Today + Next Lecture

Computational Perspectives on Source / Channel Coding.

Today: The Problem Of Channel Coding.

Ex. BSC (p).

\[ \text{Given } m \in \{0,1\}^k \]

\[ E : m \mapsto x = E(m) \in \{0,1\}^n \]

\[ \text{Given } y \in \{0,1\}^n \]

\[ D : y \mapsto \hat{m} \in \{0,1\}^k \]
Shannon: Fix $R = (1-H(p) - \epsilon)$

As $n \to \infty$, $\exists E, D$ s.t.

\[ \Pr [D(y) \neq m] \to 0 \]

Problem: $E, D$ exist.

Are they efficient to compute?

E.g., if $R = 100$ bits

\[ n = 2^{100} \text{ bits} \]

Running time of $D$ may be $2^{100}$!

\[ \Rightarrow \] \[ \Pr [D(y) \neq m] \to 0 \]

Can we do better?
Goal: E, D must run in \textit{time poly}(n).

Achieved in 1966 by [Forney]. How?

\textbf{Encoding}

\texttt{REED-SOLOMON}

\texttt{Concatenation}

\textbf{Decoding}

\texttt{Brute force}

\texttt{Algebra}
Reed-Solomon Codes

Imagine we are trying to correct errors on a \( q \)-ary channel; \( q = \text{prime} \).

Idea: view alphabet \( \Sigma = \{ 0, \ldots, q-1 \} \) as a "field" of addition/multiplication modulo \( q \).

Polynomials over field \( F \)

\[ p \in F[x] : \langle c_0, \ldots, c_{k-1} \rangle \]

Evaluation: \( p \) at \( x = \xi \) is \( \sum c_i \xi^i \)

Nice properties

Degree \( \leq \) polynomial \( p(x) \) has at most \( \leq \) roots in \( F \).
**RS encoding**

Message = $k$ elements of $F$

$\mathbb{F}_n = n$ elements of $F$

How? Fix $\alpha_1, \ldots, \alpha_n$ distinct in $F$

$$c_0, \ldots, c_{k-1} \rightarrow p \in \mathbb{F}_n[x] \rightarrow \langle p(\alpha_1), \ldots, p(\alpha_n) \rangle$$

**Properties**

$$c_0, \ldots, c_{k-1} \rightarrow p \Rightarrow (p - p') = 0$$

$$c_0, \ldots, c_{k-1} \rightarrow p'$$

$$(p - p')(\alpha_i) = 0 \text{ for at most } k-1 \text{ choices } i$$

$\Rightarrow$ differ in at least $n-(k-1)$ places.
Immediate Consequences:

- Can correct up to $\eta-(k-1)$ erasures in poly time. How?

$$p(a_1) \cdots p(a_n)$$

$$\downarrow$$

$$\langle ? \ p(a_2) \ y(a_3) \ ? \ ? \ p(a_n) \rangle$$

To find $C_0 \ldots C_m$, solve linear system

$$\begin{bmatrix}
1 & d_2 & d_2^2 & \cdots & d_2^{k-1} \\
1 & d_3 & d_3^2 & \cdots & d_3^{k-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & d_n & d_n^2 & \cdots & d_n^{k-1}
\end{bmatrix}
\begin{bmatrix}
C_0 \\
C_1 \\
\vdots \\
C_m
\end{bmatrix}
= 
\begin{bmatrix}
p(a_2) \\
p(a_3) \\
\vdots \\
p(a_n)
\end{bmatrix}$$
Solving errors?

- Can correct $\frac{n-(k-1)-1}{2}$ errors in polytime [1960 - Peterson]

What does this have to do with the BSC?

Forney

let $q = 2^l$ (field element is a bit).

Take $k$ bit message

$= \frac{k}{l}$ element of $F$

Encode as polynomial of degree $(1+\epsilon) \frac{k}{l}$.
yield \((1+\epsilon)^{\frac{1}{\epsilon}}\) elements of \(\mathbb{F}\) 

But now encode \(\mathbb{F}\) element as (say) \(2^L\) bits by a good code from \(\mathbb{F}^L\) to \(\mathbb{Z}^{2^L}\).

- Distinct messages differ in at least \(\epsilon \cdot L\) and \(\frac{10}{L}\).
- To decode

Yield \(\cdots\) polytime enc. \(\cdots\).

Also: No solution to \(2^{100}\) problem.