Welcome! Everything is fine.

When you're connected by zoom:

- Use a laptop or desktop with a large screen so you can read these words clearly.
- To conserve bandwidth, please turn off your camera.
- Please mute your microphone unless I call on you.
- Please have the chat window open to ask questions.
- Reading for today: 6.3–6.4; for Wed: 7.1–7.2.
- PS07 outline due Wed Mar 18.
- PS07 due Wed Mar 25.

No office hour today at noon (sorry, I double-booked myself)

No exam 2 until after break.

Last time: Hamming 7-code

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

binary digits of 3 (i.e., 011), upside down

$6 = 110$ binary
$= 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$

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

binary digits of 5, upside down

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

If we solve $H_7 x = 0$ using our usual methods, the columns of G_7 form the basis we get for Null(H_7)

$$G_7 = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

4 vectors in a basis for Hamming 7, so dimension 4

Key point: ith column of H_7 is binary digits of i upside down.

Encoding and decoding Hamming 7

1. Xavier copies four data bits of $\mathbf{m}$ into x_3, x_5, x_6, x_7. Other bits x_1, x_2, and x_4 satisfy

Top row of H_7 is 1010101, which gives the equation x1+x3+x5+x7 = 0, then b/c +1 = -1 (mod 2):

$$\begin{aligned} x_1 &= x_3 + x_5 + x_7 \\ x_2 &= x_3 + x_6 + x_7 \\ x_4 &= x_5 + x_6 + x_7. \end{aligned} \tag{3}$$

(mod 2)

all seven bits

2. Xavier transmits $\mathbf{x}$, Yolanda receives $\mathbf{y}$.
3. Yolanda then decodes $\mathbf{y}$:
   - 3.1 First, Yolanda corrects $\mathbf{y}$ to a codeword $\mathbf{y}'$ as follows. Let $\mathbf{s} = H_7\mathbf{y} \in \mathbf{F}_2^3$ be the syndrome of $\mathbf{y}$.
     If $\mathbf{s} = \mathbf{0}$, then $\mathbf{y}$ is a codeword, so Yolanda chooses $\mathbf{y}' = \mathbf{y}$.
     Otherwise, Yolanda reads $\mathbf{s}$ as the binary digits of a number i and chooses $\mathbf{y}'$ to be $\mathbf{y}$ with its ith bit flipped (i.e., $\mathbf{y}' = \mathbf{y} + \mathbf{e}_i$).
   - 3.2 Yolanda reads the message $\mathbf{m}'$ off of bits 3, 5, 6, and 7 of $\mathbf{y}'$.

Encoding and decoding Hamming 7: Example

$$H_7 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \qquad (4)$$

Xavier

$m = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$ → $\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$ x → error → $\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$ y Yolanda

$y = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$

$s = H_7 y$

matrix-vec mult is l.c. of cols w coeffs from vector

$= \text{col } 1 + \text{col } 5 + \text{col } 7$

$= \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ column 3!

So Yolanda corrects bit 3 to 1 and reads off intended message.

Proof that decoding works

Point of theorem: Hamming 7 corrects one error.

algebraic model of one error

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

e_i is vector w/ 1 in coord i, 0 elsewhere

Proof.
Since $\mathbf{x} \in \mathcal{H}_7 = \text{Null}(H_7)$, $H_7\mathbf{x} = \mathbf{0}$. Therefore,

syndrome

$$\mathbf{s} = H_7\mathbf{y} = H_7(\mathbf{x} + \mathbf{e}_i) = H_7\mathbf{x} + H_7\mathbf{e}_i = \mathbf{0} + H_7\mathbf{e}_i = H_7\mathbf{e}_i, \quad (5)$$

matrix mult is distributive!

the ith column of H_7. However, H_7 is the matrix whose ith column is the binary digits of i. □

Hamming 8-code

Definition

Hamming 8-code $\mathcal{H}_8$ is nullspace of

$$H_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{6}$$

Delete the first row and first column of H_8, get H_7.
So to be consistent with the Hamming 7-code, we write an arbitrary element of $\mathcal{H}_8$ as $\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_7 \end{bmatrix} \in \mathcal{H}_8$.

