Math 127, Mon Mar 15

- Use a laptop or desktop with a large screen so you can read these words clearly.
- In general, please turn off your camera and mute yourself.
- Exception: When we do groupwork, please turn both your camera and mic on. (Groupwork will not be recorded.)
- Please always have the chat window open to ask questions.
- Reading for today: 6.2–6.3.
- Reading for Wed: 6.4, 7.1.
- PS05 due tonight; PS06 outline due Wed night.
- Problem session Fri Mar 19, 10am–noon.

Last: Parity check and repetition codes

Parity check: Suppose we have n data bits $x_1, \ldots, x_n$ to transmit. We can add a **parity check** bit

$$x_0 = x_1 + \cdots + x_n \pmod 2$$

Codewords: words with checksum 0

to our message, and transmit $(x_0, x_1, \ldots, x_n)$. Note:

$$x_0 + x_1 + \cdots + x_n = 0 \quad (\text{in } \mathbf{F}_2)$$

So: If that checksum is = 1, error.

Repetition: Suppose we want to transmit one data bit $x \in \mathbf{F}_2$. We repeat x three times:

send: 000 = x   msg msg = 0 (e.g.)

Codewords: {000,111}

= span(111)

So if one error occurs in transmission: recv: 001 = y

Receiver thinks: More likely to have 1 error than 2, so most likely msg is 000.

we can fix it by **majority logic**.   m' = 0   "y'"

Binary linear codes

$= \{0, 1\}$

Definition

We define a **bit** to be an element of $\mathbf{F}_2$, and we define a **bitstring of length** n to be an element of $\mathbf{F}_2^n$.

$\rightarrow 01101$

Definition

binary

A **code** is a subset $\mathcal{C}$ of $\mathbf{F}_2^n$. Elements (vectors) of a code are called **codewords**.

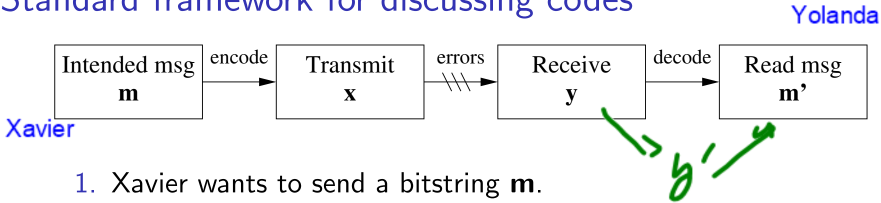
I.e., codewords are the possible messages that could have been transmitted without errors. The idea is that if you pick your code well, it should be possible to tell if an error has occurred, and if you pick it really well, it should be possible to correct the error.

Definition

A **binary linear code** $\mathcal{C}$ **of length** n is a subspace $\mathcal{C}$ of $\mathbf{F}_2^n$.

Remember: n is the number of coordinates (bits) in any codeword. But the dimension of C must be smaller than n for C to be useful.

Standard framework for discussing codes



1. Xavier wants to send a bitstring **m**.
2. Xavier **encodes** the message **m** to some codeword $\mathbf{x} \in \mathcal{C}$.
3. Xavier transmits **x**, Yolanda receives **y**.
4. Yolanda **decodes y** to the message $\mathbf{m}'$, in steps:
   - First, Yolanda **corrects y** to a valid codeword $\mathbf{y}' \in \mathcal{C}$.
   - Yolanda then **reads** $\mathbf{y}'$ as a message $\mathbf{m}'$.

Algebraic model for errors: Let $\mathbf{e}_i$ be the vector in $\mathbf{F}_2^n$ whose ith coordinate is 1 and whose other coordinates are all 0. One error in bit i means:

$$\mathbf{y} = \mathbf{x} + \mathbf{e}_i.$$

Two errors in bits i and j:

$$\mathbf{y} = \mathbf{x} + \mathbf{e}_i + \mathbf{e}_j.$$

Ex Transmit $\underline{x} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$

Error in bit 1:

$$\underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{\underline{x}} \longrightarrow \underbrace{\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{\underline{y}} = \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{\underline{x}} + \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\underline{e}_1}$$

Errors in bits 1,2: $\underline{y} = \underline{x} + \underline{e}_1 + \underline{e}_2$

How to define/describe a binary linear code of len n.

Two ways:

height of matrix is length of the code

Definition

Let G be an $n \times k$ matrix over $\mathbf{F}_2$. To say that G is the **generator matrix** of $\mathcal{C}$ of length n means that $\mathcal{C} = \mathrm{Col}(G)$.

a code

Definition

Let H be a $k \times n$ matrix over $\mathbf{F}_2$. To say that H is the **parity check matrix** of a binary linear code $\mathcal{C}$ of length n means that $\mathcal{C} = \mathrm{Null}(H)$.

I.e.: A generator matrix defines a code as a column space, and a parity check matrix defines a code as a nullspace.

width of the matrix is length of the code

Back to our examples

$x_0 + x_1 + \cdots + x_n = 0$

length of code is n+1, dimension is n.

Parity check code: The parity check code of length $n+1$ is the nullspace $\mathcal{C}$ of the $1 \times (n+1)$ matrix $H = [1 \; \ldots \; 1]$. In other words, $\mathbf{x} \in \mathbf{F}_2^{n+1}$ is in $\mathcal{C}$ exactly when $H\mathbf{x} = 0$.
Encoding is $x_0 = x_1 + \cdots + x_n$, transmit $(x_0, x_1, \ldots, x_n)$. If received message $\mathbf{y}$ satisfies $H\mathbf{y} = \mathbf{0}$, read off bits $x_1, \ldots, x_n$ as message; otherwise notify Xavier that there was an error.

Repetition code: The repetition code of length n is the span $\mathcal{C}$ of the column of the generator matrix $G = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$.

Length of code is n,
Dimension is 1

Encoding the bit x means multiplying Gx, transmit Gx, correct received bits by majority logic, then use any bit as the message bit.

The Hamming 7-code $\mathcal{H}_7$

Defined in two ways:

- $\mathcal{H}_7$ is the nullspace of the parity check matrix

$$H_7 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

- $\mathcal{H}_7$ is the column space of the generator matrix

$$G_7 = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

H_7 is in RREF, so if we think of H_7 as a system of linear equations and use our standard methods, we get the columns of G_7.

More often use $\mathcal{H}_7 = \mathrm{Null}(H_7)$.

I.e., x is a codeword <=> $H_7 \underline{x} = \underline{0}$.

Mnemonic for parity check matrix H_7

The ith column of parity check matrix H_7 is precisely the binary digits of the number i, written upside down:

$$H_7 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\begin{array}{cccc} \uparrow & & & \uparrow \\ \text{col } 1 & & & \text{col } 4 \end{array}$$

$1 = 001$

$2 = 010$

$3 = 011$

$4 = 100$

$5 = 101$

$6 = 110$

$7 = 111$

How to use the Hamming 7-code

send 4 data bits, b/c dim(Hamming 7)=4

Suppose Xavier wants to send $\mathbf{m} \in \mathbf{F}_2^4$ to Yolanda.

1. **Encode:** Bits x_3, x_5, x_6, and x_7 are precisely the contents of $\mathbf{m}$, and other bits x_1, x_2, and x_4 satisfy:

$$\begin{aligned} x_1 &= x_3 + x_5 + x_7 && \leftarrow \text{row 1} \\ x_2 &= x_3 + x_6 + x_7 && \leftarrow \text{row 2} \\ x_4 &= x_5 + x_6 + x_7. && \leftarrow \text{" 3} \end{aligned}$$

2. **Transmit x, receive y.**
3. **Decode y** by:
   - Let $\mathbf{s} = H_7\mathbf{y} \in \mathbf{F}_2^3$ be the **syndrome** of $\mathbf{y}$.
     If $\mathbf{s} = \mathbf{0}$, $\mathbf{y} \in \mathcal{H}_7$, so choose $\mathbf{y}' = \mathbf{y}$.
     Otherwise, read $\mathbf{s}$ as the binary digits of a number i, assume error in bit i, and choose $\mathbf{y}' = \mathbf{y} + \mathbf{e}_i$.
   - **Read** $\mathbf{m}'$ off of bits 3, 5, 6, and 7 of $\mathbf{y}'$.

