E2 205: Error-Control Coding
Chapter 4: Linear Codes

Navin Kashyap

Indian Institute of Science

Definitions and Notation

Notation: Henceforth, $\mathbb{F}$ or $\mathbb{F}_q$ will denote a finite field with q elements. (Alternative notation: $\mathrm{GF}(q)$)

Definition: A linear code $\mathcal{C}$ over $\mathbb{F}$ is a subspace of $\mathbb{F}^n$.

- n is the length or blocklength of the code $\mathcal{C}$.
- The dimension of $\mathcal{C}$ is its dimension as a vector space over $\mathbb{F}$; denoted by $\dim(\mathcal{C})$ or $\dim_{\mathbb{F}}(\mathcal{C})$.

Definitions and Notation

Notation: Henceforth, $\mathbb{F}$ or $\mathbb{F}_q$ will denote a finite field with q elements. (Alternative notation: $\mathrm{GF}(q)$)

Definition: A linear code $\mathcal{C}$ over $\mathbb{F}$ is a subspace of $\mathbb{F}^n$.

- n is the length or blocklength of the code $\mathcal{C}$.
- The dimension of $\mathcal{C}$ is its dimension as a vector space over $\mathbb{F}$; denoted by $\dim(\mathcal{C})$ or $\dim_{\mathbb{F}}(\mathcal{C})$.

Notation:

- An $[n, k]$ linear code over $\mathbb{F}$ is a code of blocklength n and dimension k.
- An $[n, k]_q$ linear code is a code of blocklength n and dimension k over $\mathbb{F}_q$.

(Contrast this with the (n, M) notation for block codes.)

Number of Codewords

Proposition: An $[n, k]$ linear code $\mathcal{C}$ over $\mathbb{F}_q$ has q^k codewords.

Proof: Let $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k$ be a basis of the subspace $\mathcal{C}$.

- By Proposition B1, every codeword (i.e., vector) $\mathbf{c} \in \mathcal{C}$ can be uniquely expressed as a linear combination

$$\mathbf{c} = \alpha_1 \cdot \mathbf{c}_1 + \alpha_2 \cdot \mathbf{c}_2 + \cdots + \alpha_k \cdot \mathbf{c}_k, \quad \text{with } \alpha_j \in \mathbb{F}_q \text{ for all } j$$

- Thus, there is a 1-1 correspondence between codewords $\mathbf{c} \in \mathcal{C}$ and k-tuples of coefficients $(\alpha_1, \ldots, \alpha_k) \in {\mathbb{F}_q}^k$.
- Hence, $|\mathcal{C}| = \left|{\mathbb{F}_q}^k\right| = q^k$. $\square$

Number of Codewords

Proposition: An $[n, k]$ linear code $\mathcal{C}$ over $\mathbb{F}_q$ has q^k codewords.

Proof: Let $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k$ be a basis of the subspace $\mathcal{C}$.

- By Proposition B1, every codeword (i.e., vector) $\mathbf{c} \in \mathcal{C}$ can be uniquely expressed as a linear combination

$$\mathbf{c} = \alpha_1 \cdot \mathbf{c}_1 + \alpha_2 \cdot \mathbf{c}_2 + \cdots + \alpha_k \cdot \mathbf{c}_k, \quad \text{with } \alpha_j \in \mathbb{F}_q \text{ for all } j$$

- Thus, there is a 1-1 correspondence between codewords $\mathbf{c} \in \mathcal{C}$ and k-tuples of coefficients $(\alpha_1, \ldots, \alpha_k) \in {\mathbb{F}_q}^k$.
- Hence, $|\mathcal{C}| = \left|{\mathbb{F}_q}^k\right| = q^k$. $\square$

Remarks:

- An $[n, k]$ linear code over $\mathbb{F}_q$ is an $(n, M = q^k)$ block code.
- The rate of an $[n, k]$ linear code over $\mathbb{F}_q$ is

$$R = \frac{1}{n} \log_q(q^k) = \frac{k}{n}.$$

Minimum Distance

Recall that the minimum distance of a block code $\mathcal{C}$ is defined as

$$d_{\min}(\mathcal{C}) \;=\; \min_{\substack{\mathbf{x},\mathbf{y}\in\mathcal{C}\\ \mathbf{x}\neq\mathbf{y}}} d_H(\mathbf{x},\mathbf{y}).$$

Minimum Distance

Recall that the minimum distance of a block code $\mathcal{C}$ is defined as

$$d_{\min}(\mathcal{C}) \;=\; \min_{\substack{\mathbf{x},\mathbf{y}\in\mathcal{C} \\ \mathbf{x}\neq\mathbf{y}}} d_H(\mathbf{x},\mathbf{y}).$$

Definition: The Hamming weight of a word (vector) $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}^n$ is the number of non-zero entries in $\mathbf{x}$:

$$w_H(\mathbf{x}) \;=\; \#\{i : x_i \neq 0\} \;=\; d_H(\mathbf{x},\mathbf{0}).$$

Proposition: For a linear code $\mathcal{C}$,

$$d_{\min}(\mathcal{C}) \;=\; \min_{\substack{\mathbf{c}\in\mathcal{C} \\ \mathbf{c}\neq\mathbf{0}}} w_H(\mathbf{c})$$

Minimum Distance

Recall that the minimum distance of a block code $\mathcal{C}$ is defined as

$$d_{\min}(\mathcal{C}) \;=\; \min_{\substack{\mathbf{x},\mathbf{y}\in\mathcal{C}\\ \mathbf{x}\neq\mathbf{y}}} d_H(\mathbf{x},\mathbf{y}).$$

Definition: The Hamming weight of a word (vector) $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}^n$ is the number of non-zero entries in $\mathbf{x}$:

$$w_H(\mathbf{x}) \;=\; \#\{i : x_i \neq 0\} \;=\; d_H(\mathbf{x},\mathbf{0}).$$

Proposition: For a linear code $\mathcal{C}$,

$$d_{\min}(\mathcal{C}) \;=\; \min_{\substack{\mathbf{c}\in\mathcal{C}\\ \mathbf{c}\neq\mathbf{0}}} w_H(\mathbf{c})$$

Proof: Since $\mathcal{C}$ is linear: $\mathbf{x},\mathbf{y}\in\mathcal{C} \implies \mathbf{x}-\mathbf{y}\in\mathcal{C}$. Therefore,

$$\begin{aligned} d_{\min}(\mathcal{C}) \;&=\; \min_{\substack{\mathbf{x},\mathbf{y}\in\mathcal{C}\\ \mathbf{x}\neq\mathbf{y}}} d_H(\mathbf{x},\mathbf{y}) \\ &=\; \min_{\substack{\mathbf{x},\mathbf{y}\in\mathcal{C}\\ \mathbf{x}\neq\mathbf{y}}} w_H(\mathbf{x}-\mathbf{y}) \;=\; \min_{\substack{\mathbf{c}\in\mathcal{C}\\ \mathbf{c}\neq\mathbf{0}}} w_H(\mathbf{c}). \quad \square \end{aligned}$$

$[n, k, d]$ Notation

Notation: An $[n, k, d]$ (or $[n, k, d]_q$) linear code is an $[n, k]$ (or $[n, k]_q$) linear code with minimum distance d.

