Today
- Decoding Concatenated Codes
- Achieving BSC(p)-capacity
- GMD Decoding

CS5.318.1
Coding Theory
Lecture 11 (2022-10-3)
Instructor: Prahladh Harsha.

Recall Concatenation:

Outer Code: $C_{out} : [N, K, D]_Q$ (Reed Solomon)

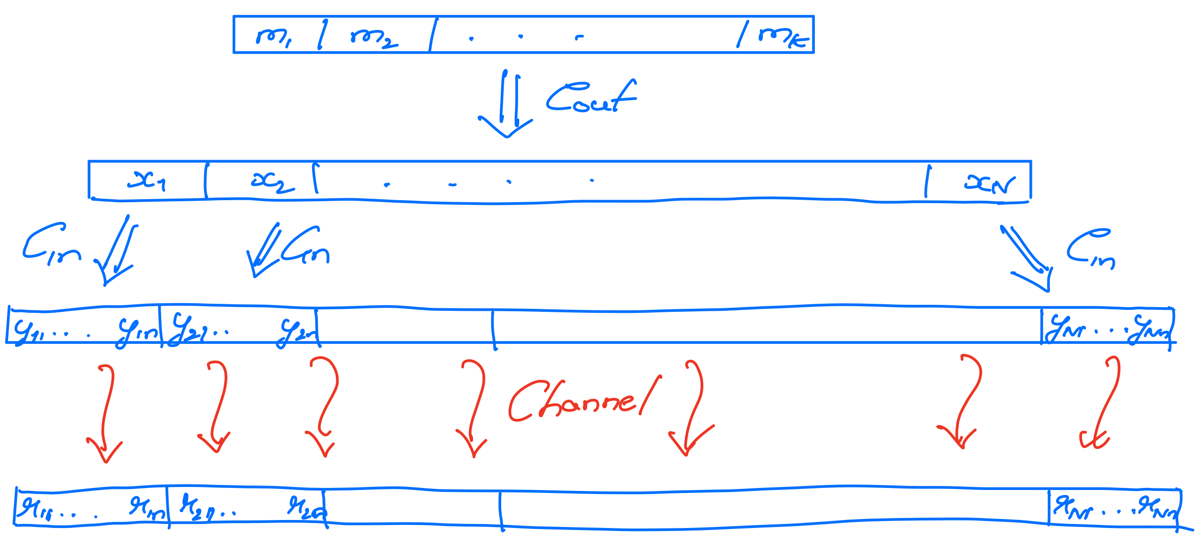
Inner Code $C_{in} : [n, k, d]_q$ (Greedy Construction)

$Q = q^k$

$\Downarrow$

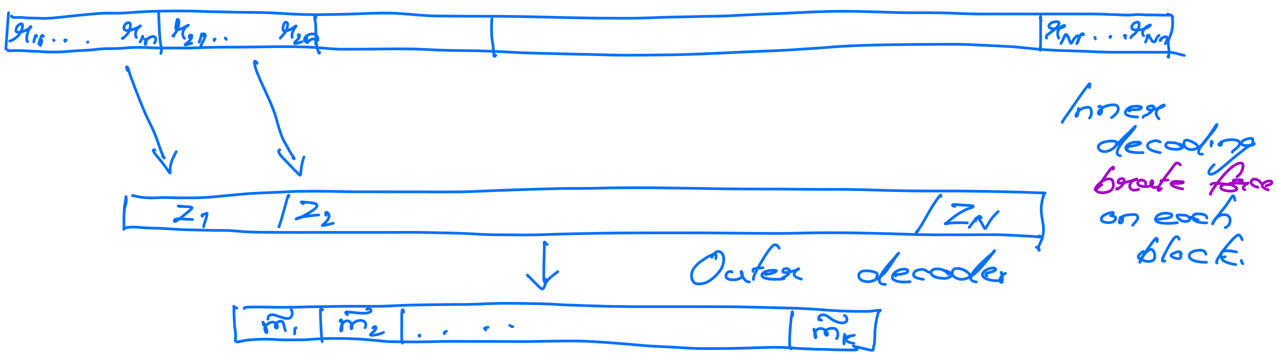
Concatenated $C = C_{out} \circledcirc C_{in}$ $[Nn, Kk, Dd]_q$

Today : Decode Concatenated Codes



Decoding Question: Given $(x_{11}, \dots x_{1n}, x_{21} \dots x_{2n}, \dots, x_{N1} \dots x_{Nn})$ compute $m_1, \dots m_k$?

Vanilla Decoding:



$$z_i = \underset{z}{\operatorname{argmin}} \left\{ \Delta\left(C_{in}(z), (x_{i1}, \dots x_{in})\right) \right\}$$

When can we guarantee $(\tilde{m}_1, \tilde{m}_2 \dots \tilde{m}_k) = (m_1, m_2 \dots m_k)$?

__Claim 1:__ If #errors in i^{th} block $< d/2$.
then $E(z_i) = y_i$ (i.e, $z_i = x_i$).

__Claim 2:__ If #{blocks i s.t. $z_i \neq x_i$} $< D/2$
then $(m_1 \dots m_k) = (\tilde{m}_1, \tilde{m}_2, \dots \tilde{m}_k)$.

__Claim 3:__ If total # of error $< \frac{D}{2} \cdot \frac{d}{2}$ then

$\#\{\text{blocks } i \;\; \geq \frac{d}{2} \text{ errors in that block}\} < \frac{D}{2}$.

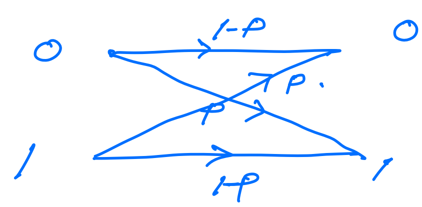
Pf: Otherwise, total # of errors $\geq \frac{D}{2} \cdot \frac{d}{2} = \frac{Dd}{4}$

Thm: Vanilla Concatenated Decoder can correct t errors if $t < \frac{Dd}{4}$.

Observations: * Not the best one can hope for (which is $Dd/2$)

* Despite that, vanilla decoder is suff to get Shannon capacity on BSC.

Recall BSC(p).



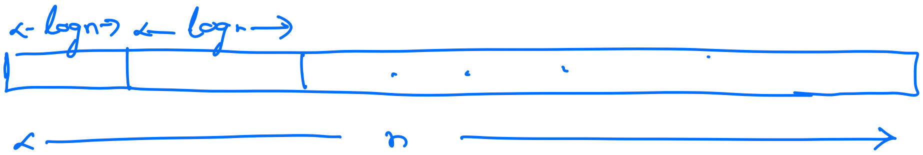
Bits flipped independently.

Goal: Design an explicit code that achieves Shannon capacity on BSC(p)?

Shannon: A random code of rate $R = 1 - H(p) - \delta$ will achieve capacity & furthermore decoding (runs in $\exp(n)$) time has error $2^{-\delta n}$.

Decoding Error – exponentially small

Encoding & Decoding $\Big\}$ exp(n) time.



$n/\log n$ blocks of $\log n$ length each & apply Shannon's Thm on each block.

Encoding & Decoding: $\frac{n}{\log n} \cdot \exp(\log n) = \text{poly}(n)$

Decoding Error: $\Pr[i^{th} \text{ block decoded incorrectly}] = \exp(-\log n)$

$$= \frac{1}{\text{poly}(n)}$$

$$\Pr[\text{error}] = \Pr[\exists i,\ i^{th} \text{ block decoded incorrectly}]$$

$$\leq \frac{n}{\log n} \cdot \frac{1}{\text{poly}(n)} = \frac{1}{\text{poly}(n)}.$$

At the cost of getting poly encoding & decoding, we have increased the decoding error to $\frac{1}{\text{poly}(n)}$.

Question: Can we improve construction to get back inverse exp error?

YES: Using vanilla decoding of concatenated codes.
[Forney].

Forney's Construction:

Outer code: Rate $(1-\varepsilon)$. length N
recover from r-fraction of errors
$(r < \varepsilon/2)$

(Reed-Solomon for instance)

Inner code: Rate $1-H(p)-\varepsilon$, length n

Shannon's (random) code

Composed Code: $R = (1-\varepsilon)(1-H(p)-\varepsilon)$
$\geq 1-H(p)-2\varepsilon$

Block length $= Nn$ $\quad (n = C\log N)$

$\Pr[i^{th}$ block decoded incorrectly$] \leq \exp(-n)$.
[Shannon].

If the # blocks that are decoded incorrectly $< \varepsilon N/2$, then the outer decoding will recover message correctly

$$Pr[\text{error}] = Pr\left[\#\text{ blocks decoded incorrectly} > \frac{\epsilon N}{2}\right]$$

$$\le 2^N \cdot \left(\exp(-n)\right)^{\epsilon N/2} \quad \text{[Chernoff Bound]}$$

$$= \exp(-nN).$$

Decoding Error has reduced to $\frac{1}{\exp(n)}$. $\boxtimes$

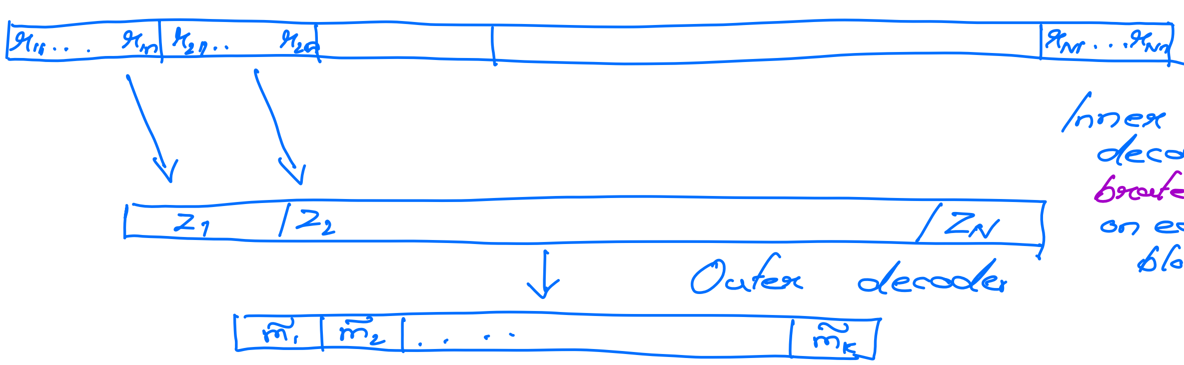
$$= \exp(-\epsilon^c N) \quad (N = \text{block length})$$

Encoding = Decoding $\text{poly}(N) + \exp\left(\frac{1}{\epsilon}\right)$.

Recover from errors $\frac{Dd}{2} > t \geq \frac{Dd}{4}$

Extreme Cases of $\frac{Dd}{2}$ errors

(1) $< D/2$ blocks are changed to different inner codewords by flipping d locations.



Inner decoding brute force on each block.

(Inner decoder fails on blocks.
But # such blocks $< D/2$, outer decoder performs well)

② $< D$ blocks have $d/2$ flips in them.

In this setting, inner decoder could signal the outer decoder that these blocks are not to be trusted.
Outer Decoder erases these blocks & decodes perfectly.

Want: An outer code that can handle both erasures and errors.

$d(C_{out}) = d_{out}$.

- Can handle if # erasures $< d_{out}$
- Can handle if # errors $< d_{out}/2$.

Claim - Can handle s erasures & e errors if $s + 2e < d_{out}$.

Pf: C $[n, k, d]_q$ - code.
$\downarrow$ s erasures

C' $[n-s, k', d-s]_q$ -codes

Can recover C' from e errors if

$e < \frac{d-s}{2}$.

ie, $2e + s < d$. $\boxtimes$

What about algorithmically?

WB decoder can handle this since the alg worked for any set of evaluation points.

Forney's GMD (Generalized Minimum Distance) Decoder

Requirements: C_{out} - Outer Code w/ Decoder Dec_{out} that can handle e errors & s erasures if $2e + s < D_{out}$.

C_{in}: Inner code w/ distance d_{in} & decoder Dec_{in} that can handle f errors if $2f < D_{in}$.

GMD Decoder: $C = C_{out} \circledcirc C_{in}$

that can handle $< D_{out} d_{in}/2$ errors.

GM Decoder:

Input: $((r_{11}, \ldots r_{1n}) (r_{21}, \ldots r_{2n}) \ldots \quad (r_{N1}, \ldots r_{Nn}))$ received word.

Algorithm:

For each $i \in [N]$.

(*) Run Dec_{in} on $(r_{i1}, \ldots r_{in})$ to obtain z_i or $\perp$

(*) $e_i = \min\{\Delta(C_{in}(z_i), (r_i)), d/2\}$

(*) With prob $e_i/(d/2)$ erase z_i.
otherwise retain z_i.

- $z_1, \ldots z_N$ — some of which are erased.
- Run Outer decoder Dec_{out} on $(z_1, \ldots z_N)$ to obtain $\tilde{m}_1, \tilde{m}_2, \ldots \tilde{m}_k$. $\boxtimes$