6.S708 Lecture 7 — 2/27/08

Note Title — 2/25/2008

Today: Binary (Algebraic) Codes
- Concatenation [Forney]
- Justesen's Codes
- BCH Codes

Review

Last time: Saw

1. Wozencraft: $x \mapsto \langle x, \alpha x \rangle$
2. Reed-Solomon: $m(\cdot) \longmapsto \langle m(\alpha_1) \ldots m(\alpha_n) \rangle$
3. Multivariate Polynomial Codes (Reed Muller)
4. Hadamard Codes

Will use 1 & 2 today.

CONCATENATION OF CODES [FORNEY]

- A naive idea (to get binary codes):
  - Start with Reed Solomon code over $\mathbb{F}_{2^t}$ $\quad t = \log n$
  - Represent $\mathbb{F}_{2^t}$ as t bits
  - Say RS code was $[n, \frac{n}{2}, \frac{n}{2}]_n$.
    Then we get $[n \log n, \frac{n}{2} \log n, \frac{n}{2}]_2$ code by this process.
  - Rate is still good; Distance suffers because $\mathbb{F}_{2^t}$ represented as t bit string. Poor Redundancy in this rep'n.

Better Idea: Represent $\mathbb{F}_{2^t}$ nicely, using "Error-Correcting Code"

- Say we "know" good code

$$C_{inner} : \{0,1\}^t \to \{0,1\}^{2t}$$

Say $(2t, t, .01t)_2$ code.

- Using C_{inner} to represent elements of $\mathbb{F}_{2^t}$ & "combining" with RS gives

$$\left(2tn, \frac{tn}{2}, \frac{.01tn}{2}\right)_2 \text{ code}$$

$R, \delta > 0$!

CONCATENATED CODES [FORNEY '66]

- Combination technique called "Concatenation"
- Can concatenate

$$(n_1, k_1, d_1)_{2^{k_2}} \circ (n_2, k_2, d_2)_2$$

code to get $(n_1 n_2, k_1 k_2, d_1 d_2)_2$ code.

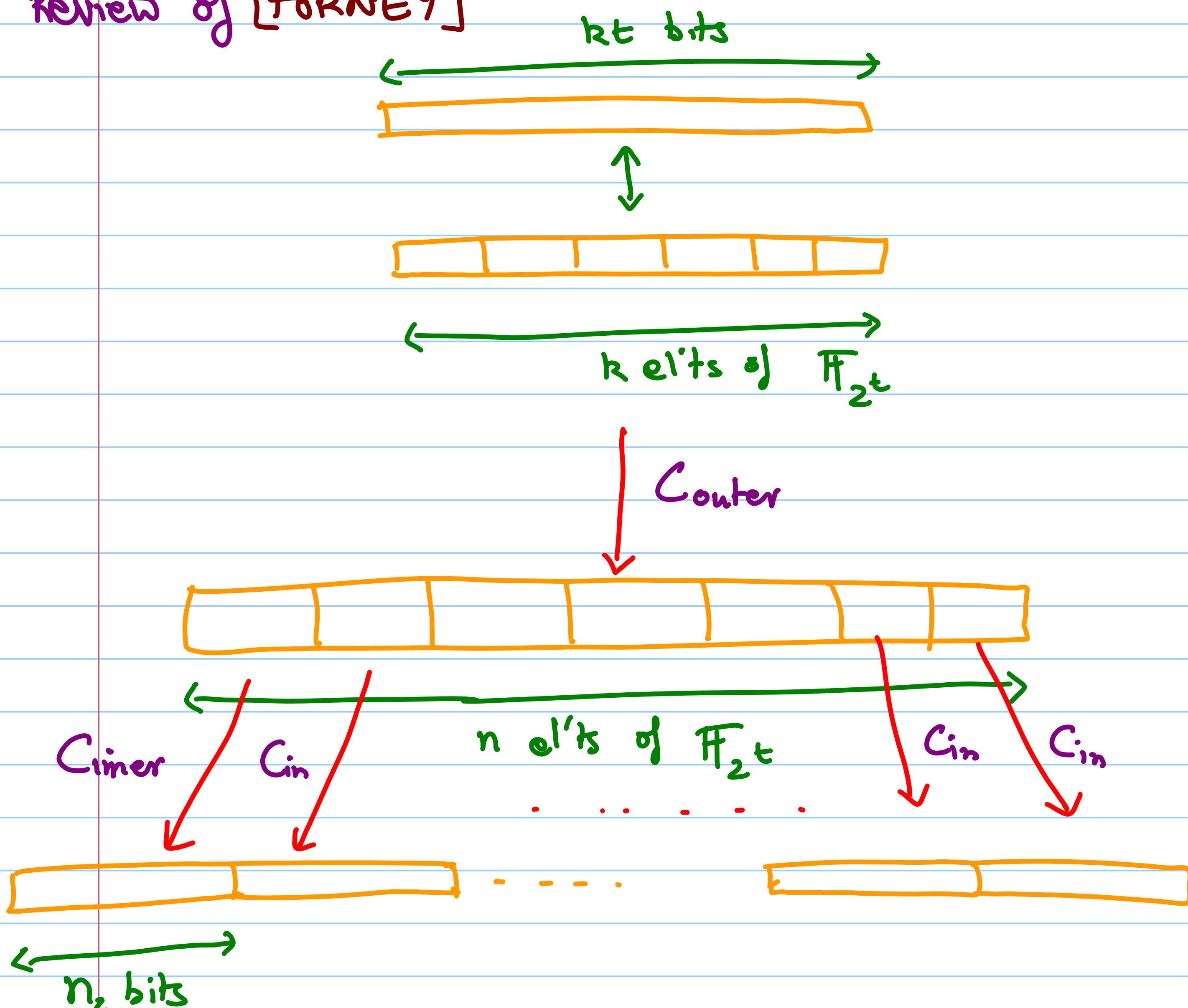
- Code over big alphabet : Outer code
  Small code over small ↓ : Inner code
- Outer alphabet ≡ Inner message space
- Both Outer, Inner linear & using
  $\mathbb{F}_{2^{k_2}} \longleftrightarrow \mathbb{F}_2^{k_2}$ correspondence yield
  linear codes.