(Contrast with (n, M, d) notation for block codes.)

$[n, k, d]$ Notation

Notation: An $[n, k, d]$ (or $[n, k, d]_q$) linear code is an $[n, k]$ (or $[n, k]_q$) linear code with minimum distance d.

(Contrast with (n, M, d) notation for block codes.)

Example:

The repetition code of blocklength n: $\{\underbrace{00\ldots0}_{n\ 0\text{s}}, \underbrace{11\ldots1}_{n\ 1\text{s}}\}$

This is a linear code over $\mathbb{F}_2$ with

blocklength n, dimension 1, and minimum distance n.

In other words, this is an $[n, 1, n]$ binary linear code.

Example: The Single Parity-Check Code

The length-n single parity-check code over $\mathbb{F}_2$:

$$\begin{aligned}\mathcal{C} &= \{x_1 x_2 \ldots x_n : \; x_1 + x_2 + \cdots + x_n \equiv 0 \pmod 2\} \\ &= \text{nullspace}(H), \quad \text{where } H = [\underbrace{1\ 1\ \ \cdots\ \ 1}_{n \text{ columns}}].\end{aligned}$$

Example: The Single Parity-Check Code

The length-n single parity-check code over $\mathbb{F}_2$:

$$\begin{aligned} \mathcal{C} &= \{x_1 x_2 \ldots x_n : \; x_1 + x_2 + \cdots + x_n \equiv 0 \pmod 2\} \\ &= \text{nullspace}(H), \quad \text{where } H = [\underbrace{1\ 1\ \cdots\ 1}_{n \text{ columns}}]. \end{aligned}$$

- By the rank-nullity theorem,

$$\dim(\mathcal{C}) \;=\; n - \text{rank}(H) \;=\; n - 1.$$

- Since there are no codewords of (odd) weight 1,
  and all binary words of (even) weight 2 are in the code,

$$d_{\min}(\mathcal{C}) = 2.$$

Example: The Single Parity-Check Code

The length-n single parity-check code over $\mathbb{F}_2$:

$$\begin{aligned} \mathcal{C} &= \{x_1 x_2 \ldots x_n : \; x_1 + x_2 + \cdots + x_n \equiv 0 \pmod 2\} \\ &= \text{nullspace}(H), \quad \text{where } H = [\underbrace{1\;1\;\cdots\;1}_{n \text{ columns}}]. \end{aligned}$$

- By the rank-nullity theorem,

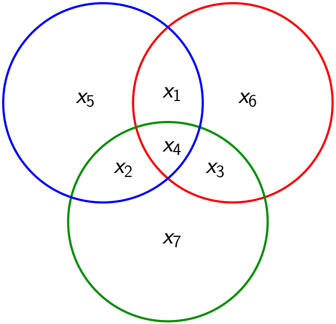
$$\dim(\mathcal{C}) = n - \text{rank}(H) = n - 1.$$

- Since there are no codewords of (odd) weight 1,
  and all binary words of (even) weight 2 are in the code,

$$d_{\min}(\mathcal{C}) = 2.$$

Thus, this is an $[n, n-1, 2]$ binary linear code.

Example: The Length-7 Hamming Code



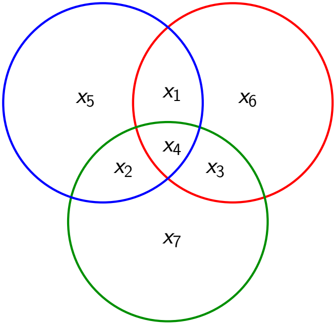
A binary word

$$x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7$$

is in the Hamming code iff

$$x_1 + x_2 + x_4 + x_5 \equiv 0 \pmod 2$$
$$x_1 + x_3 + x_4 + x_6 \equiv 0 \pmod 2$$
$$x_2 + x_3 + x_4 + x_7 \equiv 0 \pmod 2$$

Example: The Length-7 Hamming Code



A binary word

$$x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7$$

is in the Hamming code iff

$$x_1 + x_2 + x_4 + x_5 \equiv 0 \pmod 2$$
$$x_1 + x_3 + x_4 + x_6 \equiv 0 \pmod 2$$
$$x_2 + x_3 + x_4 + x_7 \equiv 0 \pmod 2$$

Re-write these equations in matrix form (over $\mathbb{F}_2$) as

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Example: The Length-7 Hamming Code

Thus, a binary word $x_1x_2x_3x_4x_5x_6x_7$ is in the Hamming code $\mathcal{C}$ iff

$$\underbrace{\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}}_{\mathbf{x}^T} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{0}}.$$

In other words, the Hamming code $\mathcal{C}$ is equal to $\text{nullspace}_{\mathbb{F}_2}(H)$.

Consequently,

- $\dim(\mathcal{C}) = n - \text{rank}_{\mathbb{F}_2}(H) = 7 - 3 = 4.$

Example: The Length-7 Hamming Code

A binary word $x_1x_2x_3x_4x_5x_6x_7$ is in the Hamming code $\mathcal{C}$ iff

$$\underbrace{\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}}_{\mathbf{x}^T} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{0}}.$$

Example: The Length-7 Hamming Code

A binary word $x_1x_2x_3x_4x_5x_6x_7$ is in the Hamming code $\mathcal{C}$ iff

$$x_1 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + x_3 \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + x_4 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + x_5 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + x_6 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + x_7 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Example: The Length-7 Hamming Code

A binary word $x_1x_2x_3x_4x_5x_6x_7$ is in the Hamming code $\mathcal{C}$ iff

$$x_1\begin{bmatrix}1\\1\\0\end{bmatrix}+x_2\begin{bmatrix}1\\0\\1\end{bmatrix}+x_3\begin{bmatrix}0\\1\\1\end{bmatrix}+x_4\begin{bmatrix}1\\1\\1\end{bmatrix}+x_5\begin{bmatrix}1\\0\\0\end{bmatrix}+x_6\begin{bmatrix}0\\1\\0\end{bmatrix}+x_7\begin{bmatrix}0\\0\\1\end{bmatrix}=\begin{bmatrix}0\\0\\0\end{bmatrix}$$

- $\mathcal{C}$ has **no** codewords of weight 1, as no column of H is $\mathbf{0}$.
- $\mathcal{C}$ has **no** codewords of weight 2, as no two columns of H are identical.
- $\mathcal{C}$ does have codewords of weight 3: e.g., the first three columns of H sum to $\mathbf{0}$ over $\mathbb{F}_2$, so 1110000 is in $\mathcal{C}$.

