Error Correcting Codes: Combinatorics, Algorithms and Applications **(Fall 2007)**

Lecture 5: Linear Codes

September 7, 2007

Lecturer: Atri Rudra *Scribe: Michel Kulhandjian & Atri Rudra*

Last lecture we talked about the Hamming bound (for the special case of $d = 3$) and informally defined binary linear codes. In today's lecture, we will present the general form of Hamming bound and define (general) linear codes. The latter will involve looking at finite fields and vector spaces over finite fields.

1 General Family of Hamming Codes

We start with a new notation.

Definition 1.1. *A code $C \subseteq \Sigma^n$ with dimension k and distance d will be called a $(n, k, d)_\Sigma$ code. We will also refer it to as a $(n, k, d)_{|\Sigma|}$ code.*

We now state the Hamming bound for $d = 3$ that we proved in the last lecture.

Theorem 1.2 (Hamming Bound for $d = 3$)**.** *For every $(n, k, 3)_2$ code*

$$k \leq n - \log_2(n + 1).$$

We now proceed to generalize the above theorem to any distance d.

Theorem 1.3 (Hamming Bound for any d)**.** *For every $(n, k, d)_q$ code*

$$k \leq n - \log_q \left(\sum_{i=0}^{\lfloor \frac{(d-1)}{2} \rfloor} \binom{n}{i} (q-1)^i \right).$$

Proof. The proof is a straightforward generalization of the proof of Theorem 1.2 that we did last lecture. For notational convenience, let $e = \left\lfloor \frac{(d-1)}{2} \right\rfloor$. Given any two codewords, $c_1 \neq c_2 \in C$, the following is true (as C has distance[1] d):

$$B(c_1, e) \cap B(c_2, e) = \emptyset, \tag{1}$$

where recall that for any vector $\mathbf{x} \in [q]^n$,

$$B(\mathbf{x}, e) = \{\mathbf{y} \in [q]^n | \Delta(\mathbf{x}, \mathbf{y}) \leq e\}.$$

[1] Assume that $\mathbf{y} \in B(c_1, e) \cap B(c_2, e)$, that is $\Delta(\mathbf{y}, c_1) \leq e$ and $\Delta(\mathbf{y}, c_2) \leq e$. Thus, by the triangle inequality, $\Delta(c_1, c_2) \leq 2e \leq d - 1$, which is a contradiction.

We claim that for all $\mathbf{x} \in [q]^n$,

$$|B(\mathbf{x}, e)| = \sum_{i=0}^{e} \binom{n}{i} (q-1)^i. \tag{2}$$

Indeed any vector in $B(\mathbf{x}, e)$ must differ from $\mathbf{x}$ in exactly $0 \leq i \leq e$ positions. In the summation $\binom{n}{i}$ is the number of ways of choosing the differing i positions and in each such position, a vector can differ from $\mathbf{x}$ in $q-1$ ways.

Now consider the union of all Hamming balls centered around some codeword. Obviously their union is a subset of $[q]^n$. In other words,

$$\left| \bigcup_{c \in C} B(c, e) \right| \leq q^n. \tag{3}$$

As (1) holds for every pair of distinct codewords,

$$\begin{aligned} \left| \bigcup_{c \in C} B(c, e) \right| &= \sum_{c \in C} |B(c, e)| \\ &= q^k \sum_{i=0}^{e} \binom{n}{i} (q-1)^i, \end{aligned} \tag{4}$$

where (4) follows from (2) and the fact that C has dimension k. Combining (4) and (3) and taking $\log_q$ of both sides we will get the desired bound:

$$k \leq n - \log_q \left(\sum_{i=0}^{e} \binom{n}{i} (q-1)^i \right).$$

□

Remark 1.4. *Codes that meet Hamming bound are called* perfect codes*. One example of perfect code is the $[7, 4, 3]_2$ Hamming code (so is the family of general Hamming codes that we will see in the next lecture). A natural question to ask is if there any other perfect codes? We will see the answer in a couple of lectures.*

2 Linear Codes

Let us now pause for a bit and think about how well we can represent a code. In general, a code $C : \{0,1\}^k \longrightarrow \{0,1\}^n$ can be stored using $n2^k$ bits. For constant rate codes, this is exponential space, which is prohibitive even for modest values of k like $k = 100$. A natural question is whether we can do better. Intuitively, the code must have some extra structure that would facilitate a succinct representation of the code. We will now look at a class of codes called *linear codes* that have more structure than general codes which leads to some other nice properties. We have already see binary linear codes, that is, $C \subseteq \{0,1\}^n$ is linear code if for all $c_1, c_2 \in C$, $c_1 + c_2 \in C$, where the "+" denotes bit-wise xor.