Does This Give Explicit Codes?

- How do you find Outer code? Easy because of larger alphabet (use RS)
- How do you find Inner code?
  - This code is smaller, can try recursion, but hasn't worked ... so far.
  - [FORNEY] Use VARSHAMOV search! Takes time $\text{poly}(2^{k_2}) = \text{poly}(n)$
- Conclusion 1: YES - this gives explicit codes .... Encoding can be done in polynomial time.
- Conclusion 2: NO - this is still "search" ......
  [Only formalized recently .... e.g. should be able to compute $(i,j)^{th}$ entry of generator in time $\text{poly}(\log n)$.]

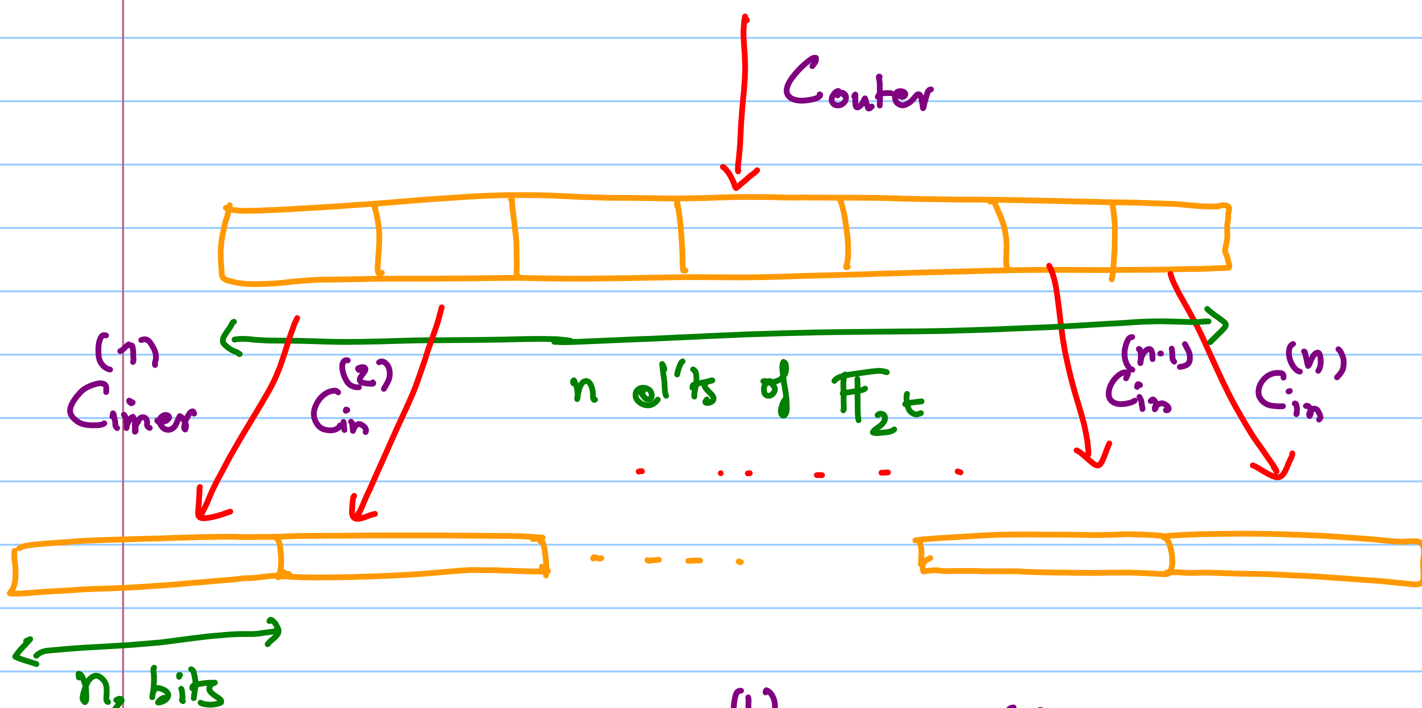
Justesen's Idea

Review of [Forney]



- Search problematic, since we need good C_{inner} so that we use it repeatedly
- But why should we use same C_{inner}? Why not "try" out many different ones, in same code?

(So replace last step of FORNEY with ...



[Using different $C_{in}^{(1)}$ .... $C_{in}^{(n)}$ ...]

- Construction certainly works if every code in $\{ C_{in}^{(1)} \dots C_{in}^{(n)} \}$ good

- But even works if "most" codes are good! As in WOZENCRAFT's ENSEMBLE

- JUSTESEN = REED-SOLOMON $\circ$ {WOZENCRAFT}

EXPLICITLY : Fix integer t

- Compute $\mathbb{F}_{2^t}$.

- Encode : $m_0, \dots m_{k-1} \in \mathbb{F}_{2^t}$

Let $M(x) \triangleq \sum m_i x^i$; $\langle M(\alpha), \alpha \cdot M(\alpha) \rangle_{\alpha \in \mathbb{F}^*}$