Hence, $d_{\min}(\mathcal{C}) = 3$.

Example: The Length-7 Hamming Code

A binary word $x_1x_2x_3x_4x_5x_6x_7$ is in the Hamming code $\mathcal{C}$ iff

$$x_1 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + x_3 \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + x_4 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + x_5 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + x_6 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + x_7 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

- $\mathcal{C}$ has **no** codewords of weight 1, as no column of H is $\mathbf{0}$.
- $\mathcal{C}$ has **no** codewords of weight 2, as no two columns of H are identical.
- $\mathcal{C}$ does have codewords of weight 3: e.g., the first three columns of H sum to $\mathbf{0}$ over $\mathbb{F}_2$, so 1110000 is in $\mathcal{C}$.

Hence, $d_{\min}(\mathcal{C}) = 3$.

Thus, the Hamming code is a $[7, 4, 3]$ binary linear code.

Minimum Distance of $\mathcal{C} = \text{nullspace}(H)$

Theorem: Let $\mathcal{C} = \text{nullspace}_{\mathbb{F}}(H)$ for some matrix H, with $\mathcal{C} \neq \{\mathbf{0}\}$. The minimum distance of $\mathcal{C}$ is the smallest integer $d > 0$ such that <u>some</u> collection of d columns of H is linearly dependent over $\mathbb{F}$.

Minimum Distance of $\mathcal{C} = \text{nullspace}(H)$

Theorem: Let $\mathcal{C} = \text{nullspace}_{\mathbb{F}}(H)$ for some matrix H, with $\mathcal{C} \neq \{\mathbf{0}\}$. The minimum distance of $\mathcal{C}$ is the smallest integer $d > 0$ such that some collection of d columns of H is linearly dependent over $\mathbb{F}$.

(Equivalently, $d_{\min}(\mathcal{C})$ is the largest integer d such that every collection of $d - 1$ columns of H is linearly independent over $\mathbb{F}$.)

Minimum Distance of $\mathcal{C}$ = nullspace(H)

Theorem: Let $\mathcal{C} = \text{nullspace}_{\mathbb{F}}(H)$ for some matrix H, with $\mathcal{C} \neq \{\mathbf{0}\}$. The minimum distance of $\mathcal{C}$ is the smallest integer $d > 0$ such that some collection of d columns of H is linearly dependent over $\mathbb{F}$.

(Equivalently, $d_{\text{min}}(\mathcal{C})$ is the largest integer d such that every collection of $d-1$ columns of H is linearly independent over $\mathbb{F}$.)

Proof:

- Some collection of d columns of H is linearly dependent
  $\implies$ there exists a codeword of weight $\leq d$
  $\implies$ $d_{\text{min}} \leq d$
- d is the smallest such integer
  $\implies$ all $d-1$ or fewer columns are linearly indep.
  $\implies$ there are no codewords of weight $< d$
  $\implies$ $d_{\text{min}} \geq d$ $\square$

Minimum Distance of $\mathcal{C} = \text{nullspace}(H)$

Theorem: Let $\mathcal{C} = \text{nullspace}_{\mathbb{F}}(H)$ for some matrix H, with $\mathcal{C} \neq \{\mathbf{0}\}$. The minimum distance of $\mathcal{C}$ is the smallest integer $d > 0$ such that some collection of d columns of H is linearly dependent over $\mathbb{F}$.

(Equivalently, $d_{\min}(\mathcal{C})$ is the largest integer d such that every collection of $d-1$ columns of H is linearly independent over $\mathbb{F}$.)

Proof:

- Some collection of d columns of H is linearly dependent
  $\implies$ there exists a codeword of weight $\leq d$
  $\implies$ $d_{\min} \leq d$

- d is the smallest such integer
  $\implies$ all $d-1$ or fewer columns are linearly indep.
  $\implies$ there are no codewords of weight $< d$
  $\implies d_{\min} \geq d$ $\square$

If $\mathcal{C} = \text{nullspace}_{\mathbb{F}}(H)$, then H is called a parity-check matrix for $\mathcal{C}$.

Generator Matrices

Definition: A generator matrix for an $[n, k]$ linear code $\mathcal{C}$ over $\mathbb{F}$ is a $k \times n$ matrix whose rows form a basis of $\mathcal{C}$, i.e.,

$$G = \begin{bmatrix} \rule{2cm}{0.4pt} & \mathbf{g}_1 & \rule{2cm}{0.4pt} \\ \rule{2cm}{0.4pt} & \mathbf{g}_2 & \rule{2cm}{0.4pt} \\ & \vdots & \\ \rule{2cm}{0.4pt} & \mathbf{g}_k & \rule{2cm}{0.4pt} \end{bmatrix},$$

where $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_k$ constitute a basis for $\mathcal{C}$.

- In particular, $\mathrm{rank}_{\mathbb{F}}(G) = \dim_{\mathbb{F}}(\mathcal{C})$.

Generator Matrices

Definition: A generator matrix for an $[n, k]$ linear code $\mathcal{C}$ over $\mathbb{F}$ is a $k \times n$ matrix whose rows form a basis of $\mathcal{C}$, i.e.,

$$G = \begin{bmatrix} \text{———} & \mathbf{g}_1 & \text{———} \\ \text{———} & \mathbf{g}_2 & \text{———} \\ & \vdots & \\ \text{———} & \mathbf{g}_k & \text{———} \end{bmatrix},$$

where $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_k$ constitute a basis for $\mathcal{C}$.

- In particular, $\text{rank}_{\mathbb{F}}(G) = \dim_{\mathbb{F}}(\mathcal{C})$.
- Since $\mathcal{C}$, in general, will have many different bases, it will have many different generator matrices. In fact, we can count the number of its generator matrices quite explicitly.
- Generator matrices are used for encoding.

Number of Generator Matrices

Proposition: An $[n, k]$ linear code over $\mathbb{F}_q$ has

$$\prod_{j=0}^{k-1}(q^k - q^j) \;=\; (q^k - 1)(q^k - q)(q^k - q^2)\cdots(q^k - q^{k-1})$$

distinct generator matrices.

Number of Generator Matrices

Proposition: An $[n, k]$ linear code over $\mathbb{F}_q$ has

$$\prod_{j=0}^{k-1}(q^k - q^j) \;=\; (q^k - 1)(q^k - q)(q^k - q^2)\cdots(q^k - q^{k-1})$$

distinct generator matrices.

Proof: Let $\mathcal{C}$ be an $[n, k]$ linear code over $\mathbb{F}_q$
We can construct a $k \times n$ generator matrix for $\mathcal{C}$ as follows:

- The first row, $\mathbf{g}_1$ can be any non-zero codeword from $\mathcal{C}$: there are $q^k - 1$ choices for $\mathbf{g}_1$.
- The second row $\mathbf{g}_2$ can be any codeword from $\mathcal{C}$ other than those in $\text{span}(\mathbf{g}_1)$: there are $q^k - q$ choices for $\mathbf{g}_2$.
- In this manner, having picked rows $\mathbf{g}_1, \ldots, \mathbf{g}_j$, the $(j+1)$th row $\mathbf{g}_{j+1}$ can be any codeword from $\mathcal{C}$, except for those in $\text{span}(\mathbf{g}_1, \ldots, \mathbf{g}_j)$: there are $q^k - q^j$ choices for $\mathbf{g}_{j+1}$.

Combining the no. of choices for $\mathbf{g}_1, \ldots, \mathbf{g}_k$, we get $\prod_{j=0}^{k-1}(q^k - q^j)$. $\square$

Generator Matrices and Encoding

An $[n, k]$ linear code $\mathcal{C}$ over $\mathbb{F}_q$ has q^k codewords.
So, there is a 1-1 correspondence between ${\mathbb{F}_q}^k$ and $\mathcal{C} \subseteq {\mathbb{F}_q}^n$.

Generator Matrices and Encoding

An $[n, k]$ linear code $\mathcal{C}$ over $\mathbb{F}_q$ has q^k codewords.
So, there is a 1-1 correspondence between ${\mathbb{F}_q}^k$ and $\mathcal{C} \subseteq {\mathbb{F}_q}^n$.

An encoder for $\mathcal{C}$ is a 1-1 mapping from message words $\mathbf{u} = (u_1, \ldots, u_k) \in {\mathbb{F}_q}^k$ to codewords $\mathbf{c} \in \mathcal{C}$.

Generator Matrices and Encoding

An $[n, k]$ linear code $\mathcal{C}$ over $\mathbb{F}_q$ has q^k codewords.
So, there is a 1-1 correspondence between ${\mathbb{F}_q}^k$ and $\mathcal{C} \subseteq {\mathbb{F}_q}^n$.

An encoder for $\mathcal{C}$ is a 1-1 mapping from message words $\mathbf{u} = (u_1, \ldots, u_k) \in {\mathbb{F}_q}^k$ to codewords $\mathbf{c} \in \mathcal{C}$.

Generator matrices give rise to encoders:

- Let G be a $k \times n$ generator matrix for $\mathcal{C}$. Its rows $\mathbf{g}_1, \ldots, \mathbf{g}_k$ form a basis of $\mathcal{C}$.
- Recall that every $\mathbf{c} \in \mathcal{C}$ can be uniquely expressed as a linear combination $\sum_{j=1}^{k} u_j \, \mathbf{g}_j$, with $u_j \in \mathbb{F}_q$ for all j.

Generator Matrices and Encoding

An $[n, k]$ linear code $\mathcal{C}$ over $\mathbb{F}_q$ has q^k codewords.
So, there is a 1-1 correspondence between ${\mathbb{F}_q}^k$ and $\mathcal{C} \subseteq {\mathbb{F}_q}^n$.

An encoder for $\mathcal{C}$ is a 1-1 mapping from message words $\mathbf{u} = (u_1, \ldots, u_k) \in {\mathbb{F}_q}^k$ to codewords $\mathbf{c} \in \mathcal{C}$.

Generator matrices give rise to encoders:

- Let G be a $k \times n$ generator matrix for $\mathcal{C}$. Its rows $\mathbf{g}_1, \ldots, \mathbf{g}_k$ form a basis of $\mathcal{C}$.
- Recall that every $\mathbf{c} \in \mathcal{C}$ can be uniquely expressed as a linear combination $\sum_{j=1}^{k} u_j \, \mathbf{g}_j$, with $u_j \in \mathbb{F}_q$ for all j.
- Hence, the mapping

$$\mathbf{u} = \underbrace{(u_1, \ldots, u_k)}_{\in {\mathbb{F}_q}^k} \longmapsto \mathbf{u}\, G = \sum_{j=1}^{k} u_j \, \mathbf{g}_j$$

  is a bijection between ${\mathbb{F}_q}^k$ and $\mathcal{C}$, i.e., an encoder mapping.

Systematic Generator Matrices

G is called a systematic generator matrix if it is of the form

$$G \;=\; [\, I_k \mid B \,],$$

where I_k is the $k \times k$ identity matrix and B is a $k \times (n-k)$ matrix.

In such a case, the encoder $\mathbf{u} \mapsto \mathbf{u}G$ maps a message $\mathbf{u} \in {\mathbb{F}_q}^k$ to the codeword

$$\mathbf{c} = [\ \underbrace{\mathbf{u}}_{k \text{ symbols}} \ \mid \ \underbrace{\mathbf{u}B}_{n-k \text{ symbols}}\ ].$$

- In $\mathbf{c}$, the first k symbols constitute the message $\mathbf{u}$ itself; they are called information symbols.
- The remaining $n-k$ symbols, $\mathbf{u}B$, are parity-check symbols.

So, retrieving the message encoded within a codeword is easy.

Example

Not every code has a generator matrix in systematic form.

For example,

$$\mathcal{C} = \{0000, 0011, 1100, 1111\}$$

is a $[4, 2, 2]$ binary linear code.

- In every codeword, the 1st coordinate is the same as the 2nd.
- So, it is not possible to have a generator matrix of the form

$$G = \begin{bmatrix} 1 & 0 & * & * \\ 0 & 1 & * & * \end{bmatrix}$$

Row Operations and Column Permutations

In general, a generator matrix for an $[n, k]$ linear code $\mathcal{C}$ can always be found which has the $k \times k$ identity matrix as a submatrix.

Row Operations and Column Permutations

In general, a generator matrix for an $[n, k]$ linear code $\mathcal{C}$ can always be found which has the $k \times k$ identity matrix as a submatrix.

- Pick any generator matrix G for $\mathcal{C}$.
- Since $\text{rank}(G) = k$, some k columns of G are lin. indep.
- Apply elementary row operations on G to obtain a matrix $\tilde{G}$ in which these k columns form an identity matrix.
- Since elementary row operations correspond to exchanging rows or replacing a row with a linear combination of rows, the rows of $\tilde{G}$ still form a basis of $\mathcal{C}$.

Row Operations and Column Permutations

In general, a generator matrix for an $[n, k]$ linear code $\mathcal{C}$ can always be found which has the $k \times k$ identity matrix as a submatrix.

- Pick any generator matrix G for $\mathcal{C}$.
- Since $\mathrm{rank}(G) = k$, some k columns of G are lin. indep.
- Apply elementary row operations on G to obtain a matrix $\tilde{G}$ in which these k columns form an identity matrix.
- Since elementary row operations correspond to exchanging rows or replacing a row with a linear combination of rows, the rows of $\tilde{G}$ still form a basis of $\mathcal{C}$.

If we wish, we may permute the columns of $\tilde{G}$ to bring the identity matrix to the front.

The resulting matrix, call it $\overline{G}$, no longer generates the same code $\mathcal{C}$ as G (or $\tilde{G}$); instead it generates an equivalent code $\overline{\mathcal{C}}$.

Equivalent Codes

Two codes are equivalent if one can be obtained from the other by a permutation of coordinates.

For example,

$$
\begin{array}{ccc}
\mathcal{C} & & \overline{\mathcal{C}} \\
0000 & & 0000 \\
0011 & \overset{\text{exchange 2nd and 3rd coords}}{\longleftrightarrow} & 0101 \\
1100 & & 1010 \\
1111 & & 1111
\end{array}
$$

The code $\overline{\mathcal{C}}$ has a systematic generator matrix

$$
\overline{G} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}
$$

Basis for $\mathcal{C} = \text{nullspace}(H)$

Let $\mathcal{C}$ be an $[n, k]$ linear code over $\mathbb{F}$ with parity-check matrix H, i.e., $\mathcal{C} = \text{nullspace}_{\mathbb{F}}(H)$.

Claim: Vectors $\mathbf{g}_1, \ldots, \mathbf{g}_k$ from $\mathbb{F}^n$ form a basis of $\mathcal{C}$ iff

- $\mathbf{g}_1, \ldots, \mathbf{g}_k$ are linearly independent over $\mathbb{F}$; and
- $H\mathbf{g}_i = \mathbf{0}$ for $i = 1, \ldots, k$.

Basis for $\mathcal{C} = \text{nullspace}(H)$

Let $\mathcal{C}$ be an $[n, k]$ linear code over $\mathbb{F}$ with parity-check matrix H, i.e., $\mathcal{C} = \text{nullspace}_{\mathbb{F}}(H)$.

Claim: Vectors $\mathbf{g}_1, \ldots, \mathbf{g}_k$ from $\mathbb{F}^n$ form a basis of $\mathcal{C}$ iff

- $\mathbf{g}_1, \ldots, \mathbf{g}_k$ are linearly independent over $\mathbb{F}$; and
- $H\mathbf{g}_i = \mathbf{0}$ for $i = 1, \ldots, k$.

Proof: $(\Longrightarrow)$ If $\mathbf{g}_1, \ldots, \mathbf{g}_k$ form a basis of $\mathcal{C}$, then

- they are, by definition of basis, linearly independent
- also, they lie in $\mathcal{C} = \text{nullspace}_{\mathbb{F}}(H)$, so $H\mathbf{g}_i = 0$ for all i.

Basis for $\mathcal{C} = \text{nullspace}(H)$

Let $\mathcal{C}$ be an $[n, k]$ linear code over $\mathbb{F}$ with parity-check matrix H, i.e., $\mathcal{C} = \text{nullspace}_{\mathbb{F}}(H)$.

Claim: Vectors $\mathbf{g}_1, \ldots, \mathbf{g}_k$ from $\mathbb{F}^n$ form a basis of $\mathcal{C}$ iff

- $\mathbf{g}_1, \ldots, \mathbf{g}_k$ are linearly independent over $\mathbb{F}$; and
- $H\mathbf{g}_i = \mathbf{0}$ for $i = 1, \ldots, k$.

Proof: $(\Longrightarrow)$ If $\mathbf{g}_1, \ldots, \mathbf{g}_k$ form a basis of $\mathcal{C}$, then

- they are, by definition of basis, linearly independent
- also, they lie in $\mathcal{C} = \text{nullspace}_{\mathbb{F}}(H)$, so $H\mathbf{g}_i = 0$ for all i.

$(\Longleftarrow)$ Assume $\mathbf{g}_1, \ldots, \mathbf{g}_k$ are lin. indep., and $H\mathbf{g}_i = 0$ for all i.

- $\text{span}(\mathbf{g}_1, \ldots, \mathbf{g}_k)$ is a vector space of dimension k.
- $H \cdot \left(\sum_{i=1}^{k} \alpha_i \mathbf{g}_i\right) \;=\; \sum_{i=1}^{k} \alpha_i \cdot H\mathbf{g}_i \;=\; \mathbf{0}$,
  so $\text{span}(\mathbf{g}_1, \ldots, \mathbf{g}_k) \subseteq \text{nullspace}(H) = \mathcal{C}$.
- Since $\text{span}(\mathbf{g}_1, \ldots, \mathbf{g}_k)$ and $\mathcal{C}$ both have dimension k,
  the $\subseteq$ above is in fact an equality: $\text{span}(\mathbf{g}_1, \ldots, \mathbf{g}_k) = \mathcal{C}$. $\square$

Generator Matrix for $\mathcal{C} = \text{nullspace}(H)$

The result of the prev. claim, when expressed in matrix notation, is:

Proposition: Let $\mathcal{C}$ be an $[n, k]$ linear code over $\mathbb{F}$, with parity-check matrix H. Then, a $k \times n$ matrix G over $\mathbb{F}$ is a generator matrix for $\mathcal{C}$ iff

- $\text{rank}_{\mathbb{F}}(G) = k$, and
- $H\,G^T = 0$. (Here, 0 denotes an all-zero matrix.)

Generator Matrix for $\mathcal{C} = \text{nullspace}(H)$

The result of the prev. claim, when expressed in matrix notation, is:

Proposition: Let $\mathcal{C}$ be an $[n, k]$ linear code over $\mathbb{F}$, with parity-check matrix H. Then, a $k \times n$ matrix G over $\mathbb{F}$ is a generator matrix for $\mathcal{C}$ iff

- $\text{rank}_{\mathbb{F}}(G) = k$, and
- $H\,G^T = 0$. (Here, 0 denotes an all-zero matrix.)

Corollary: Suppose that H is of the form $[\,A \mid I_{n-k}\,]$. Then,

$$G = [\,I_k \mid -A^T\,]$$

is a generator matrix for $\mathcal{C} = \text{nullspace}(H)$.

Generator Matrix for $\mathcal{C} = \text{nullspace}(H)$

The result of the prev. claim, when expressed in matrix notation, is:

Proposition: Let $\mathcal{C}$ be an $[n, k]$ linear code over $\mathbb{F}$, with parity-check matrix H. Then, a $k \times n$ matrix G over $\mathbb{F}$ is a generator matrix for $\mathcal{C}$ iff

- $\text{rank}_{\mathbb{F}}(G) = k$, and
- $H\,G^T = 0$. (Here, 0 denotes an all-zero matrix.)

Corollary: Suppose that H is of the form $[\,A \mid I_{n-k}\,]$. Then,

$$G = [\,I_k \mid -A^T\,]$$

is a generator matrix for $\mathcal{C} = \text{nullspace}(H)$.

Proof:

- $\text{rank}(G) = k$
- $HG^T = [\,A \;\; I_{n-k}\,] \begin{bmatrix} I_k \\ -A \end{bmatrix} = A + (-A) = 0$ $\square$

Example: The [7, 4] Hamming Code

The $[7,4]$ binary Hamming code has parity-check matrix

$$H = \underbrace{\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{I_3},$$

Hence,

$$G = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{I_4} \underbrace{\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}}_{-A^T},$$

is a generator matrix for the code. (Note that $-A^T = A^T$ over $\mathbb{F}_2$.)

"Dot Product"

For $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ belonging to $\mathbb{F}^n$, define the "dot product" $\mathbf{x} \cdot \mathbf{y} = x_1y_1 + x_2y_2 + \cdots + x_ny_n$, all operations being over the field $\mathbb{F}$.

Example:

- Over $\mathbb{R}$, $(1, 0, 1, 0) \cdot (1, 0, 1, 0) = 2$.

"Dot Product"

For $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ belonging to $\mathbb{F}^n$, define the "dot product" $\mathbf{x} \cdot \mathbf{y} = x_1y_1 + x_2y_2 + \cdots + x_ny_n$, all operations being over the field $\mathbb{F}$.

Example:

- Over $\mathbb{R}$, $(1, 0, 1, 0) \cdot (1, 0, 1, 0) = 2$.
- Over $\mathbb{F}_2$, $(1, 0, 1, 0) \cdot (1, 0, 1, 0) = 0$.

  (Thus, over $\mathbb{F}_2$, a vector can be "orthogonal" to itself!)

"Dot Product"

For $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ belonging to $\mathbb{F}^n$, define the "dot product" $\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n$, all operations being over the field $\mathbb{F}$.

Example:

- Over $\mathbb{R}$, $(1, 0, 1, 0) \cdot (1, 0, 1, 0) = 2$.
- Over $\mathbb{F}_2$, $(1, 0, 1, 0) \cdot (1, 0, 1, 0) = 0$.
  (Thus, over $\mathbb{F}_2$, a vector can be "orthogonal" to itself!)

It is easy to verify that

- $\mathbf{x} \cdot \mathbf{y} = \mathbf{y} \cdot \mathbf{x}$ (commutativity)
- $\mathbf{x} \cdot (\alpha_1 \mathbf{y}_1 + \alpha_2 \mathbf{y}_2) = \alpha_1 (\mathbf{x} \cdot \mathbf{y}_1) + \alpha_2 (\mathbf{x} \cdot \mathbf{y}_2)$ for all $\alpha_1, \alpha_2 \in \mathbb{F}$ (linearity)

Dual Codes

Definition: For a linear code $\mathcal{C}$ of blocklength n over $\mathbb{F}$, the dual code is defined as

$$\mathcal{C}^{\perp} \;=\; \{\mathbf{x} \in \mathbb{F}^n : \mathbf{x} \cdot \mathbf{c} = 0 \text{ for all } \mathbf{c} \in \mathcal{C}\}$$

- Informally, $\mathcal{C}^{\perp}$ consists of all vectors in $\mathbb{F}^n$ that are "orthogonal" to all codewords in $\mathcal{C}$.
- It is easy to verify (using the subspace test) that $\mathcal{C}^{\perp}$ is also a linear code.

Dual Codes

Lemma: Let $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k$ be a basis of $\mathcal{C}$. Then, for any $\mathbf{x} \in \mathbb{F}^n$,

$$\mathbf{x} \in \mathcal{C}^\perp \iff \mathbf{x} \cdot \mathbf{c}_j = 0 \text{ for } j = 1, 2, \ldots, k$$

Dual Codes

Lemma: Let $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k$ be a basis of $\mathcal{C}$. Then, for any $\mathbf{x} \in \mathbb{F}^n$,

$$\mathbf{x} \in \mathcal{C}^\perp \iff \mathbf{x} \cdot \mathbf{c}_j = 0 \text{ for } j = 1, 2, \ldots, k$$

Proof: ($\Longrightarrow$) is obvious by definition of $\mathcal{C}^\perp$.

($\Longleftarrow$) Any $\mathbf{c} \in \mathcal{C}$ is expressible as $\sum_{j=1}^{k} \alpha_j \mathbf{c}_j$ for some $\alpha_1, \ldots, \alpha_k \in \mathbb{F}$. So, if $\mathbf{x} \cdot \mathbf{c}_j = 0$ for all j, then by the linearity property of the "dot product",

$$\mathbf{x} \cdot \mathbf{c} = \mathbf{x} \cdot \left(\sum_{j=1}^{k} \alpha_j \mathbf{c}_j \right) = \sum_{j=1}^{k} \alpha_j \underbrace{(\mathbf{x} \cdot \mathbf{c}_j)}_{=0} = 0. \quad \square$$

Dual Codes

Proposition: Let G be any generator matrix for $\mathcal{C}$. Then,

$$\mathcal{C}^\perp = \{\mathbf{x} \in \mathbb{F}^n : G\mathbf{x}^T = 0\}.$$

In other words, $\mathcal{C}^\perp = \text{nullspace}(G)$.

Dual Codes

Proposition: Let G be any generator matrix for $\mathcal{C}$. Then,

$$\mathcal{C}^{\perp} = \{\mathbf{x} \in \mathbb{F}^n : G\mathbf{x}^T = 0\}.$$

In other words, $\mathcal{C}^{\perp} = \text{nullspace}(G)$.

Proof: Write

$$G = \begin{bmatrix} \text{———} & \mathbf{c}_1 & \text{———} \\ \text{———} & \mathbf{c}_2 & \text{———} \\ & \vdots & \\ \text{———} & \mathbf{c}_k & \text{———} \end{bmatrix},$$

where $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k$ constitute a basis for $\mathcal{C}$, and note that

$$G\mathbf{x}^T = \begin{bmatrix} \mathbf{c}_1 \cdot \mathbf{x} \\ \mathbf{c}_2 \cdot \mathbf{x} \\ \vdots \\ \mathbf{c}_k \cdot \mathbf{x} \end{bmatrix}.$$

Dual Codes

Proposition: Let G be any generator matrix for $\mathcal{C}$. Then,

$$\mathcal{C}^{\perp} = \{\mathbf{x} \in \mathbb{F}^n : G\mathbf{x}^T = 0\}.$$

In other words, $\mathcal{C}^{\perp} = \text{nullspace}(G)$.

Proof: Write

$$G = \begin{bmatrix} \rule{2cm}{1pt} & \mathbf{c}_1 & \rule{2cm}{1pt} \\ \rule{2cm}{1pt} & \mathbf{c}_2 & \rule{2cm}{1pt} \\ & \vdots & \\ \rule{2cm}{1pt} & \mathbf{c}_k & \rule{2cm}{1pt} \end{bmatrix},$$

where $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k$ constitute a basis for $\mathcal{C}$, and note that

$$G\mathbf{x}^T = \begin{bmatrix} \mathbf{c}_1 \cdot \mathbf{x} \\ \mathbf{c}_2 \cdot \mathbf{x} \\ \vdots \\ \mathbf{c}_k \cdot \mathbf{x} \end{bmatrix}.$$

Corollary: $\dim(\mathcal{C}^{\perp}) = n - \dim(\mathcal{C})$.

Examples

If $\mathcal{C}$ is an $[n, k]$ linear code, then $\mathcal{C}^\perp$ is an $[n, n-k]$ linear code.

- $G = [1\ 1\ \cdots\ 1]$ generates a repetition code, which is an $[n, 1, n]$ linear code.
  Its dual code, nullspace(G), is the single parity-check code, which is an $[n, n-1, 2]$ linear code.

Examples

If $\mathcal{C}$ is an $[n, k]$ linear code, then $\mathcal{C}^\perp$ is an $[n, n-k]$ linear code.

- $G = [1\ 1\ \cdots\ 1]$ generates a repetition code, which is an $[n, 1, n]$ linear code.
  Its dual code, nullspace(G), is the single parity-check code, which is an $[n, n-1, 2]$ linear code.

- The $[7, 4]$ binary Hamming code is generated by

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

  Its dual code, nullspace(G), is a $[7, 3]$ binary linear code called the simplex code. It has the property that every non-zero codeword has Hamming weight equal to 4.

The Dual of a Dual

Proposition: $(\mathcal{C}^\perp)^\perp = \mathcal{C}$.

Proof: It is easy to see that $\mathcal{C} \subseteq (\mathcal{C}^\perp)^\perp$: indeed, any $\mathbf{c} \in \mathcal{C}$ is "orthogonal" to all codewords in $\mathcal{C}^\perp$, by definition.

Furthermore,

$$\dim\big((\mathcal{C}^\perp)^\perp\big) = n - \dim(\mathcal{C}^\perp) = n - (n - \dim(\mathcal{C})) = \dim(\mathcal{C}).$$

Since $\mathcal{C}$ and $(\mathcal{C}^\perp)^\perp$ are vector spaces of the same (finite) dimension, the inclusion $\mathcal{C} \subseteq (\mathcal{C}^\perp)^\perp$ is in fact an equality. $\square$

The Dual of a Dual

Proposition: $\left(\mathcal{C}^{\perp}\right)^{\perp} = \mathcal{C}$.

Proof: It is easy to see that $\mathcal{C} \subseteq \left(\mathcal{C}^{\perp}\right)^{\perp}$: indeed, any $\mathbf{c} \in \mathcal{C}$ is "orthogonal" to all codewords in $\mathcal{C}^{\perp}$, by definition.

Furthermore,

$$\dim\left(\left(\mathcal{C}^{\perp}\right)^{\perp}\right) = n - \dim(\mathcal{C}^{\perp}) = n - (n - \dim(\mathcal{C})) = \dim(\mathcal{C}).$$

Since $\mathcal{C}$ and $\left(\mathcal{C}^{\perp}\right)^{\perp}$ are vector spaces of the same (finite) dimension, the inclusion $\mathcal{C} \subseteq \left(\mathcal{C}^{\perp}\right)^{\perp}$ is in fact an equality. $\square$

Corollary: Any linear code $\mathcal{C}$ has a parity-check matrix, i.e., $\mathcal{C} = \text{nullspace}(H)$ for some matrix H.

The Dual of a Dual

Proposition: $(\mathcal{C}^\perp)^\perp = \mathcal{C}$.

Proof: It is easy to see that $\mathcal{C} \subseteq (\mathcal{C}^\perp)^\perp$: indeed, any $\mathbf{c} \in \mathcal{C}$ is "orthogonal" to all codewords in $\mathcal{C}^\perp$, by definition.

Furthermore,

$$\dim\big((\mathcal{C}^\perp)^\perp\big) = n - \dim(\mathcal{C}^\perp) = n - (n - \dim(\mathcal{C})) = \dim(\mathcal{C}).$$

Since $\mathcal{C}$ and $(\mathcal{C}^\perp)^\perp$ are vector spaces of the same (finite) dimension, the inclusion $\mathcal{C} \subseteq (\mathcal{C}^\perp)^\perp$ is in fact an equality. $\square$

Corollary: Any linear code $\mathcal{C}$ has a parity-check matrix, i.e., $\mathcal{C} = \text{nullspace}(H)$ for some matrix H.

Proof: Let H be a generator matrix for $\mathcal{C}^\perp$. Then, nullspace(H) is the code $(\mathcal{C}^\perp)^\perp$, which is the same as $\mathcal{C}$. $\square$

Remarks

- Any generator matrix for $\mathcal{C}^\perp$ is a parity-check matrix for $\mathcal{C}$.
- Any generator matrix for $\mathcal{C}$ is a parity-check matrix for $\mathcal{C}^\perp$.

Remarks

- Any generator matrix for $\mathcal{C}^\perp$ is a parity-check matrix for $\mathcal{C}$.
- Any generator matrix for $\mathcal{C}$ is a parity-check matrix for $\mathcal{C}^\perp$.

- The minimum distance of $\mathcal{C}^\perp$ cannot be directly determined from the minimum distance of $\mathcal{C}$.

  More information is needed: The complete weight distribution of $\mathcal{C}$ allows one to determine the complete weight distribution (and hence, d_{min}) of $\mathcal{C}^\perp$, via the MacWilliams Identities.

