$$E_i = \left(u_0^{(i)} | u_1^{(i)} | \ldots | u_k^{(i)}\right) \left(u_0^{(i)} | u_1^{(i)} | \ldots | u_k^{(i)}\right)^t \tag{3.4}$$

3.4 The Eigenmatrices P and Q

Let $\mathcal{A}$ be an association scheme with associate matrices $A_0, A_1, \ldots, A_d$ and orthogonal idempotents $E_0, E_1, \ldots, E_d$. The *first eigenmatrix* P is defined to be the matrix such that

$$(A_0, A_1, \ldots, A_d) = (E_0, E_1, \ldots, E_d)P. \tag{3.5}$$

From this definition, it is clear that the entries in the j^{th} column of P are the eigenvalues of A_j. The *second eigenmatrix* Q is defined to be the matrix such that

$$(E_0, E_1, \ldots, E_d) = |X|^{-1}(A_0, A_1, \ldots, A_d)Q. \tag{3.6}$$

From these definitions, and the relations between the A_i's and E_i's, we have that

$$PQ = |X|I \tag{3.7}$$

The eigenmatrices of the Hamming scheme $H(3, 2)$ from example (3.1.3) are

$$P = \begin{pmatrix} 1 & 3 & 3 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -3 & 3 & -1 \end{pmatrix} \tag{3.8}$$

and

$$Q = \begin{pmatrix} 1 & 3 & 3 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -3 & 3 & -1 \end{pmatrix} \tag{3.9}$$

The matrix Q is central to the formulation of the linear programming bound we wish to establish, and so explicit formulas for the matrices P and Q for an arbitrary Hamming scheme $H(n, q)$ will be derived in the next section.

3.5 The Eigenmatrices of the Hamming Scheme

The calculation of the P and Q matrices for a particular scheme is non-trivial. Delsarte [3] showed that the entries of the eigenmatrix Q for the Hamming scheme $H(n, q)$ can be found using the Krawtchouk polynomials

$$K_k(x) = \sum_{i=1}^{k} \binom{x}{i} \binom{n-x}{k-i} (-1)^i (q-1)^{k-i}. \tag{3.10}$$

This is particularly helpful for the Hamming schemes, which are self dual, and so satisfy $Q = P$ [10]. As a preliminary to the development of the matrix Q for the Hamming schemes, we must first discuss inner products in Abelian groups.

Let F denote a finite Abelian group. It is well known (see [6], for example) that F is isomorphic to a multiplicative group of complex characters, homomorphisms from F to $\mathbb{C}$. For each $a \in F$, let φ_a denote the corresponding character. Thus we can define the inner product $\langle\ ,\ \rangle_* : F \times F \to \mathbb{C}$ by

$$\langle a, b \rangle_* = \varphi_a(b) \tag{3.11}$$

By its definition, $\langle\ ,\ \rangle_*$ has the following properties (see [6]):

1. $\langle a, b\rangle_* = \langle b, a\rangle_*$ for all a, b in F

2. $\langle a, b\rangle_*\langle a, c\rangle_* = \langle a, b + c\rangle_*$ for all a, b, c in F

3. If $\langle a, b\rangle_* = \langle a, c\rangle_*$ for all a in F, then $b = c$

Delsarte showed that this inner product has some very useful properties. A preliminary lemma makes these clear.

3.5.1 Lemma. *Let F be an Abelian group and let G be a subgroup of F. Define the dual subgroup G^0 of G by*

$$G^0 = \{a \in F : \langle a, b\rangle_* = 1 \ \textit{for all} \ b \in G\}. \tag{3.12}$$

Then the inner product $\langle\ ,\ \rangle_*$ defined above satisfies

$$\sum_{a\in G}\langle b, a\rangle_* = \begin{cases} |G| & b \in G^0 \\ 0 & \textit{else} \end{cases} \tag{3.13}$$

Proof: Suppose $b \in G^0$. Then each of the terms in the sum on the left side of (3.13) is 1 by the definition of G^0, and since the sum is taken over all elements of G, we get $|G|$. If $b \notin G^0$, then let $c \in G$. By the properties of the inner product