Definition 2.1 (Linear Codes)**.** *Let q be a prime power. $C \subseteq \{0,1,...,q-1\}^n$ is a* linear code *if it is a* linear subspace *of $\{0,1,...,q-1\}^n$. If C has dimension k and distance d then it will be referred to as an $[n,k,d]_q$ or just an $[n,k]_q$ code.*

Of course the above definition is not complete because we have not defined a linear subspace yet. We do that next.

2.1 Finite Fields

To define linear subspaces, we will need to work with (finite) fields. We begin with a quick overview of fields. For a more through treatment refer to any standard text on algebra or the book on finite fields by Lidl and Niederreiter [1].

Informally speaking, a field is a set of elements on which one can do addition, subtraction, multiplication and division and still stay in the set. More formally,

Definition 2.2. *A field $\mathbb{F}$ is given by a triple $(S,+,\cdot)$, where S is the set of elements containing special elements 0 and 1 and $+,\cdot$ are functions $\mathbb{F}\times\mathbb{F}\to\mathbb{F}$ with the following properties:*

- Associativity: *$+$ and $\cdot$ are associative, that is, for every $a,b,c\in S$, $a+(b+c)=(a+b)+c$ and $a\cdot(b\cdot c)=(a\cdot b)\cdot c$.*
- Commutativity: *$+$ and $\cdot$ are commutative, that is, for every $a,b\in S$, $a+b=b+a$ and $a\cdot b=b\cdot a$.*
- Distributivity: *$\cdot$ distributes over $+$, that is for every $a,b,c\in S$, $a\cdot(b+c)=a\cdot b+a\cdot c$.*
- Identities: *For every $a\in S$, $a+0=a$ and $a\cdot 1=a$.*
- Inverses: *For every $a\in S$, there exists its* additive inverse *$-a$ such that $a+(-a)=0$. Also for every $a\in S\setminus\{0\}$, there exists its* multiplicative inverse *a^{-1} such that $a\cdot a^{-1}=1$.*

With the usual semantics for $+$ and $\cdot$, $\mathbb{R}$ (set of real number) is a field but $\mathbb{Z}$ (set of integers) is not as division of two integers can give rise to a rational number. In this course, we will exclusively deal with *finite fields*. As the name suggests these are fields with a fixed size set of elements. The following is a well known result.

Theorem 2.3 (Size of Finite Fields)**.** *The size of any finite field is p^s for prime p and integer $s\geq 1$.*

One example of finite fields that we have seen is the field of two elements $\{0,1\}$, which we will denote by $\mathbb{F}_2$ (we have seen this field in the context of binary linear codes). For $\mathbb{F}_2$, addition is the XOR operation, while multiplication is the AND operation. The additive inverse of an element in $\mathbb{F}_2$ is the number itself while the multiplicative inverse of 1 is 1 itself.

Let p be a prime number. Then the integers modulo p form a field, which in our notation is $\mathbb{F}_p$ and is also denoted by $\mathbb{Z}_p$. Note that the elements of $\mathbb{F}_p$ are $\{0,1,...,p-1\}$ and the addition, multiplication are carried out $\mod p$. One might think that there could be different fields with the same number of elements. However, this is not the case:

Theorem 2.4. *For every prime power q there is a* unique *finite field with q elements (up to isomorphism).*

Thus, we are justified in just using $\mathbb{F}_q$ to denote a finite field on q elements.

We are finally ready to define the notion of linear subspace.

Definition 2.5 (Linear Subspace)**.** *$S \subseteq {\mathbb{F}_q}^n$ is a linear subspace if the following properties hold:*

1. *For every $\mathbf{x}, \mathbf{y} \in S$, $\mathbf{x} + \mathbf{y} \in S$ where the addition is vector addition over $\mathbb{F}_q$ (that is, do addition component wise over $\mathbb{F}_q$).*

2. *For every $a \in \mathbb{F}_q$ and $\mathbf{x} \in S$, $a \cdot \mathbf{x} \in S$ where the multiplication is over $\mathbb{F}_q$.*

Here is a (trivial) example of a linear subspace of $\mathbb{F}_5^3$: $S = \{(0,0,0), (1,1,1), (2,2,2), (3,3,3), (4,4,4)\}$. Note that e.g. $(1,1,1) + (3,3,3) = (4,4,4) \in S$ and $2 \cdot (4,4,4) = (3,3,3) \in S$ as required by the definition.

