REMINDERS: Make sure you:
(1) Sign up on piazza + consider filling in the feedback polls
(2) Read the syllabus on course webpage
(3) Make sure you have 4/5/45 on Autolab

RECAP:
- $C \subseteq \Sigma^n$ or $C : [M] \rightarrow \Sigma^n$ where $M = [C]$  
- $\Sigma \rightarrow$ alphabet, $n \rightarrow$ block length
- $q = |\Sigma|$ (alphabet size)
- $\text{Dim}(C) = \log_q |C| = k$
- Rate $C$ = $\frac{k}{n}$
- $C \oplus (x_1, x_2, x_3, x_4) = (x_1, x_2, x_3, x_4, x_1 \oplus x_2 \oplus x_3 \oplus x_4)$
- $C_{3,1} \oplus (x_1, x_2, x_3, x_4) = (x_1, x_1, x_1, x_2, x_2, x_3, x_3, x_3, x_4, x_4, x_4)$
- High level goal: Optimal tradeoff between redundancy & rate

PROOF READING: Volunteer for today
Make sure you have a section each on Types, Formatting, suggestions for improvement

ERROR CORRECTION!

Definition (Encoding): $C \subseteq \Sigma^n \equiv \Sigma : [|C|] \rightarrow \Sigma^n$

$\Rightarrow$ Need ‘reverse’ for error correction

Definition (Decoding func): $D : \Sigma^n \rightarrow [|C|]

Definition (error correction): $C \subseteq \Sigma^n$. Let $t > 1$ be an integer.

$C$ is $t$-error correcting if $\exists$ a decoding $D$ s.t.

$\exists m \in [|C|]$ and any error pattern $e$ with $\leq t$ errors,

$D(C(m) + e) = m$

$\overline{m} \rightarrow C(\overline{m}) \overset{e}{\rightarrow} C(m) + e \overset{Dec}{{\downarrow}} \overline{m}$

Channel Encoding

received word
Def (error detection): \( C \subseteq \mathbb{F}_2^n \), \( t \geq 1 \) integer. \( C \) is \( t \)-error detecting if \( \exists \) a error detection fn: \( D : \mathbb{F}_2^n \rightarrow \{0,1\}^t \)

\[ D(C, y) = \sum_{i=0}^{t-1} y_i \] if \( y \in C \)
o/w

Eg: CRC checksum on TCP/IP

\[ C_0: \quad q = 2, \ n = 5, \ k = 4, \ R = \frac{4}{5} \]

\[ C_{3,\text{rep}}: \quad q = 2, \ n = 12, \ k = 4, \ R = \frac{1}{3} \]

Claim 1: \( C_{3,\text{rep}} \) is 1-error correcting.

Pf sketch: Consider \( y \in \{0,1\}^{12} = (\overline{y}_1, \overline{y}_2, \overline{y}_3, \overline{y}_4) \)

Set \( x_j = \text{Maj} (\overline{y}_j) \)

Claim 2: \( C_{3,\text{rep}} \) is not 2-error correcting.

Pf sketch:

\[
\begin{array}{c}
\begin{array}{c}
111
\end{array}
\end{array}
\rightarrow
\begin{array}{c}
\begin{array}{c}
\text{2 errors}
\end{array}
\end{array}
\rightarrow
\begin{array}{c}
\begin{array}{c}
010
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
000
\end{array}
\end{array}
\rightarrow
\begin{array}{c}
\begin{array}{c}
1\text{ error}
\end{array}
\end{array}
\]

Issue: Decoder cannot say for sure if transmitted bit is 0 or 1. (There is NO side information)

So far...

Adversarial noise model (Hamming '50)

Stochastic (Eg: BSCp (Shannon '48))

Let flip each bit independently w/ \( p \) prob.

Noise modeling is an important task (but we'll ignore this)

Note: Can make up noise models that are "better" for certain codes BUT we will not.

Prop 1. \( C_0 \) is 1-error detecting.