Facts of Hamming 8

Same codewords as Hamming 7, but x0 is a parity check bit added to each one.

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.

Proof is HW.

Abstract code stats

6.4

Notation:

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

- $\mathcal{C}$ has length n; (C is subspace of F_2^n)
- $\dim \mathcal{C} = k$; and
- d is the smallest number of nonzero coordinates appearing in a nonzero codeword of $\mathcal{C}$.

$n =$ **length**, $k =$ **dimension**, $d =$ **minimum distance** of **C**.

Turns out that having large min dist is same as correcting lots of errors.

Examples of abstract code stats

[n+1,n,2] data bits x1,...,xn, parity bit x0

Parity check code of length $n+1$

length is n+1, dimension is n b/c data bits chosen freely
any nonzero codeword has even # of 1s, so min distance is 2 (from 1100000).
That is: X never transmits 0100000 b/c that doesn't satisfy x0+x1+...+xn=0.

Repetition code of length n {000000000,111111111} length 9

length is n; dimension is 1 b/c {111111111} is a basis; min dist is
n b/c all 1s vector is the only nonzero codeword. So: [n,1,n] code.

Hamming codes $\mathcal{H}_7$ and $\mathcal{H}_8$

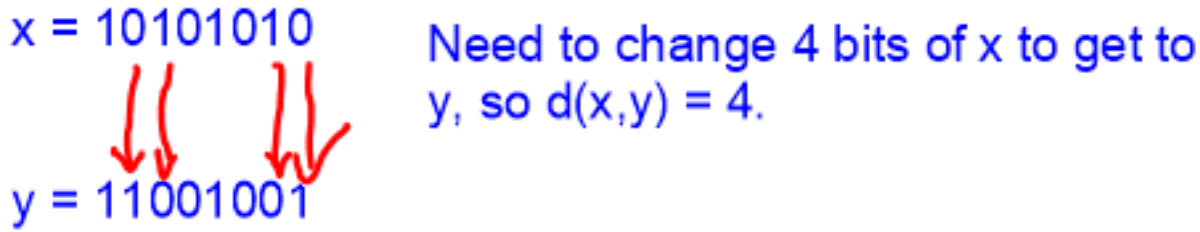
Hamming 7: Length 7, dim 4
Hamming 8: Length 8, dim 4 (calculation omitted)
Min distance is not easy to calculate -- on HW, by brute force
(list all nonzero codewords and check). Turns out to be 3 for
Hamming 7 and 4 for Hamming 8, so these are [7,4,3] and [8,4,4]
codes.

Hamming distance

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

$$\begin{aligned} d(\mathbf{x}, \mathbf{y}) &= \text{the number of coordinates in which } \mathbf{x} \text{ and } \mathbf{y} \text{ differ} \\ &= \text{the number of nonzero coordinates in } \mathbf{x} - \mathbf{y} \\ &= \text{the number of coordinate changes needed from } \mathbf{x} \text{ to } \mathbf{y}. \end{aligned} \tag{7}$$

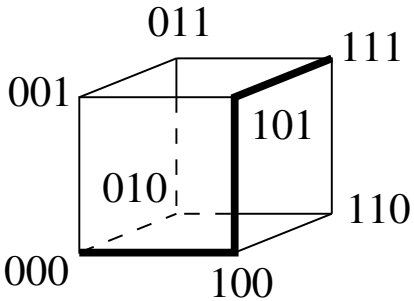
Hamming weight of $\mathbf{x}$ is $d(\mathbf{x}, \mathbf{0})$ = number of nonzero coords in $\mathbf{x}$.
Example:



Hamming paths and Hamming 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 $\mathbf{x}_k$.



