(12)

Hamming Code Parity Checks :

$$s_1 = b_1 + b_2 + b_3 + P_1$$

$$s_2 = b_1 + b_3 + b_4 + P_2$$

$$s_3 = b_2 + b_3 + b_4 + P_3$$

MOD - 2 Sum

What happens if a codeword $c = \{c_1, c_2, \ldots, c_7\}$

is received with one bit in error ?

→ Depends on which bit is in error.

→ Suppose $c_7 = P_3$ is in error

$s_1 = 0$, $s_2 = 0$, but $s_3 = \underline{1}$

→ Suppose $c_1 = b_1$ is in error

$s_1 = 1$, $s_2 = 1$, but $s_3 = 0$

→ Hamming Code Error Tabulation

Consider all possible error cases with 1 or 0 bit in error. Let $\overline{e} = \overline{c} - \overline{r}$ denote the error vector.

$\overline{S} = \{s_1, s_2, s_3\}$ called syndrome vector

| $e_1$ | $e_2$ | $e_3$ | $e_4$ | $e_5$ | $e_6$ | $e_7$ | $s_1$ | $s_2$ | $s_3$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

Decoding strategy: (1) receive $\overline{r}$

Notes: syndrome for $\overline{r}$ and $\overline{e}$ $\xrightarrow{\text{are the same.}}$ (2) Compute syndrome using $\overline{r}$.

: Requires to store above table. → (3) Look up the corresponding row.

: Assumes that only one bit has occurred. → (4) Flip the bit at which error occurs in that row.

(14) Definitions: Hamming Distance

$d_H(\bar{x}, \bar{y})$ = Number of elements for which $x_i \neq y_i$

Minimum Distance :

$$d_{min} = \min_{\bar{c}_1, \bar{c}_2 \in \mathcal{C}} d_H(\bar{c}_1, \bar{c}_2)$$

set of all codewords

Minimum Distance Decoder :

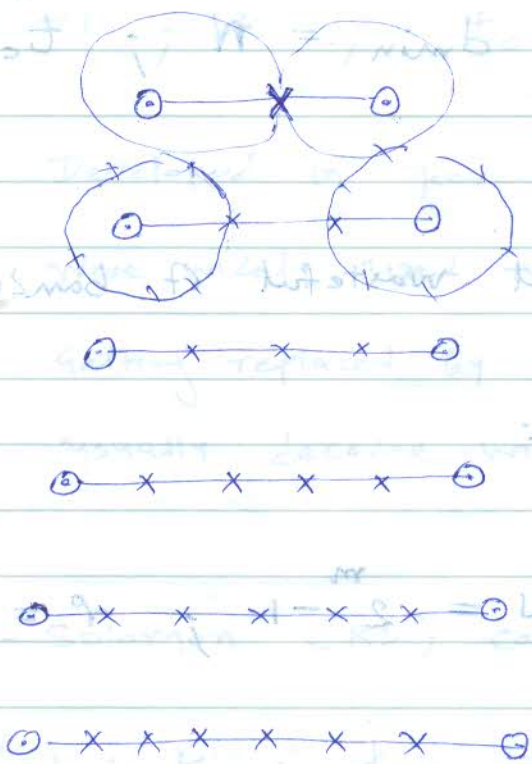
Given the received vector $\bar{r}$, output the codeword $\bar{c}$ such that $d_H(\bar{r}, \bar{c})$ is as small as possible. If there are multiple such $\bar{c}$, pick any one of them.

$t_d$ : Number of detectable (bit) errors by the code

$t_c$ : Number of correctable (bit) errors by the code.

15

## Code Geometry :



| $d_{min}$ | $t_d = d_{min} - 1$ | $t_c = \left\lfloor \frac{d_{min}-1}{2} \right\rfloor$ |
|---|---|---|
| 2 | 1 | 0 |
| 3 | 2 | 1 |
| 4 | 3 | 1 |
| 5 | 4 | 2 |
| 6 | 5 | 2 |
| 7 | 6 | 3 |

## DESIGN PARAMETERS :

Block CODES (take a block of K bits at a time and convert them to a block of N coded bits)

$K$ = # of Information Bits

$N$ = # of encoded bits (size of codeword)

$r = K/N$ = rate of encoder ; $0 \leq r < 1$

$d_{min}$ = Min. Hamming Distance

$t_c = \left\lfloor \frac{d_{min}-1}{2} \right\rfloor$ = # of correctable bits.

Example Product code : $N = 9$; $K = 4$

| | | |
|---|---|---|
| $c_1 = m_1$ | $c_2 = m_2$ | $c_3 = m_1 \oplus m_2$ |
| $c_4 = m_3$ | $c_5 = m_4$ | $c_6 = m_3 \oplus m_4$ |
| $c_7 = m_1 \oplus m_3$ | $c_8 = m_2 \oplus m_4$ | $c_9 = c_3 \oplus c_6$ |

| | | |
|---|---|---|
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 0 | 1 | 1 |

Example Decoding with five of 9 erasure bits

| | | |
|---|---|---|
| x | x | 1 |
| 1 | x | 0 |
| 0 | x | x |

$\Rightarrow$

| | | |
|---|---|---|
| x | x | 1 |
| 1 | 1 | 0 |
| 0 | x | x |

row by row

$\Rightarrow$

| | | |
|---|---|---|
| 1 | x | 1 |
| 1 | 1 | 0 |
| 0 | x | 1 |

column by column

$\Rightarrow$

| | | |
|---|---|---|
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 0 | 1 | 1 |

row decoding

Iterative row $\rightarrow$ column $\rightarrow$ row ... decoding.

Decoding of product codes :

→ Can get stuck

$$\boxed{\begin{matrix} \cancel{0} & \cancel{0} & 1 \\ \cancel{0} & \cancel{0} & 0 \\ 0 & 1 & 1 \end{matrix}}$$

← Not possible to perform decoding

For The above case,

$e = 4/9$, $r = 4/9$, $c = 1 - e = 1 - \frac{4}{9}$

$= 5/9 > r$

Thus, This case should have been correctable but The product code fails here.

→ This product code is characterized by $(n - k = 9 - 4 = 5)$ a set of five linearly-independent equations.

$c_9 = c_3 \oplus c_6 \Rightarrow c_3 \oplus c_6 \oplus c_9 = 0$

$c_3 = c_1 \oplus c_2 \Rightarrow c_1 \oplus c_2 \oplus c_3 = 0$

$c_6 = c_4 \oplus c_5 \Rightarrow c_4 \oplus c_5 \oplus c_6 = 0$

$c_7 = c_1 \oplus c_4 \Rightarrow c_1 \oplus c_4 \oplus c_7 = 0$

$c_8 = c_2 \oplus c_5 \Rightarrow c_2 \oplus c_5 \oplus c_8 = 0$

→ In general, it takes $(n-k)$ linearly indep. equations to specify $(n, k)$ linear code.

These five equations can be written as the following matrix product :

$$H_{(n-k)\times n} \cdot C_{n\times 1} = \bar{O}_{N\times 1}$$

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \\ c_9 \end{bmatrix} = \bar{O}_{N\times 1}$$

Parity check matrix $H$

This parity-check matrix can be represented by a Tanner Graph, which is a Bipartite Graph.

→ Two types of nodes : $N-K$ check nodes, and $N$ codeword bit nodes

→ Nodes of one type are not connected to the same type of nodes (bi-partition of nodes)