Coding Theory
Lecture 2

Lecture by Dr. Elette Boyle
Typeset by Steven Karas

2017-11-02
Last edited 18:04:30 2017-11-02

Disclaimer These lecture notes are based on the lecture for the course Coding Theory, taught by Dr. Elette Boyle at IDC Herzliyah in the fall semester of 2017/2018. Sections may be based on the lecture slides written by Dr. Elette Boyle.

Admin No class next week. A survey will go out to schedule the makeup class.
A piazza forum may be opened.

Agenda

- Finite Fields
- Linear subspaces over finite fields
- Properties of linear codes
  - Generator Matrix G
  - Parity-Check Matrix H
- Hamming Codes

1 Review of definitions

1.1 Code

A code of block length n over an alphabet Σ is just a subset $C \subseteq \Sigma^n$. We mark $q = |\Sigma|$
Example: $\Sigma = \{0,1\}$ would be bits whereas $\Sigma = \{0,1\}^8$ would be bytes.
As a special case, if $\Sigma = \{0,1\}$, we call it a binary code.
An alternative view of a code is as a mapping $C : [M] \to \Sigma^n$ if $|C| = M$.

1.2 Dimension

The dimension of a code $C \subseteq \Sigma^n$ is $k = \log_q |C|$ where $q = |\Sigma|$.

1.3 Rate

The rate of a code with dimension k and block length n is $R = \frac{k}{n}$.
We use the rate as a measure of the quality of the code.

1.4 Error Correction/Detection

Given a message m, we will add some redundancy using $C(m)$ such that $D(C(m) + e)$ where e is the noise added by the communication channel hopefully maps back to m.

1.4.1 Encoding

Let $C \subseteq \Sigma^n$ be a code. An equivalent description of the code C is by an injective mapping $E : [|C|] \to \Sigma^n$ called the encoding function.

1.4.2 Decoding

Let $C \subseteq \Sigma^n$ be a code. A mapping $D : \Sigma^n \to [|C|]$ is called the decoding function for C.

1.4.3 Error Correction

Let $C \subseteq \Sigma^n$ and $t \geq 1 \in \mathbb{N}$. C is said to be t-error-correcting if there exists some decoding function D such that $\forall m \in [|C|]$ and for every error pattern $e \in \Sigma^n$ with at most t errors a decoding function $D(c(m) + e) = m$.

1.4.4 Error Detection

Let $C \subseteq \Sigma^n$ and $t \geq 1 \in \mathbb{N}$. C is said to be t-error-detecting if there exists some detection function D_{detect} such that $\forall m \in [|C|]$ and for every error pattern $e \in \Sigma^n$ with at most t errors a detection function $D_{detect}(c(m)) = \text{accept}$, and $D_{detect}(c(m) + e) = \text{reject}$.[1]

Note: t-error-correcting implies t-error-detecting

1.5 Channel noise/Errors

When talking about errors we typically think of Σ as having some additive structure with a 0 element. We express error to a code word $c \in \Sigma^n$ as a vector $e \in \Sigma^n$, where the received (noisy) word is $y = c + e$. We say that e has t errors if the number of non-zero members of e is t.

There are many ways to model channel noise.

Adversarial noise Assuming any worst case error patterns, as long as the number of symbols in the error is bounded.

Stochastic noise Assuming the expected case. Binary symmetric channel of probability $p : 0 \leq p \leq 1$ called BSC_p; each bit is flipped with probability p.

There are lots of different extensions of these. We will focus on adversarial noise.

1.5.1 Erasure

An error where the receiver knows that an error was in a given position.

1.6 Distance

First, we define a useful distance measure between vectors: the number of positions where elements differ.

For some $u, v \in \Sigma^n$, the Hamming distance between u and v, $\Delta(u, v)$ is defined to be the number of positions where u and v differ.

Minimum Distance Let $C \subseteq \Sigma^n$ be a code. The minimum distance of C is $d = \min_{c \neq c' \in C} \Delta(c, c')$

Equivalence of distance properties The following statements are equivalent for any code C:

1. C has distance $d \geq 2$.
2. C can correct $(d-1)/2$ errors if d is odd, or $d/2 - 1$ errors if d is even.
3. C can detect $(d-1)$ errors.
4. C can correct $(d-1)$ erasures.

[1] This definition is different from the one in the text book. The textbook definition is incorrect, as it ignores cases where the error mutates the message into another valid code word

2 Fields

A field is an abstraction of the real numbers that captures the applicable properties.

2.1 Formal definition

A field is a $\mathbb{F}$ is given by a 3-tuple $(S, +, \cdot)$:

- S is a set containing two special elements: 0 and 1.
- $+, \cdot$ are binary operators $\mathbb{F} \times \mathbb{F} \to \mathbb{F}$ such that the following properties hold:

2.1.1 Closure

$\forall a, b \in S$, $a + b \in S$, $a \cdot b \in S$.

2.1.2 Associativity

$\forall a, b, c \in S$, $(a + b) + c = a + (b + c)$, $(a \cdot b) \cdot c = a \cdot (b \cdot c)$

2.1.3 Commutativity

$\forall a, b, c \in S$, $a + b = b + a$, $a \cdot b = b \cdot a$

2.1.4 Distributivity

$\forall a, b, c \in S$, $a \cdot (b + c) = a \cdot b + a \cdot c$

2.1.5 Identities

$\forall a \in S$, $a + 0 = a$, and $a \cdot 1 = a$

2.1.6 Inverses

$\forall a \in S$, there exists a unique additive inverse $-a$ such that $a + -a = 0$
$\forall b \neq 0 \in S$, there exists a unique multiplicative inverse b^{-1} such that $b \cdot (b^{-1}) = 1$

2.2 Finite Fields

A **finite** field is a field $\mathbb{F} = (S, +, \cdot)$ for which $|S|$ is finite. We denote this as $|\mathbb{F}|$.

2.3 Modular Arithmetic

Modular fields are a useful class of
Let p be prime:

$$\mathbb{F}_p = (\{0, \ldots, p-1\}, +_p, \cdot_p)$$

Where $+_p, \cdot_p$ denote addition and multiplication modulo p.
For convenience, mark $S_p = \{0, 1, \ldots, p-1\}$

2.3.1 Proof

Associativity, Commutativity, Distributivity, and Identities we get because they hold for integer arithmetic.

Proof of closure Because you take the remainder mod p, which brings you back to S_p

Proof of inverses $\forall a \in S_p$, consider $b = p - a \mod p$ for some $b \in S_p$.

$$= a +_p b \mod p \tag{1}$$
$$= a +_p (p - a) \mod p \tag{2}$$
$$= p \tag{3}$$
$$= 0 \tag{4}$$

Let $a \neq 0 \in S_p$.

We will prove there exists a unique multiplication inverse of a by showing the set $\{a \cdot_p b\}_{b \in S_p}$ is equal to all of S_p.

Suppose the contrary that $\exists b \neq b' \in S_p$ such that $a \cdot_p b = a \cdot_p b'$. It follows that $0 = (a \cdot_p b) - (a \cdot_p b') = a(b -_p b')$. This means that $p \mid a(b - b')$. Note that $a \in S_p$, so $p \nmid a$. Recalling that $b, b' \in S_p$, it follows that $-(p-1) \leq b - b' \leq p - 1$. Further, $b \neq b'$ and $b - b' \neq 0$, and therefore $p \nmid (p - p')$.

However, this contradicts that p is prime. Because p is prime, if it divides the product of $a(b - b')$, either $p \mid a$ or $p \mid (b - b')$.

2.3.2 Interesting properties

The size of any finite field is a prime power: p^s for some prime p and integer $s \geq 1$.

For any prime power $q = p^s$, there exists a **unique** finite field $\mathbb{F}_q$ with q elements, up to isomorphism[2].

3 Linear Subspaces over Finite Fields

The punchline here is that this is bog standard linear algebra as we've seen before, but with coefficients/arithmetic over $\mathbb{F}_q$ instead of $\mathbb{R}$

Formal Definition $S \subseteq \mathbb{F}_q$ is a linear subspace if all the following properties hold.

3.1 Properties

3.1.1 Closure over addition

$\forall \vec{x}, \vec{y} \in S$, then $\vec{x} + \vec{y} \in S$, where $+$ means component wise addition over $\mathbb{F}_q$

3.1.2 Closure over scalar multiplication

$\forall a \in \mathbb{F}_q, \vec{x} \in S$, then $a \cdot \vec{x} \in S$, where $\cdot$ means component wise multiplication by a over $\mathbb{F}_q$.

Note that this implies that the zero vector $\vec{0}$ is contained in any linear space.

3.2 Span

Given a set of vectors $B = \{\vec{v}_1, \ldots, \vec{v}_l\}$, the **span** of B over $\mathbb{F}_q$ is defined as:

$$\text{Span}(B) = \left\{ \sum_{i=1}^{l} a_i \vec{v}_i \middle| a_i \in \mathbb{F}_q; \forall 1 \leq i \leq l \right\}$$

3.3 Linear Independence

We say that $\vec{v_1}, \ldots, \vec{v}_l$ are **linearly independent** over $\mathbb{F}_q$ if $\nexists a_1, \ldots, a_l \in \mathbb{F}_q$ such that a_i not all 0 and:

$$\sum_{i=1}^{l} a_i \vec{v}_i = \vec{0}$$

Equivalently, this means $\forall 1 \leq i \leq l$, it holds that $\vec{v}_i \notin \text{Span}(\vec{v}_1, \ldots, \vec{v}_l)$

[2] renaming of the elements

3.4 Rank

The **rank** of a matrix in ${\mathbb{F}_q}^{k\times n}$ is the maximum number of linearly independent rows (or columns). A matrix ${\mathbb{F}_q}^{k\times n}$ with rank $\min(k,n)$ is said to have *full rank*.