An example

$$H_7 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad \mathcal{H}_7 = Null(H_7)$$

$$\text{Data: } \underline{m} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\underline{x} = \begin{array}{c} \\ \\ 3 \\ \\ 5 \\ 6 \\ 7 \end{array} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{array}{l} = x_3 + x_5 + x_7 \\ = x_3 + x_6 + x_7 \\ \\ = x_5 + x_6 + x_7 \\ \\ \\ \\ \end{array}$$

error bit 5

$$\underline{y} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \underline{x} + \underline{e}_5$$

Matrix*vector is l.c. of columns of matrix with coefficients from entries of the vector

$$H_7 \underline{y} = \underbrace{\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}}_{H_7} \underbrace{\begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\underline{y}} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

the bin dig of 5!

So we correct bit number 5 in y, to get
y' = 0111100. Error corrected!

And then Yolanda can read bits 3,5,6,7
to get m' = 1100.

Proof that the Hamming 7-code corrects one error

Theorem

If $\mathbf{y} = \mathbf{x} + \mathbf{e}_i$ (i.e., one error in bit i), then the syndrome $\mathbf{s} = H_7\mathbf{y}$ is precisely the binary digits of i.

Pf

$$H_7 \underline{y} = H_7(\underline{x} + \underline{e}_i)$$

$$= \underbrace{H_7 \underline{x}}_{x \in \mathrm{Nul}(H_7)} + H_7 \underline{e}_i$$

$$= \underline{0} + H_7 \underline{e}_i = \text{ith col of } H_7$$

And we chose the columns of H_7 so that the ith column is the binary digits of the number i.

Extension: The Hamming 8-code $\mathcal{H}_8$

$\mathcal{H}_8$ is defined to be the nullspace of the parity check matrix

$$H_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Note:

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

So to be consistent with the Hamming 7-code, we write $\mathbf{x} \in \mathcal{H}_8$ as

$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_7 \end{bmatrix}.$$

Key properties of $\mathcal{H}_8$

Theorem

The Hamming 8-code $\mathcal{H}_8$ is the Hamming 7-code $\mathcal{H}_7$, extended by a parity check bit x_0; and $\mathcal{H}_8$ corrects 1 error and detects 2 errors.

See PS06.

Generalizations?

Can we find other, similar codes, but maybe better?

Definition

An $[n, k, d]$ **binary code** is a binary linear code $\mathcal{C}$ such that:

- $\mathcal{C}$ has length n;
- $\dim \mathcal{C} = k$; and
- d is the smallest number of nonzero coordinates appearing in a nonzero codeword of $\mathcal{C}$.

The numbers n, k, and d are called the **length**, **dimension**, and **minimum distance** of **C**, respectively.

Examples

Example: Parity check code of length $n + 1$.

Example: Repetition code of length n.

Example: Hamming code $\mathcal{H}_7$.

Example: Hamming code $\mathcal{H}_8$.

An IDEA: Look at a code using geometry

Definition

$\mathbf{x}, \mathbf{y} \in \mathbf{F}_2^n$; **Hamming distance** between $\mathbf{x}$ and $\mathbf{y}$ is:

$$\begin{aligned} d(\mathbf{x}, \mathbf{y}) &= \text{the number of coordinates in which } \mathbf{x} \text{ and } \mathbf{y} \text{ differ} \\ &= \text{the number of nonzero coordinates in } \mathbf{x} - \mathbf{y} \\ &= \text{the number of coordinate changes needed to go from } \mathbf{x} \text{ to } \mathbf{y}. \end{aligned}$$

Hamming weight $\mathrm{wt}(\mathbf{x})$ is the number of nonzero coordinates of $\mathbf{x}$, i.e.:

$$\mathrm{wt}(\mathbf{x}) = d(\mathbf{x}, 0), \qquad d(\mathbf{x}, \mathbf{y}) = \mathrm{wt}(\mathbf{x} - \mathbf{y}).$$

Why is Hamming distance a distance?

Definition

A **Hamming path of length** k in $\mathbf{F}_2^n$ is a sequence $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathbf{F}_2^n$ such that for $1 \leq i \leq k$, the vectors $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ differ in exactly one coordinate (i.e., $\mathbf{x}_i - \mathbf{x}_{i-1}$ has exactly one nonzero coordinate). We also say that the path $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k$ goes from $\mathbf{x}_0$ to $\mathbf{x}_k$.

Picture in a cube:

Hamming distance is a path distance

Theorem

For $\mathbf{x}, \mathbf{y} \in \mathbf{F}_2^n$, the Hamming distance $d(\mathbf{x}, \mathbf{y})$ is precisely the length of a shortest Hamming path from $\mathbf{x}$ to $\mathbf{y}$.