$\langle\ ,\ \rangle_*$, we have that

$$\langle b,c\rangle_* \sum_{a\in G}\langle b,a\rangle_* = \sum_{a\in G}\langle b,c\rangle_*\langle b,a\rangle_* \tag{3.14}$$

$$= \sum_{a\in G}\langle b,a+c\rangle_* \tag{3.15}$$

$$= \sum_{d\in G}\langle b,d\rangle_*. \tag{3.16}$$

Suppose the sum in (3.13) was not zero. Then we would have that $\langle b,c\rangle_* = 1 = \langle 0,c\rangle_*$ for all $c \in G$, and so $b = 0$ by the properties of $\langle,\rangle_*$. Thus $b \in G^0$, which is the case already considered. Thus the lemma holds. **Done.**

From this lemma we have the following corollary.

3.5.2 Corollary. *If F is a field of order q, then*

$$\sum_{b\in F*}\langle a,b\rangle_* = \begin{cases} q-1 & \text{for } a=0, \\ -1 & \text{for } a\neq 0, \end{cases} \tag{3.17}$$

where F^* is the set of non-zero elements of F.

Proof: Referring to Lemma 3.5.1, we take G to be F, and so $G^0 = \{0\}$. Then since $\langle a,0\rangle = 1$ for all $a \in F$, the result follows immediately from Lemma 3.5.1. **Done.**

Now let X be the set of n-tuples with entries from the finite Abelian group F. The inner product on F can be extended to an inner product $\langle \mathbf{x},\mathbf{y}\rangle$ on $X = F^n$

which inherits the properties 1-3 above by taking

$$\langle \mathbf{x}, \mathbf{y} \rangle = \prod_{i=1}^{n} \langle x_i, y_i \rangle_*. \tag{3.18}$$

This inner product on X can be used to define an orthogonal matrix as follows.

3.5.3 Corollary. *Let $S \in Mat_X(\mathbb{C})$ be defined as*

$$S_{\mathbf{x},\mathbf{x}'} = \langle \mathbf{x}, \mathbf{x}' \rangle. \tag{3.19}$$

Then S is an orthogonal matrix. That is, it satisfies $S\widetilde{S} = |X|I$.

Proof: As in the previous corollary, we take $G = F$ and so $G^0 = \{0\}$. Then the result follows immediately from matrix multiplication and the properties of the inner product on X. **Done.**

Given a finite dimensional vector space X over a field F, we define the *weight partition* of X as follows.

3.5.4 Definition. If X is a finite dimensional vector space over a field F we define the i^{th} weight partition X_i of X to be the set of all vectors in X with precisely i non-zero entries. In the language of coding theory, this amounts to a word having weight i. The collection of all such X_i's is the *weight partition* of X.

3.5.5 Lemma. *Let X be the set of n-tuples with entries from the finite Abelian group F with $|F| = q$ where q is a prime power, and let F^* denote the set of non-zero elements in F. The Krawtchouk polynomials, given in (3.10), and the*

inner product defined above are related in the following fashion.

$$\sum_{\mathbf{x}' \in X_k} \langle \mathbf{x}, \mathbf{x}' \rangle = K_k(u) \qquad \forall \quad \mathbf{x} \in X_u \tag{3.20}$$

Proof: Let J be a subset of $\{1, 2, \ldots, n\}$ with $|J| = k$. There are $(q-1)^k$ words $\mathbf{x}'$ in X with $x'_i \neq 0$ for all i in J, and the product $\langle \mathbf{x}, \mathbf{x}' \rangle$ for each of those $\mathbf{x}'$ can be expressed, using equation (3.18) as

$$\prod_{i \in J} \langle x_i, x'_i \rangle_*, \tag{3.21}$$

since when $i \notin J$, $x'_i = 0$ and so the factor $\langle x_i, x'_i \rangle$ in the product in (3.18) would be 1. This can be expressed as

$$\prod_{i \in J} \left(\sum_{b \in F^*} \langle x_i, b \rangle_* \right) \tag{3.22}$$

since, in multiplying all sums together, we are combining the entries in $\mathbf{x}$, corresponding to non-zero entries in $\mathbf{x}'$, with all non-zero elements in F.

