Class 26

Pick any collection of k linearly ind vectors from $\mathbb{F}_2^n$ $\{g_1, \ldots, g_k\}$ & put them as rows of a matrix

$$G = \underset{k \times n}{\begin{bmatrix} \underline{g_1} \\ \vdots \\ \underline{g_k} \end{bmatrix}}$$

Rowspace (G) = Span (Rows of G) : k-dimensional subspace of $\mathbb{F}_2^n$

↓

valid linear code

'Dimension' of linear code $\mathcal{C}$ $\overset{\Delta}{=}$ Dim of subspace $\mathcal{C} \subseteq \mathbb{F}_2^n$

$\dim(\mathcal{C})$ $= k$.

$|\mathcal{C}| = 2^k$.

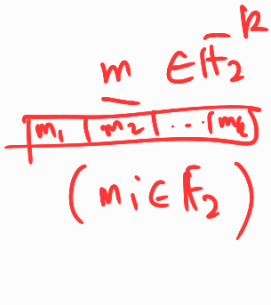
$R_{\mathcal{C}} = k/n = \dfrac{\dim(\mathcal{C})}{n}$

$d_{min}(\mathcal{C}) = \min\limits_{\substack{\underline{c} \neq 0 \\ \underline{c} \in \mathcal{C}}} w_H(\underline{c})$

Encoding: → Operation of mapping $2^{nR=k}$ length msgs to the n-length codewords in a unique manner

→ - Mapping from k-length vectors over $\mathbb{F}_2$ to $\mathcal{C} \subseteq \mathbb{F}_2^n$.

→ For linear codes, we can do this encoding as a linear mapping



$$(m_1, \dots, m_k) \begin{bmatrix} g_1 \\ \vdots \\ g_k \end{bmatrix} = \sum_{i=1}^{k} m_i g_i \in \mathcal{C}$$

↑ rowspace(G_1)

→ Encoding operation requires

Storage + Computation polynomial (in n) unlike non-linear codes which in general require exp complexity.

Examples:

Encoding: $m \in \mathbb{F}_2 \rightarrow$ [encoder] $\rightarrow mG_1 = (m, \dots, m)$

Repetition Code:

$$G_1 = [1\ 1\ 1 \dots 1]_{1\times n}$$

$$\mathcal{C} = \text{Rowspace } G_1 = \{(0 \dots 0), (1, \dots, 1)\}$$

$\dim(\mathcal{C}) = n$, $\dim(\mathcal{C}) = 1 = k$, $R = \frac{k}{n} = \frac{1}{n}$.

Decoding: How to implement min. distance decoder more efficiently?

$$\hat{c} = \underset{c \in \mathcal{C}}{\operatorname{argmin}}\, d_H(y, c)$$

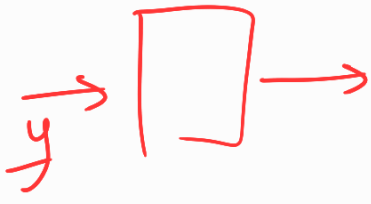
For $n=5$:

Suppose $y = (11100)$ → Then min dist decoder (MDD) output is $\hat{c} = (11111)$

MDD (y) = o/p of MDD given x vector y as input

$$= \begin{cases} \underline{0} = (0,\ldots,0), & \text{if } w_H(y) < \frac{n}{2} \\ \underline{1} = (1,\ldots,1), & \text{if } w_H(y) > \frac{n}{2} \\ \text{pick } \underline{0} \text{ or } \underline{1} & \text{if } w_H(y) = \frac{n}{2} \end{cases}$$

Simple Decoding Rule [Majority Logic Decoder]:



Hamming code: This is a class of codes (Binary Hamming Codes)

$n = 2^r - 1$

$k = 2^r - 1 - r$

$d = 3$

for each $r \geq 3$

↓

Pg of linear code

We take up a particular example: (a particular smallest code in the class of Hamming codes)

For $r = 3$

$$n = 7,\ k = 4,\ d = 3.$$

$$G = \underset{4\times 7}{\left(\; I_4 \;\middle|\; \begin{matrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{matrix} \right)}$$

4×7

↑ appending I_4 with 3 columns to the right.

Note: The 4 rows of G are linearly independent vectors of $\mathbb{F}_2^7$ ($\text{Rank}(G) =$ no of linearly ind vectors in rows (or cols) $= 4 =$ no of lin ind rows)

$\mathcal{C} = \text{rowspace}(G)$ is a 4-dim linear code.

Rate $= 4/7$, $|\mathcal{C}| = 2^k = 2^4 = 16$

$d_{min}(\mathcal{C}) =$ min wt of non-zero codewords $= 3$ (please verify)

Next class: Decoding of Hamming Codes. (this code can correct single errors $t = 1 = \left\lfloor \frac{d_{min}-1}{2} \right\rfloor$)