Lecture 6

Note Title 1/11/2008

d = min # of linearly dependent columns of H.

Ex: 1)

$$H = \begin{array}{c} \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \\ \begin{array}{cccccc} & & \times & & \times & \times \end{array} \end{array}$$

$d = 1$? $\times$ $d \geq 2$

$d = 2$? $\times$ $d \geq 3$

$d = 3$ ✓

2)

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

$d = 1$? $\times$

$d = 2$? $\times$

$\boxed{d = 3\ ?}$ more difficult to eliminate

Design of H for $d=3$: → Given n, minimize r

$r = n-k \times n$ → Each col of $H \in \{0,1\}^r$

Ex: 1) $n = 6$, $r = 3$

$$H = \begin{bmatrix} - & - & - & - & - & - \end{bmatrix} \qquad r = \lceil \log_2 n \rceil$$

2) $n = 13$, $r = 4$

→ Given r, maximize n

Ex: 1) $r = 1$, $n = 1$ $\qquad H = [1]$

2) $r = 2$, $n = 3$ $\qquad H = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ $\quad C = \{000, 111\}$

$\underline{k} \quad \underline{m} \quad (3, 1, 3)$

3) $r=3$, $n=7$ $\quad H=\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$

$(7, 4, 3)$ Code

$\longrightarrow$ Hamming Code (Binary) $\quad n = 2^r - 1$

$k = 2^r - 1 - r$

$d = 3$

r	n	k	k/n	d	d/n
3	7	4	4/7	3	3/7
4	15	11	11/15	3	3/15
5	31	26	26/31	3	3/31
$\longrightarrow$ 6	63	57	57/63	3	3/63

Syndrome decoding for (7,4,3) Hamming Code

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$\underline{r} \xRightarrow{H\underline{r}^T} \underline{s} \longrightarrow 3$ bits

$\underline{s}$	$\hat{\underline{e}}$
000	0000000
001	1000000
⋮	⋮
101	0000100
⋮	⋮
111	0000001

16 codewords

# of vectors in each non-overlapping sphere = 8

$16 \times 8 = 128$ ✓

$\longrightarrow$ Perfect codes

Design for $d=4$

→ Idea is from modification of existing codes

1) Extending a code:

(n, k, d) Code C

New code $C^{(e)} = \{[x_0 x_1 \dots x_{n-1}\; x_n] :$

$[x_0 x_1 \dots x_{n-1}] \in C,$

$x_n = x_0 + x_1 + \dots + x_{n-1}\}$

$$(n, k, d) \longrightarrow (n+1, k, d+1), \quad d: \text{odd}$$

$$d, \quad d: \text{even}$$

C : G, $H \rightarrow n-k \times n$

$C^{(e)}$: $G^{(e)}$, $H^{(e)} \rightarrow n-k+1 \times n+1$

$G^{(e)}$: $k \times n+1$

$$H^{(e)} = \left[\begin{array}{ccccc|c} & & H & & & \begin{matrix} 0 \\ 0 \\ \vdots \\ 0 \end{matrix} \\ \hline 1 & 1 & \dots & \dots & 1 & 1 \end{array}\right]$$

$G^{(e)} = [\, G \mid \;] \rightarrow$ parity of row of G

→ puncturing $\quad (n, k, d)$ code C

$$\underline{c} = [\underbrace{\underline{m}}_{\substack{k \\ \text{message} \\ \text{bits}}} \quad \underbrace{\underline{p}}_{\substack{n-k \\ \text{parity} \\ \text{bits}}}]$$

→ Drop parity bits. #punctured $= p < n-k$
"puncture"

$C^{(p)}$: Punctured code
$(n-p, k, \leq d)$

C : $G = [I \mid \overbrace{P}^{p \text{ bits}}]$

$C^{(p)}$: $\underbrace{G^{(p)}}_{k \times n-p} = [I \mid \quad] X$

$H^{(p)}$:
$n-p-k \times n-p$