Using equation (3.17), each of the sums in (3.22) is equal to -1 or $q-1$, depending on whether x_i is zero or not. Thus if we denote by j the number of non-zero entries x_j in $\mathbf{x}$ where x'_j is also not zero, the product in (3.22) becomes

$$(-1)^j (q-1)^{k-j}. \tag{3.23}$$

We will get such a term for each choice of J, and there are $\binom{u}{j}\binom{n-u}{k-j}$ such terms. This is because, of the u non-zero elements in $\mathbf{x}$, we choose j of these for $\mathbf{x}$ and

$\mathbf{x}'$ to have in common, and then choose $k-j$ more entries from the $n-u$ entries remaining in $\mathbf{x}'$ to give $\mathbf{x}'$ a weight of k. **Done.**

3.5.6 Theorem. *For the Hamming scheme $H(n,q)$,*

$$Q_{i,k} = K_k(i) \tag{3.24}$$

for $i,k = 0,1,\ldots,d$, where $K_k(x)$ is the Krawtchouk polynomial of degree k.

Proof: Let X be a finite dimensional vector space over the finite field F and let $\mathcal{W} = \{X_0, X_1, \ldots, X_n\}$ be the weight partition of X. That is, X_k is the set of all words in X with weight k. Let F^* denote the set of non-zero elements of F.

Define a matrix $S \in \mathbb{C}(X,X)$ by $S_{\mathbf{x},\mathbf{x}'} := \langle \mathbf{x}, \mathbf{x}' \rangle = \prod_{i=1}^{n} \langle x_i, x_i' \rangle$. For each k, let S_k be the $n \times |X_k|$ submatrix of S consisting of the columns of S corresponding to the vectors in X_k. Then the $\mathbf{x},\mathbf{y}$-entry of $S_k\widetilde{S_k}$, where $\widetilde{A}$ denotes the conjugate transpose of A, is given by

$$\begin{aligned}
\left(S_k\widetilde{S_k}\right)_{\mathbf{x},\mathbf{y}} &= \sum_{\mathbf{x}' \in X_k} (S_k)_{\mathbf{x},\mathbf{x}'} (\widetilde{S_k})_{\mathbf{x}',\mathbf{y}} \\
&= \sum_{\mathbf{x}' \in X_k} \langle \mathbf{x}, \mathbf{x}' \rangle \langle -\mathbf{y}, \mathbf{x}' \rangle \\
&= \sum_{\mathbf{x}' \in X_k} \langle \mathbf{x} - \mathbf{y}, \mathbf{x}' \rangle \\
&= K_k(\omega(\mathbf{x}-\mathbf{y})).
\end{aligned}$$

By the definition of the Hamming scheme, two vectors $\mathbf{x}$ and $\mathbf{y}$ are in the i^{th} relation if and only if they differ by i entries, and this number is given by $\omega(\mathbf{x}-\mathbf{y})$.

The associate matrices represent these relations, and so $\omega(\mathbf{x}-\mathbf{y}) = i$ if and only if $(A_i)_{\mathbf{x},\mathbf{y}} = 1$. So the product $S_k\widetilde{S_k}$ is given by

$$S_k\widetilde{S_k} = \sum_{i=0}^{d} K_k(i)A_i. \tag{3.25}$$

Since the matrix S is an orthogonal matrix, the matrices given by $|X|^{-1}S_k\widetilde{S_k}$ are mutually orthogonal idempotents, which by equation (3.25) are in the Bose-Mesner algebra of X. Thus they are equal to the E_i's defined above, and so we have that

$$E_i = |X|^{-1}\sum_{i=0}^{d} K_k(i)A_i, \tag{3.26}$$

which is precisely how the entries in the second eigenmatrix Q were defined. Thus the i,k-entry in Q is given by $K_k(i)$. **Done.**

Chapter 4

The Linear Programming Bound

4.1 Distribution vectors

If $(X, \mathcal{R})$ is an association scheme and Y is a subset of X, then we can define the distribution vector and distribution matrix of Y as follows.

The *distribution vector* of Y is the vector $\mathbf{a}$ whose i^{th} entry is given by

$$a_i = \frac{|(Y \times Y) \cap R_i|}{|Y|}. \tag{4.1}$$

The a_i's can be thought of as the average number of elements in Y at distance i from some other element of Y. Note also that

$$\sum_{i=0}^{d} a_i = |Y| \tag{4.2}$$

As an example, recall the Hamming scheme $H(3, 2)$. If we take the set $\mathrm{Ham}_2(3) =$

$\{000, 111\}$ as our subset Y, then the distribution vector for Y is $\langle 1, 0, 0, 1\rangle$.

