

# ST05 LECTURE 25

Note Title

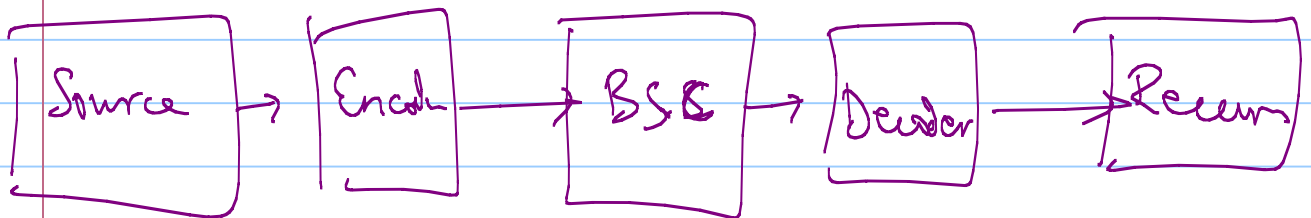
5/16/2006

TODAY + NEXT LECTURE

- COMPUTATIONAL PERSPECTIVES ON SOURCE / CHANNEL CODING.

TODAY = THE PROBLEM OF CHANNEL CODING.

Ex. BSC (p).



Given  $m \in \{0,1\}^k$

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

Given  $y \in \{0,1\}^n$

$D: y \mapsto \hat{m} \in \{0,1\}^k$

Shannon: Fix  $R = (1 - H(p) - \epsilon)$

As  $n \rightarrow \infty$ ,  $\exists E, D$  st.

$$\Pr [D(y) \neq m] \rightarrow 0$$

$m \rightarrow x \rightarrow y \rightarrow \hat{m}$

Problem:  $E, D$  exist.

Are they efficient to compute

e.g. if  $R = 100$  bits

$n = 200$  bits

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

..  $E$  may be  $2^{100}$ !

Can we do better?

Goal: E, D must run in time  $\text{poly}(n)$ .

Achieved in 1966 by [Forney]. How?

Encoding

REED - SOLOMON

+

CONCATENATION

Decoding

Brute force

+

Algebra.

## REED-SOLOMON Codes

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

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

## Polynomials over field $\mathbb{F}$

$$p \in \mathbb{F}[x] : \langle c_0 \dots c_{k-1} \rangle$$

$$= \sum c_i x^i$$

Evaluation:  $p$  at  $\alpha = \sum c_i \alpha^i$  ( $\text{Eva}_\alpha : \mathbb{F} \rightarrow \mathbb{F}$ )

## Nice properties

degree  $l$  polynomial  $p(x)$  has  
at most  $l$  roots in  $\mathbb{F}$ .

## RS encoding

Message =  $k$  elements of  $\mathbb{F}$

En  $n$  elements of  $\mathbb{F}$

How? Fix  $\alpha_1, \dots, \alpha_n$  distinct in  $\mathbb{F}$

$$C_0, \dots, C_{k-1} \rightarrow p \in \mathbb{F}[x] \rightarrow \langle p(\alpha_1), \dots, p(\alpha_n) \rangle$$

## Properties

$$C_0, \dots, C_{k-1} \rightarrow p \quad \Rightarrow \quad (p-p') \neq 0$$

$\# \qquad \qquad \qquad \#$

$$C'_0, \dots, C'_{k-1} \rightarrow p' \quad \Downarrow$$

$$(p-p')(\alpha_i) = 0 \quad \text{for at most}$$

$k-1$  choices of  $i$

$\Rightarrow$  differ in at least  $n-(k-1)$  places!

## Immediate Consequences:

- Can correct upto  $n - (k-1)$  erasures in poly time. How?

$$p(\alpha_1) \dots p(\alpha_n)$$

↓

$$\langle ? \quad p(\alpha_2) \quad p(\alpha_3) \quad ? \quad ? \quad p(\alpha_6) \dots p(\alpha_n) \rangle$$

To find  $C_0 \dots C_{k-1}$  solve lin system

$$\begin{bmatrix} 1 & \alpha_2 & \alpha_2^2 & \dots & \alpha_2^{k-1} \\ 1 & \alpha_3 & \alpha_3^2 & \dots & \alpha_3^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_n & \alpha_n^2 & \dots & \alpha_n^{k-1} \end{bmatrix} \begin{bmatrix} C_0 \\ \vdots \\ C_{k-1} \end{bmatrix} = \begin{bmatrix} p(\alpha_2) \\ p(\alpha_3) \\ \vdots \\ p(\alpha_n) \end{bmatrix}$$

## Solving errors ?

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

What does this have to do with the  
BSC ?

## Forney

let  $q \approx 2^l$  (field element is  $l$  bits).

take  $k$  bit message

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

encode as polynomial of degree  $(1+G) \frac{k}{l}$ .

yield  $(1 + \epsilon) \frac{R}{l}$  elements of  $\mathbb{F}$

But now encode  $\mathbb{F}$  elements as (say)  
 $2l$  bits by a good <sup>(Shannon)</sup> code from  
 $\{0, 1\}^l \rightarrow \{0, 1\}^{2l}$ .

- Distinct messages differ in at least

$$\epsilon \cdot \frac{l}{10} \text{ bits.}$$

- To decode

Yields --- polytime enc. + dec. ---

Also: No solution to  $2^{100}$  problem.