- Exercise : Verify this is "explicit".

CONCLUSION

- FORNEY gives "explicit" (polytime encodeable) codes with $R, \delta > 0$
- JUSTESEN gives even more "explicit" (entries of generator polylog n-time computable) codes with same $R, \delta > 0$
- Codes do not match GV bound!

$$\underbrace{R = \max_{\delta \le \delta_2 \le \frac{1}{2}} \left\{ \underbrace{(1 - H(\delta_2))}_{\text{inner rate}} \cdot \underbrace{\left(1 - \frac{\delta}{\delta_2}\right)}_{\text{RS rate}} \right\}}_{\text{ZYABLOV bound.}}$$

Next: BCH Codes

· Co-invented by [Bose, Chaudhuri] ~ '59
[Hocquenghem]

Motivation: - Say want long binary code to correct 5 bit flip errors.

- How do n, k relate?

[Hamming bound]: $2^k \cdot \binom{n}{5} \le 2^n$

$$\Rightarrow k + (5 - o(1)) \cdot \log n \le n$$

Construction from Reed-Solomon codes

- Let $n = 2^t$
- Let $k = n - 10$
- Let $C_{outer} = [n, n-10, 11]_n$ RS code
- Represent n-ary elements as t-bit strings (Concatenate with $[t, t, 1]_2$ code!)

  and get $C = [nt, (n-10)t, 11]_2$ code

$$= [n', n' - 10 \log n, 11]_2$$

- $n' - k' \approx 10 \log n$ vs. 5 in Hamming bound.

Which is right?

BCH Codes :

- Achieve $n-k = 5\log n$
- Generally $[n, n-e\log n, 2e+1]_2$ codes.
- Asymptotically $\frac{k}{n} \to 0$ or $\frac{d}{n} \to 0$
- But have half the redundancy of random codes, as long as $d = n^{\epsilon}$.

BCH Construction

- Let $n = 2^t$; d = desired distance
- Let C be $[n, n-d+1, d]_n$ code with parity check matrix

$$\begin{bmatrix} 1 & 1 & \cdots & 1 & \cdots & 1 \\ \alpha_1 & & & \alpha_i & & \alpha_n \\ \alpha_1^2 & & \cdots & \vdots & \cdots & \vdots \\ \vdots & & & & & \\ \alpha_1^{d-2} & & \cdots & \alpha_i^{d-2} & \cdots & \alpha_n^{d-2} \end{bmatrix}$$

(Verifying: # rows = $d-1$ $\Rightarrow$ $k = n-d+1$

Every $(d-1) \times (d-1)$ submatrix Vandermonde & so non-singular $\Rightarrow$ no wt. d code words).

- $C_{BCH} = C \cap \{0,1\}^n$
- Easy to see: $n = n$ [length same]

  $d \geq d$ [distance does not decrease]
- Non trivial: $k = ?$

Weak Analysis

Recall
$$\begin{array}{ccc} \mathbb{F}_{2^t} & \longleftrightarrow & \mathbb{F}_2^t \\ \rotatebox{90}{$\in$} & & \rotatebox{90}{$\in$} \\ \alpha & \longleftrightarrow & V_\alpha \\ 1 & \longleftrightarrow & (1,0,0,0,\dots 0) \end{array}$$

Using this rep'n C_{BCH} has parity check

$$H' = \begin{bmatrix} 1 & \cdots & 1 \\ V_{\alpha_1} & \cdots & V_{\alpha_n} \\ \vdots & & \\ V_{\alpha_1^{d-2}} & \cdots & V_{\alpha_n^{d-2}} \end{bmatrix}$$

Has $(d-2)t+1$ rows

Still off by factor 2.

$$\Rightarrow \quad n-k = (d-2)t+1 \approx d \log n$$

Stronger Analysis

- KEY INSIGHT : $(\alpha_1 + \cdots \alpha_\ell)^2 = \alpha_1^2 + \cdots \alpha_\ell^2$

  Since $\mathbb{F} = \mathbb{F}_{2^t}$

  (In general over $\mathbb{F}_{p^t}$

  $$(x+y)^p = x^p + y^p \quad)$$

- Let

$$\tilde{H} = \begin{bmatrix} 1 & & & & & & 1 \\ V_{\alpha_1} & & & & & & V_{\alpha_n} \\ V_{\alpha_1^3} & \cdot & \cdot & \cdot & \cdot & \cdot & V_{\alpha_n^3} \\ \vdots & & & & & & V_{\alpha_n^5} \\ & & & & & & \vdots \\ V_{\alpha_1^{d-2}} & & & & & & V_{\alpha_n^{d-2}} \end{bmatrix}$$

(throw away even power rows.)

- Claim: Code with parity check $\tilde{H}$ same as code with parity check H'.
- (Modulo Proof of Claim:)

  # rows $= \left\lceil \frac{d-2}{2} \right\rceil t + 1 = n-k$

  $\approx \frac{d}{2} t$

  matches Hamming!

- <u>Proof of Claim:</u>

  Suppose $(x_1 \ldots x_n) \cdot \tilde{H} = 0$ $\quad x_i \in \{0,1\}$

  $\Rightarrow \forall$ odd $j \in \{1, \ldots, d-2\}$

  $$\sum_{i=1}^{n} x_i v_{\alpha_i}^{j} = 0$$

$\Rightarrow$ Using $\mathbb{F}_{2^t} \leftrightarrow \mathbb{F}_2^t$ correspondence

$$\sum x_i \alpha_i^j = 0$$

Now consider $2^a \cdot j$ $\quad j \in [1 \ldots d-2]$ odd

$$\sum x_i \alpha_i^{2^a \cdot j} = \left(\sum x_i \alpha_i^j\right)^{2^a}$$

$$= 0$$

$$\Rightarrow \sum x_i V_{\alpha_i^{2^a \cdot j}} = 0$$

$$\Rightarrow (x_1 \ldots x_n) \cdot H' = 0 \qquad \boxtimes$$

Conclusions:

- Algebra can give surprisingly strong results.

- Elementary techniques.

  1. finite fields exist
  2. Deg d poly has $\leq d$ roots.
  3. $(X+y)^p = x^p + y^p$