Remark 2.6. *Note that the second property implies that $\mathbf{0}$ is contained in every linear subspace. Further for any subspace over $\mathbb{F}_2$, the second property is redundant (note that as $c + c = \mathbf{0}$ over $\mathbf{F}_2$, the first property guarantees that $\mathbf{0} \in S$).*

Any linear subspace satisfies the following properties which we state without proof (which can be found in any standard linear algebra textbook).

Theorem 2.7. *If $S \subseteq {\mathbb{F}_q}^n$ is a linear subspace then*

1. *$|S| = q^k$ for some $k \geq 0$. The parameter k is called the* dimension *of S.*

2. *There exists $\mathbf{v}_1, ..., \mathbf{v}_k \in S$ called* basis elements *(which need not be unique) such that every $\mathbf{x} \in S$ can be expressed as $\mathbf{x} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + ... + a_n\mathbf{v}_n$ where $a_i \in \mathbb{F}_q$ for $1 \leq i \leq k$. In other words, there exists a $k \times n$ matrix G (also known as* generator matrix*) with entries from $\mathbb{F}_q$ such that every $\mathbf{x} \in S$, $x = (a_1, a_2, ...a_k) \cdot G$ where*

$$G = \begin{pmatrix} \longleftarrow & \mathbf{v}_1 & \longrightarrow \\ \longleftarrow & \mathbf{v}_2 & \longrightarrow \\ & \vdots & \\ \longleftarrow & \mathbf{v}_k & \longrightarrow \end{pmatrix}$$

3. *There exists a $(n-k) \times n$ matrix H (called the* parity check matrix*) such that for every $\mathbf{x} \in S$, $H\mathbf{x}^T = \mathbf{0}$.*

Property 3 above follows from another fact that every linear subspace S has a null space $N \subseteq \mathbb{F}^n$ such that for every $\mathbf{x} \in S$ and $\mathbf{y} \in N$, $\langle \mathbf{x}, \mathbf{y} \rangle = 0$. Further, it is known that N itself is a linear subspace of dimension $n - k$. In other words, there exists a generator matrix H for it. This matrix H is called the parity check matrix of S. (We will see the utility of a parity check matrix soon.)

The above theorem gives two alternate characterizations of an $[n, k]_q$ linear code C:

- C is generated by its $k \times n$ generator matrix G. As an example that we have already seen, the $[7, 4, 3]_2$ Hamming code has the following generator matrix:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

- C is also characterized by an $(n-k) \times n$ parity check matrix H. We claim that the following matrix is the parity check matrix of the $[7, 4, 3]_2$ Hamming code (we will see a proof later on):

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

We now look at some consequences of the above characterizations of an $[n, k]_2$ linear code C.

1. We started this section with a quest for succinct representation of a code. Note that both the generator matrix and the parity check matrix can be represented using $O(n^2)$ symbols from $\mathbb{F}_q$ (which is much smaller than the exponential representation of a general code).

2. There is an encoding algorithm for C that runs in $O(n^2)$ (in particular $O(kn)$) time– given a message $\mathbf{m} \in \mathbb{F}_q^k$, the corresponding codeword $C(\mathbf{m}) = \mathbf{m} \cdot G$, where G is the generator matrix of C.

3. There is an error-detecting algorithm for C that runs in $O(n^2)$ (in particular $O(n^2 - nk)$) time– given a received word $\mathbf{y} \in \mathbb{F}_q^n$, check if $H \cdot \mathbf{y}^T = \mathbf{0}$. This is a big improvement over the naive brute force exponential time algorithm (that goes through all possible codewords $\mathbf{c} \in C$ and checks if $\mathbf{y} = \mathbf{c}$).

As a concluding remark we note that the generator matrix G and parity check matrix H by definition are orthogonal, that is, $G \cdot H^T = 0$. The proof of this fact is left as an exercise. (We will use this property later on in the course.)

References

[1] Rudolf Lidl and Harald Niederreiter. *Introduction to Finite Fields and their applications*. Cambridge University Press, Cambridge, MA, 1986.