4.2 The Linear Programming Bound

The linear programming bound, presented by Phillipe Delsarte in 1973, improved significantly upon the Hamming bound given in Theorem 2.1.3. It is essentially an algebraic bound, owing little to combinatorial arguments, and this is perhaps why it succeeds so well.

4.2.1 Theorem. *If* $\mathbf{a}$ *is the distribution vector of a subset* Y *of an association scheme* $(X, \mathcal{R})$ *with dual eigenmatrix* Q*, then* $\mathbf{a}Q \geq \mathbf{0}$*.*

Proof: Let $\mathbf{y}$ be the characteristic vector of Y. Then the i^{th} entry of $\mathbf{a}$ is given by

$$a_i = \frac{\mathbf{y}A_i\mathbf{y}^t}{|Y|}. \tag{4.3}$$

Then, since the E_i's are idempotent and symmetric,

$$\begin{aligned} 0 \leq & ||\mathbf{y}E_k||^2 \\ = & (\mathbf{y}E_k)(\mathbf{y}E_k)^t \\ = & \mathbf{y}E_k\mathbf{y}^t. \end{aligned}$$

Using equations (3.6) and (4.3), this can be expressed as

$$\begin{aligned} 0 \leq & \frac{1}{|X|}\mathbf{y}\left(\sum_{j=0}^{d} Q_{j,k}A_j\right)\mathbf{y}^t \\ = & \frac{1}{|X|}\left(\sum_{j=0}^{d} Q_{j,k}\mathbf{y}A_j\mathbf{y}^t\right) \\ = & \frac{|Y|}{|X|}\left(\sum_{j=0}^{d} a_j Q_{j,k}\right). \\ = & \frac{|Y|}{|X|}(\mathbf{a}Q)_k \end{aligned}$$

Thus we have the desired inequality. **Done.**

The utility of Theorem 4.2.1 may not be immediately apparent, and so we give the following corollary.

4.2.2 Corollary. *[9] Let $\mathcal{A}$ be an association scheme with dual eigenmatrix Q, diameter d, and distribution vector $\mathbf{a} = \langle a_0, a_1, \ldots, a_d \rangle$. Then any code C with minimum distance r in $\mathcal{A}$ satisfies*

$$|C| \leq \max\left(\sum_{i=0}^{d} a_i\right) \tag{4.4}$$

where the maximum is taken over all $\{a_0, \ldots, a_d\}$ where the a_i's satisfy

1. $a_0 = 1$,
2. $a_i = 0$ *for* $1 \leq i \leq r$,
3. $a_i \geq 0 \qquad \forall \quad i$, *and*

4. $\mathbf{a}Q \geq \boldsymbol{0}$.

Proof: Immediate from Theorem 4.2.1. **Done.**

When constructing a code C with a particular minimum distance, Theorem 4.2.1 gives a restriction on the distribution vector of C. The first element of $\mathbf{a}$ will be 1, and if we want to restrict our bound to codes of minimum distance r, the entries a_1 through a_{r-1} should all be zero and the remaining entries should be non-negative. So we wish to maximize the sum

$$1 + a_r + a_{r+1} + \ldots + a_d \tag{4.5}$$

subject to the restrictions

$$a_i \geq 0 \qquad \forall \quad i \tag{4.6}$$

and

$$\mathbf{a}Q \geq \mathbf{0}. \tag{4.7}$$

The linear programming bound gives us this last relationship, and so the solution to this linear program will be an upper bound on the size of our code. Note that Theorem 4.2.1 did not use any properties of codes, and so this bound applies to all subsets of an association scheme. In [3] Delsarte used the linear programming bound to establish the Hamming bound, and so the linear programming bound is always at least as strong as the Hamming bound. This has some important ramifications, as we see in the next section.