Geometrically: F_2^n is an n-dim cube! (whoooooaaaa)

Theorem

This helps b/c we can think of Hamming dist w/paths.

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* $\mathbf{y}$.

Why:

Each step of a Hamming path changes exactly one coord.

Hamming distance is a metric

Definition

A **metric** on a set X is a function $d : X \times X \to \mathbf{R}$ (i.e., two inputs in X, output is a real number) s.t.:

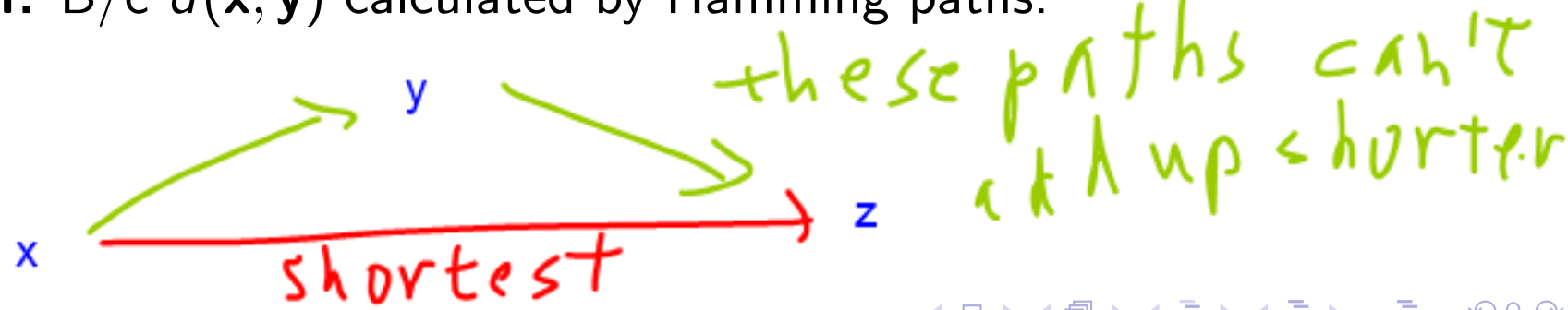
1. $d(x, y) \geq 0$.
2. $d(x, y) = 0$ if and only if $x = y$.
3. $d(x, y) = d(y, x)$.
4. (Triangle inequality) $d(x, z) \leq d(x, y) + d(y, z)$.

No shortcut thru extra destination y

Theorem

Hamming distance $d(\mathbf{x}, \mathbf{y})$ is a metric on $\mathbf{F}_2^n$.

Proof: B/c $d(\mathbf{x}, \mathbf{y})$ calculated by Hamming paths:



Nearest neighbor error correction

The **nearest neighbor** error-correction method in $\mathcal{C}$ is:

- If there is a unique $\mathbf{y}' \in \mathcal{C}$ such that $d(\mathbf{y}, \mathbf{y}')$ is minimized, we correct $\mathbf{y}$ to $\mathbf{y}'$. (Think: Yolanda receives y, wants to fix to y')
- If there is more than one vector $\mathbf{y}' \in \mathcal{C}$ such that $d(\mathbf{y}, \mathbf{y}') > 0$ is minimized, we state that $\mathbf{y}$ has been detected as an erroneous transmission, but cannot be corrected. (The idea is that the different $\mathbf{y}'$ minimizing $d(\mathbf{y}, \mathbf{y}')$ are equally probable as intended transmissions.)

Note: This is a scheme used only in theory, not in practice, b/c the most naive way to implement this would be to list all possible received words (2^n of them!) and list nearest neighbor(s) of each. This scheme only exists to show that you *can* correct a certain number of errors, not a practical way to do so.

Min weight and error correction

Theorem

Let $\mathcal{C}$ be a binary linear code with minimum distance d. Then the nearest neighbor method, applied to $\mathcal{C}$, corrects $\lfloor (d-1)/2 \rfloor$ errors and detects $\lfloor d/2 \rfloor$ errors.

Proof: