-: 2nd Class :-

Brief Recap of last Class:

- Goal is to formalize the fundamental question in Coding theory "How much redundancy do we need to correct certain amount of error."
- Formalized redundancy via Rate(R) $:= \frac{k \to \text{dimension}}{n \to \text{blocklength}}$
- Examples of two simple codes:

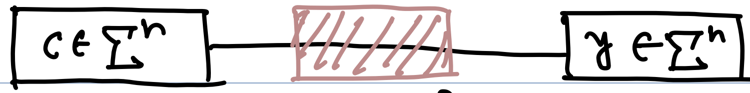
  $C_{\oplus}$ & $C_{3,rep}$

Formalizing Error Correction:

Encoding function: for any code $C \subseteq \Sigma^n$, the alternate view of C, $E : [|C|] \xrightarrow[\text{injective}]{} \Sigma^n$ is called encoding function.

Decoding function: $D : \Sigma^n \to [|C|]$

Recovers original message from corrupt received words $\implies$ Need to understand nature of errors & how it affects the received word.



noisy channel.

Quantifying Errors via Hamming Distance:

Hamming Distance: $u, v \in \Sigma^n$, Hamming distance b/w u & v, $\Delta(u,v)$, is the no. of positions in which u & v differ. & relative Hamming distance,

$$\delta(u,v) = \frac{1}{n}\Delta(u,v) \in [0,1]$$

Example:

$u = 0000$, $v = 10001$, $w = 0101 \in \{0,1\}^4$

$\Delta(u,v) = \Delta(v,w) = 2$ &

$\delta(u,v) = \delta(v,w) = \frac{1}{2}$

Hamming distance is a distance in a very formal mathematical sense: i.e., 1. $\Delta(x,y) \geq 0$.

2. $\Delta(x,y) = 0$ iff $x = y$
3. $\Delta(x,y) = \Delta(y,x)$
4. $\Delta(x,y) \leq \Delta(x,z) + \Delta(y,z)$ [Triangular inequality]

[Exercise].

Quantifying error: hamming distance b/w received & transmitted words.

Formalizing performance of Error Correcting Codes:

t-Error Channel: An n-symbol t-error channel over the alphabet Σ is a function $ch: \Sigma^n \to \Sigma^n$ satisfies $\Delta(v, ch(v)) \leq t$ for $v \in \Sigma^n$.

Error correcting code: C is a t-error correcting code if $\exists$ D s.t. for every $m \in [|C|]$ & every t-error channel ch, $D(ch(E(m))) = m$

$\Delta(E(m), ch(E(m))) \leq t$

$$
\begin{tikzcd}
\boxed{m} \arrow[r, "\text{Encoding}", "E"'] & \boxed{E(m)} \arrow[r, "t\text{-error}", "\text{channel}"'] & \boxed{Ch(E(m))} \arrow[d, "\text{Decoding}", "D"'] \\
 & m \cong E(m) & \arrow[l]
\end{tikzcd}
$$

Example: a codeword $(0,0,0,0)$ of 1-error correcting code & transmitted. Then, the 1-error correcting code will decode it from any of the following received word.

$(0,0,0,0)$, $(1,0,0,0)$, $(0,1,0,0)$, $(0,0,1,0)$ $(0,0,0,1)$.

Some weaker notation:

Error Detection code: C is t-error detecting code if $\exists$ a detecting procedure D s.t. $\forall$ message $m \in [|C|]$ & every received word v with $\Delta(v, E(m)) \leq t$,

D outputs 1 if $v = E(m)$

0 otherwise.

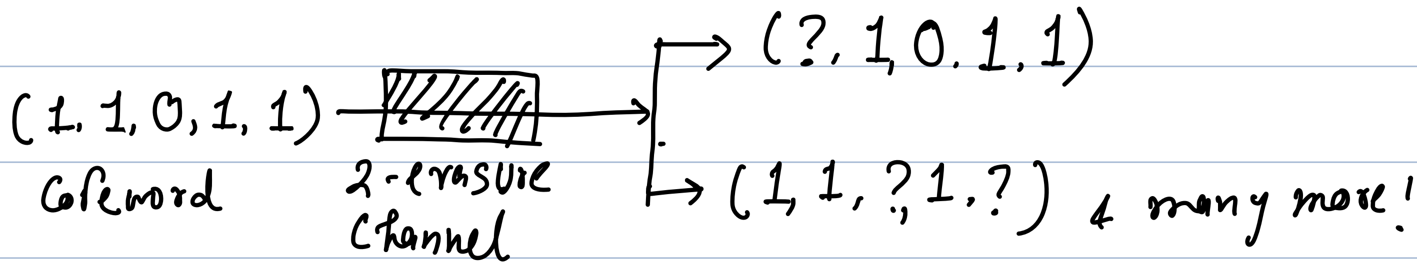
Another benign model of errors referred to as "erasures"
a symbol is merely omitted from the transmission as opposed to being replaced by other symbol.

t-erasure channel: $ch : \Sigma^n \to (\Sigma \cup \{?\})^n$ s.t.

$\forall v \in \Sigma^n$, $\Delta(v, ch(v)) \leq t$

- if we see both v, $ch(v)$ is a string over $\Sigma \cup \{?\}$ then, $\forall i \in [n]$, $v_i \neq ch(v)_i$, we have $ch(v)_i = ?$

Example:



Erasure Correcting Codes: Code $C \subseteq \Sigma^n$ & $t \in \mathbb{N}$,
C is t-erasure correcting code if $\exists$ a decoding function D s.t. $\forall m \in \Sigma^n$, $\forall$ t-erasure channel Ch

$$D(Ch(E(m))) = E(m) \cong m.$$

Error correction capabilities of $C_{\textcircled{f}}$ & $C_{3,rep}$:

$C_{\textcircled{f}}$: $q=2$, $k=4$, $n=5$, $R=4/5$

$C_{3,rep}$: $q=2$, $k=4$, $n=12$, $R=1/3$.

Error correction capability of $C_{3,rep}$:

- trivial decoding function:
  divide into four blocks (y_1, y_2, y_3, y_4)
  each block 3-bits;
  $\forall$ block y_i, output 1 if 1 is the majority bit
  0 otherwise.

Exercise: $C_{3,rep}$ 1-error correcting code.

But it cannot correct two errors:

one received block: $101 \to 000$
$\qquad\qquad\qquad\qquad\quad\ \to 111$

Pin-point: Can correct one error but not two or more!

Error Correction capability of $C_{\oplus}$: Cannot even correct one error!

Example: Received word: $\boxed{10000}$

Possible transmitted word.

$\boxed{00000}$ $\boxed{10001}$

But, can detect one bit of error!

Any odd number of error, but not even number of errors!

Distance of a code:

- a parameter of code which captures its error correction capability.

Distance: For a code C, distance of C, $d(C) = \min\limits_{c_1 \neq c_2 \in C} \Delta(c_1, c_2)$.

Example: $d(C_{3,rep}) = 3$

$d(C_{\oplus}) = 2$.

Formal connection b/w Error-correction & distance:

Proposition: Given a code C, the following are equivalent.

1. $dist(C) \geq d$
2. If d is odd, C can correct $\frac{d-1}{2}$ errors.
3. C can detect $d-1$ errors
4. C can correct $d-1$ erasures.

Remark: For Property 2 for even d: can correct up to $\frac{d}{2}-1$ errors but may not correct $\frac{d}{2}$ errors. [Exercise]

Application: $C_{3,rep}$ can correct up to 1 error

$C_\oplus$ can detect up to 1 error, but cannot 1 error.

Proof Strategy: $P1 \Rightarrow P2, P3, P4$ & $\neg P1 \Rightarrow \neg P2, \neg P3, \neg P4$.

$P1 \Rightarrow P2, P3, P4$:

1. $P1 \Rightarrow P2$: Maximum likelihood decoding (MLD):

$D_{MLD}: \Sigma^n \to C$ s.t.

$\forall y \in \Sigma^n$, $D_{MLD}(y) =$ any $c \in C$ s.t. $\Delta(c,y) = \min_{c' \in C} \Delta(c', y)$

Naive Maximumlikelihood Decoder:

- **Input:** Received word $y \in \Sigma^n$
- **Output:** $D_{MLD}(y)$

1. Pick an arbitrary $c \in C$ & $z \leftarrow c$
2. $\forall c' \in C$
3. if $\Delta(c', y) < \Delta(z, y)$ then
4. $z \leftarrow c'$
5. Return z.

$\boxed{\text{Convince the above algorithm computes } D_{MLD}.}$

Let $d = 2t+1$.

Goal: $\exists$ a decoder D s.t. for at most t errors, it outputs the transmitted message!

Claim: MLD is our desired decoder.

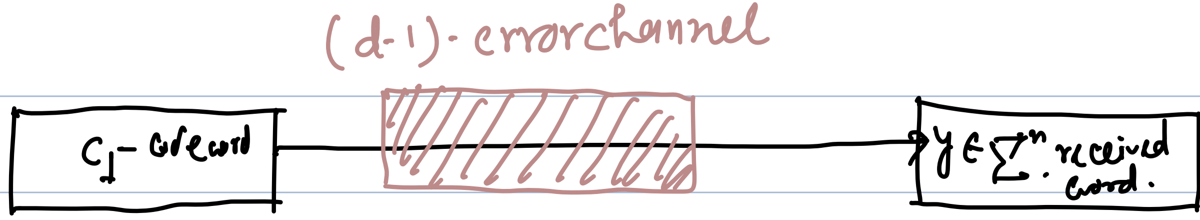
Proof: Let c_1 be a transmitted word & y be the received word & $\Delta(c_1, y) \le t$.

Assume: $D_{MLD}(y) = c_2 \ne c_1 \Rightarrow \Delta(c_2, y) \le t$.

$$\Rightarrow \Delta(c_1, c_2) \le \Delta(c_1, y) + \Delta(y, c_2)$$
$$\le t + t$$
$$< 2t \quad \text{(Contradiction)}$$

P1 $\Rightarrow$ P3

Error detection Algo: if $y \in C$, output 1.
otherwise 0.



If no error happens, then $y = c_1$,

If error happens, then $1 \le \Delta(y, c_1) \le d-1$

$\Rightarrow y \notin C$.

$P1 \Rightarrow P4$: $y \in (\Sigma \cup ?)^n \rightarrow$ received code word.

$\exists$ unique code word $c = (c_1, \ldots, c_n)$ s.t.

$y_i = c_i \;\; \forall i \in [n]$ s.t. $y_i \ne ?$ $\Rightarrow$ Exhaustive search.

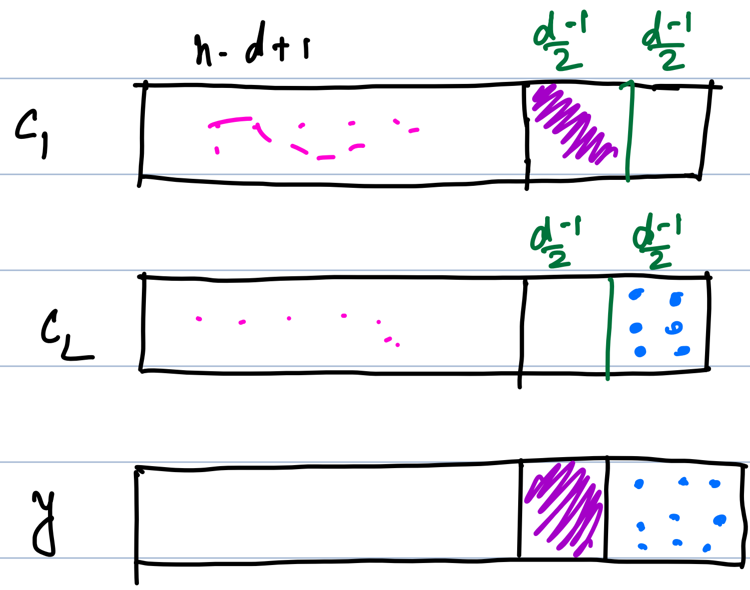
Contradiction: $\exists$ c_1 & c_2 s.t. both agrees with y in unagreed positions; $\Rightarrow \Delta(c_1, c_2) \le d-1$

$\neg P_1 \Rightarrow \neg P_2$, $\neg P_3$, $\neg P_4$:

1. $\neg P_1 \Rightarrow \neg P_2$: $d(C) < d = 2t+1$ & can correct $\frac{d-1}{2}$ errors $(P_1) \Rightarrow \perp$

$\Downarrow$

$\exists c_1, c_2 : \Delta(c_1, c_2) = d-1$



$\Delta(c_1, y) = \Delta(c_2, y) = \frac{d-1}{2}$

$\Rightarrow$ No decoding function can work.

Similar argument proves that $7P_1 \Rightarrow 7P_3, 7P_4$ (Exercise).

Error correction capability $\Leftrightarrow$ distance

Goal: minimizing redundancy & maximizing error-correction

$\Downarrow$

maximizing rate & maximizing distance

Q: What is the largest rate R that a code with distance d can have?

Family of codes:

Until now, we have studied codes of fixed block lengths & dimensions. But, when we talk about algorithmic & lower bound questions related to code, it make more sense to study a family of codes & their asymptotic rate & distance.

Code family, Rate & distance: Let $q \geq 2$. Let $\{n_i\}_{i\geq 1}$ be an increasing sequence of block lengths. Let $\{k_i\}_{i\geq 1}$ & $\{d_i\}_{i\geq 1}$ s.t. $\forall\, i \geq 1\ \exists$ an $(n_i, k_i, d_i)_q$ Code C_i. Then $C = \{C_i\}_{i\geq 1}$ is family of codes.

$$R(C) = \lim_{i\to\infty} \frac{k_i}{n_i}, \text{ when limit exists.}$$

Relative distance of C, $\delta(C) = \lim_{i\to\infty} \frac{d_i}{n_i}$, when limit exists.

Eg: $C^{(i)}_{3\,rep} : \{0,1\}^i \longrightarrow \{0,1\}^{3i}$ & $C = \{C^{(i)}_{3,rep}\}_{i \geq 1}$.

$(a_1, \ldots, a_i) \longrightarrow (a_1, a_1, a_1, a_2, a_2, a_2, \ldots, a_i, a_i, a_i)$.

$$R(C) = \lim_{i \to \infty} \frac{i}{3i} = \frac{1}{3}$$

$$\delta(C) = \lim_{i \to \infty} \frac{3}{3i} = \lim_{i \to \infty} \frac{1}{i} = 0.$$

$\Rightarrow$ formally talk asymptotic running time of algorithms.

$\Rightarrow$ $O(n^2)$-time decoding algorithm for a code $C \Rightarrow$ implicitly $\exists$ a family of code C & that algorithm has running time $O(n^2)$ for large enough block.

Efficient algorithm $\Rightarrow$ poly(n).

Subtle point: $\{C_i\}_{i \geq 1}$, each C_i is the same code but with different parameters; [example: $\{C^{(i)}_{3,rep}\}_{i \geq 1}$.

Q: Given q, what is the optimal tradeoff b/w $R(C)$ & $\delta(C)$ that can be achieved by some family of q-ary codes.

Q: Does $\exists$ a constant q & a q-ary family of codes $\mathcal{C}$ which is asymptotically good; i.e., both $R(\mathcal{C})$ & $\delta(\mathcal{C})$ are positive?

[observe that $\{C^{(i)}_{3,rep}\}$ is not asymptotically good code].