Note that basic facts from linear algebra such as row rank, column rank, etc hold here as well.

3.5 Properties

If $S \subseteq {\mathbb{F}_q}^n$ is a linear subspace,

$|S| = q^k$ for some $k \geq 0$. This k is called the dimension of S.

There exists $\vec{v}_1, \ldots, \vec{v}_l \in S$ called the basis elements such that $S = \text{Span}(\vec{v}_1, \ldots, \vec{v}_l)$.

In other words, there exists a full rank matrix $G \in {\mathbb{F}_q}^{k\times n}$ such that $\forall \vec{x} \in S : \vec{x} = \vec{a}G$ for some coefficient vector $\vec{a} \in {\mathbb{F}_q}^k$. This matrix G is called a generator matrix for S.

There exists a full rank matrix $H \in {\mathbb{F}_q}^{(n-k)\times n}$ such that $\forall \vec{x} \in S : \vec{x}H^\intercal = \vec{0}$. This matrix H is a *parity check* matrix for S.

$GH^\intercal = 0$

3.6 Parity as inverse of generator

Suppose $G \in {\mathbb{F}_q}^{k\times n}$ is a generator matrix for space $S_1 \subseteq {\mathbb{F}_q}^n$.

As a proof sketch, we need to show that $S_1 \subseteq S_2$, and that $\dim S_1 = \dim S_2$

4 Linear Codes

Intuition Codes in general are just some subset of the language. Linear codes have a repeating, patterned structure.

Linear codes will be linear subspaces over some alphabet Σ. For this to make sense, Σ must support certain arithmetic operations (e.g. addition, multiplication, etc). We abstractly characterize these properties as a field.

4.1 Formal Definition

$C \subseteq {\mathbb{F}_q}^n$ is a linear code if it is a linear subspace of ${\mathbb{F}_q}^n$.

Notation If $C \subset {\mathbb{F}_q}^n$ has dimension k and distance d, we will refer to it as an $[n, k, d]_q$ code, or sometimes just a $[n, k]_q$ code[3].

From the previous theorem, it follows that we can characterize a $[n, k]_q$ code C as either:

- C is generated by a $k \times n$ generator matrix G
- C is characterized by a $(n - k) \times n$ parity check matrix H.

This means that any $[n, k]_q$ code can be represented by $\min(nk, n(n - k))$ elements in $\mathbb{F}_q$.

Encoded by translating the message $\vec{a} \in {\mathbb{F}_q}^k$ to a codeword $\vec{c} \in {\mathbb{F}_q}^n$ given a generator matrix G by $O(nk)$ $\mathbb{F}_q$ operations (by matrix multiplication).

Error detection given the parity check matrix H by $O(n(n - k))$ $\mathbb{F}_q$ operations (by matrix multiplication): determine if $\vec{y} \in {\mathbb{F}_q}^n$ is a valid code word.

4.2 Distance equivalence to Hamming weight

For any $[n, k, d]_q$ code C:

$$d = \min_{\vec{0} \neq \vec{c} \in C} wt(\vec{c})$$

Where $wt(\vec{c}) = \Delta(\vec{c}, \vec{0})$.

The proof was in HW1.

4.3 Distance equivalence to parity check linear dependence

For some $[n, k, d]_q$ code C with parity-check matrix H, the distance of C is the minimum number of linearly dependent columns of H.

The proof will be in HW2[4].

5 Hamming Codes

6 Examples

6.1 Fields

- $(\mathbb{R}, +, \cdot)$ is a field.
- $(\mathbb{Z}, +, \cdot)$ is not a field, because it lacks a multiplicative inverse.
- $(\{0, 1\}, \oplus, \wedge)$ is a field.

6.2 Linear Subspaces over Finite Fields

$S_1 \subseteq \mathbb{F}_5^3$ is:

$$S_1 = \{(0,0,0), (1,1,1), (2,2,2), (3,3,3), (4,4,4)\}$$

$S_2 \subseteq \mathbb{F}_3^3$ is:

$$S_2 = \begin{matrix} \{(0,0,0) & (1,0,1) & (2,0,2) \\ (0,1,1) & (1,1,2) & (2,1,0) \\ (0,2,2) & (1,2,0) & (2,2,1)\} \end{matrix}$$

[3]Square brackets always mean a linear code for the purposes of this course

[4]Hint for the proof: show $l \leq d$ and $d \leq l$, and recall that $\vec{x} \in C$ iff $H\vec{x}^\intercal = \vec{0}$

6.3 Linear Codes

$[7, 4, 3]_2$ **Hamming Code** This code from last week and the homework has:

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

and

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