4.3 Bound Comparison

The following table is a modification of one which appears in [10], and compares the bounds produced by the sphere packing bound (Theorem 2.1.3) and the linear programming bound (Theorem 4.2.1) for a binary $[n, d]$-code in the Hamming scheme:

n	δ	Hamming Bound	Linear Programming Bound
11	3	170.7	170.7
11	5	30.6	24
11	7	8.8	4
12	3	315.1	292.6
12	5	51.9	40
12	7	13.7	5.3
13	3	585.1	512
13	5	89.0	64
13	7	21.7	8
14	3	1092.3	1024
14	5	154.6	128
14	7	34.9	16
15	3	2048	2048
15	5	270.8	256
15	7	56.9	32

Table 4.1: Hamming and Linear Programming bounds for $q = 2$

There is a Hamming code with word length 15 which has 2048 codewords, and so that bound is realizable. Note that for larger minimum distances, the Linear Programming bound is approximately twice as strong as the Hamming bound.

The same parameters were used to construct the following table for a ternary $[n, d]$-code in the Hamming scheme.

n	δ	Hamming Bound	Linear Programming Bound
11	3	7702.0	7029.6
11	5	729	729
11	7	113.3	63
12	3	21257.6	19683
12	5	1838.9	1562.14
12	7	259.4	138.6
13	3	59049	59049
13	5	4703.0	4217.8
13	7	606.9	363.3
14	3	164930.0	153527
14	5	12170.4	10736.2
14	7	1447.2	836.0
15	3	462868.0	434815
15	5	31815.8	29524.5
15	7	3507.4	2268.6

Table 4.2: Hamming and Linear Programming bounds for $q = 3$.

The same sort of relationship between the Hamming bound and Linear Programming bound seen in Table 4.1 is seen in this table. The Linear Programming bound is always stronger than the Hamming bound, but the difference is greater for larger minimum distances. There is a Hamming [13,3]-code with 59049 words, so that bound is realized. The interesting entry is the (11,5) entry where the two bounds agree and produce a whole number bound. This is strong evidence that such a code exists. The above tables also tell us the bounds on the size of codes of larger minimum distance than the Hamming codes in the Hamming scheme.

Perfect codes are remarkable structures, and so one would expect them to be relatively rare. In Corollary 2.1.4 we showed that a code achieves the Hamming bound if and only if it is perfect. Since the linear programming bound is generally lower than the Hamming bound, this implies that perfect codes are not abundant. The bound given by the (11,5) entry in Table 4.2 is realized by a particular perfect code known as the ternary Golay code.

The Hamming codes can be used to construct codes with larger minimum distances, such as the famous Golay codes (See [8]). Thus bounds on code sizes in the proximity of Hamming codes are useful. The linear programming bound also has applications outside of coding theory, as in the work of Henry Cohn and Noam Elkies in sphere packing and harmonic analysis [2].

4.4 Conclusion

We have developed two of the main goals in coding theory: Classifying codes and determining when codes of a certain type exist. The Hamming codes, while interesting for their pedagogical purposes, are also practically useful. The codes with shorter word length are economical in their encoding and easy to decode. The codes with longer word length can be used to construct new codes, often increasing their minimum distance and thus their ability to detect and correct transmission errors. The linear programming bound aids in this search for new codes by limiting the number of possible codes of a particular kind a set can contain.

Bibliography

[1] A. Barg, S. Guritman, and J. Simonis. Strengthening the Gilbert-Varshamov bound. *Linear Algebra and its Applications*, 307:119–129, 2000.

[2] H. Cohn and N. Elkies. New upper bounds on sphere packings I. *arXiv*, 2, 2002.

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

[4] Chris Godsil and Gordon Royle. *Algebraic Graph Theory*. Springer-Verlag, New York, 2001.

[5] R. W. Hamming. Error detecting and error correcting codes. *Bell Sys. Tech. J.*, 29, 1950.

[6] W. Ledermann. *Introduction to Group Characters, Second Edition*. Cambridge University Press, Cambridge, 1987.

[7] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error Correcting Codes, Vols. 1 and 2*. North-Holland, Amsterdam, 1977.

[8] O. Pretzel. *Error Correcting Codes and Finite Fields*. Oxford University Press, New York, 1992.

[9] S. Roman. *Coding and Information Theory*. Springer-Verlag, New York, 1992.

[10] J. H. van Lindt and R. M. Wilson. *A Course in Combinatorics*. Cambridge University Press, Cambridge, 1992.