Pf idea: Error-detecting fn: parity of all 5 bits

Cor: \( C_0 \) can detect odd # errors

Prop 2. \( C_0 \) is not 1-error correcting

\[ 00000 \rightarrow 11000 \]

\[ 10001 \rightarrow 10001 \]

\[ 0000 \rightarrow 10000 \]
DISTANCE OF A CODE

Def (Hamming distance): \( \overline{a}, \overline{w} \in \mathbb{E}^n \), \( \Delta(\overline{a}, \overline{w}) \)

\[ \Delta(111, 010) = 2 \quad \Delta(111, 100) = \Delta(111, 001) = 2 \]

Def (minimum distance) of a code \( C \)

\[ d(C) = d_0 = \min_{\overline{c}_1 \neq \overline{c}_2 \in C} \Delta(\overline{c}_1, \overline{c}_2) \]

Claim: \( d(C_{\oplus}) = 2 \)

\[ \geq 2 : \quad \Rightarrow \Delta(m_1, m_2) \geq 2 \Rightarrow (C_{\oplus}(m_1), C_{\oplus}(m_2)) \]

\[ \Delta(m_1, m_2) = 1 \Rightarrow \text{parity } (m_1) \neq \text{ parity } (m_2) \]

\[ \Rightarrow \Delta(C_{\oplus}(m_1), C_{\oplus}(m_2)) = 2 \]

\[ \leq 2 : \quad \text{Pick any } m_1, m_2 \text{ s.t. } \Delta(m_1, m_2) \leq 2 \]

\[ \geq 3 : \quad \text{for any } m_1 \neq m_2, \text{ they differ in } \geq 3 \text{ bits} \]

\[ \leq 3 : \quad m_1 = 0000, m_2 = 1000 \Rightarrow \Delta(C_{3, \text{rep}}(m_1), C_{3, \text{rep}}(m_2)) = 3 \]

PROPOSITION: The following statements are equivalent for a code \( C \).

1. \( d(C) \geq 2 \)
2. \( C \) is \( d-1 \)-error correcting (\( d \) is odd)
3. \( C \) is \( (d-1) \)-error detecting

1 \( \oplus \): 1-error detecting but not 2-error detecting
1 \( \text{rep.} \): 1 — correcting — 2 — correcting.

Today: \( (1) \equiv (2) \)
First "official" algo:

Maximum Likelihood decoder (MLD)

\[ D_{MLD} : \mathcal{E}^n \rightarrow \mathcal{C} \quad \text{Output the closest codeword} \]

\[ D_{MLD}(y) = \arg \min_{\tilde{c} \in \mathcal{C}} \Delta(c, y) \]

Pf of prop: (1) \( \Rightarrow \) (2) \( d = 2t+1 \)

Show: \( \mathcal{C} \) is \( t \)-error correcting

Claim: \( \nexists \ m \in \mathcal{E}^k \), all error patterns with \( \leq t \) errors

\[ D_{MLD}(c(m) + \tilde{e}) = \min_{\tilde{c} \in \mathcal{C}} \Delta(c, \tilde{c}) \]

Pf: For sake of contradiction assume \( \nexists \ m \in \mathcal{E}^k \), \( \exists \tilde{e} \) with \( \leq t \) errors

\[ D_{MLD}(y) \neq (m) \]

\[ \tilde{c} = c(m) \]

\[ \overline{\tilde{c}} = c(\overline{m}) \]

\[ \Delta(c, \overline{\tilde{c}}') \leq \Delta(c, \overline{\tilde{c}}) + \Delta(\overline{\tilde{c}}', y) \]

\[ \leq \Delta(c, \overline{\tilde{c}}) + \Delta(c, \overline{\tilde{c}}) \]

\[ = 2 \Delta(c, \overline{\tilde{c}}) \]

\[ \leq 2t \Rightarrow \text{contradicts the fact that } d(c) = 2t+1 \]