Decoding

Recall the Minimum Distance Decoding (MDD) rule:

> *Given a received vector* $\mathbf{y} \in \mathbb{F}^n$*, decode to a codeword* $\mathbf{c} \in \mathcal{C}$ *that minimizes* $d_H(\mathbf{y}, \mathbf{c})$*.* *(Ties are broken arbitrarily.)*

We now exploit linearity to give some new perspectives on MDD.

Recall that $d_H(\mathbf{y}, \mathbf{c}) = w_H(\mathbf{y} - \mathbf{c})$.

Decoding

Recall the Minimum Distance Decoding (MDD) rule:

> *Given a received vector* $\mathbf{y} \in \mathbb{F}^n$*, decode to a codeword* $\mathbf{c} \in \mathcal{C}$ *that minimizes* $d_H(\mathbf{y}, \mathbf{c})$*.* (*Ties are broken arbitrarily.*)

We now exploit linearity to give some new perspectives on MDD.

Recall that $d_H(\mathbf{y}, \mathbf{c}) = w_H(\mathbf{y} - \mathbf{c})$.

Let $\mathbf{y} - \mathbf{c} =:$ error vector $\mathbf{e}$

Decoding

Recall the Minimum Distance Decoding (MDD) rule:

> *Given a received vector* $\mathbf{y} \in \mathbb{F}^n$*, decode to a codeword* $\mathbf{c} \in \mathcal{C}$ *that minimizes* $d_H(\mathbf{y}, \mathbf{c})$*.* *(Ties are broken arbitrarily.)*

We now exploit linearity to give some new perspectives on MDD.

Recall that $d_H(\mathbf{y}, \mathbf{c}) = w_H(\mathbf{y} - \mathbf{c})$.

Let $\mathbf{y} - \mathbf{c} =:$ error vector $\mathbf{e} \quad \Longleftrightarrow \quad \boxed{\mathbf{c} = \mathbf{y} - \mathbf{e}}$

Decoding

Recall the Minimum Distance Decoding (MDD) rule:

> *Given a received vector* $\mathbf{y} \in \mathbb{F}^n$*, decode to a codeword* $\mathbf{c} \in \mathcal{C}$ *that minimizes* $d_H(\mathbf{y}, \mathbf{c})$*.* *(Ties are broken arbitrarily.)*

We now exploit linearity to give some new perspectives on MDD.

Recall that $d_H(\mathbf{y}, \mathbf{c}) = w_H(\mathbf{y} - \mathbf{c})$.

Let $\mathbf{y} - \mathbf{c} =:$ error vector $\mathbf{e}$ $\quad \Longleftrightarrow \quad \boxed{\mathbf{c} = \mathbf{y} - \mathbf{e}}$

So, the MDD rule is equivalent to the following:

> *Given a received vector* $\mathbf{y} \in \mathbb{F}^n$*, find an error vector* $\mathbf{e}$ *of least Hamming weight such that* $\mathbf{y} - \mathbf{e} \in \mathcal{C}$*.*
> *Decode to* $\hat{\mathbf{c}} = \mathbf{y} - \mathbf{e}$*.*

The Set of Error Vectors

Define $\mathcal{E}(\mathbf{y}) := \{\mathbf{e} : \mathbf{y} - \mathbf{e} \in \mathcal{C}\}$.

(This is the set of all error vectors that cause codewords to get transformed to $\mathbf{y}$.)

The Set of Error Vectors

Define $\mathcal{E}(\mathbf{y}) := \{\mathbf{e} : \mathbf{y} - \mathbf{e} \in \mathcal{C}\}$.

(This is the set of all error vectors that cause codewords to get transformed to $\mathbf{y}$.)

Note that

$$\begin{aligned} \mathbf{e} \in \mathcal{E}(\mathbf{y}) &\iff \mathbf{y} - \mathbf{e} = \mathbf{c}' \quad \text{for some } \mathbf{c}' \in \mathcal{C} \\ &\iff \mathbf{y} - \mathbf{e} = -\mathbf{c} \quad \text{for some } \mathbf{c} \in \mathcal{C} \\ &\iff \mathbf{e} = \mathbf{y} + \mathbf{c} \quad \text{for some } \mathbf{c} \in \mathcal{C} \end{aligned}$$

Hence,

$$\mathcal{E}(\mathbf{y}) \;=\; \{\mathbf{y} + \mathbf{c} : \; \mathbf{c} \in \mathcal{C}\} \;=:\; \mathbf{y} + \mathcal{C}.$$

The Set of Error Vectors

Define $\mathcal{E}(\mathbf{y}) := \{\mathbf{e} : \mathbf{y} - \mathbf{e} \in \mathcal{C}\}$.

(This is the set of all error vectors that cause codewords to get transformed to $\mathbf{y}$.)

Note that

$$\begin{aligned} \mathbf{e} \in \mathcal{E}(\mathbf{y}) &\iff \mathbf{y} - \mathbf{e} = \mathbf{c}' \quad \text{for some } \mathbf{c}' \in \mathcal{C} \\ &\iff \mathbf{y} - \mathbf{e} = -\mathbf{c} \quad \text{for some } \mathbf{c} \in \mathcal{C} \\ &\iff \mathbf{e} = \mathbf{y} + \mathbf{c} \quad \text{for some } \mathbf{c} \in \mathcal{C} \end{aligned}$$

Hence,

$$\mathcal{E}(\mathbf{y}) = \{\mathbf{y} + \mathbf{c} : \mathbf{c} \in \mathcal{C}\} =: \mathbf{y} + \mathcal{C}.$$

$\boxed{\text{Thus, } \mathcal{E}(\mathbf{y}) \text{ is a coset of } \mathcal{C}.}$

Another Perspective on MDD

MDD can now be viewed as the following algorithm: Given a received vector $\mathbf{y} \in \mathbb{F}^n$,

1. find the coset of $\mathcal{C}$ to which $\mathbf{y}$ belongs
2. identify a vector, $\mathbf{e}$, of least weight from that coset
3. set $\hat{\mathbf{c}} = \mathbf{y} - \mathbf{e}$

Another Perspective on MDD

MDD can now be viewed as the following algorithm: Given a received vector $\mathbf{y} \in \mathbb{F}^n$,

1. find the coset of $\mathcal{C}$ to which $\mathbf{y}$ belongs
2. identify a vector, $\mathbf{e}$, of least weight from that coset
3. set $\hat{\mathbf{c}} = \mathbf{y} - \mathbf{e}$

Some advantages offered by this perspective on MDD:

- All cosets of $\mathcal{C}$ can be pre-calculated and stored at the decoder. This pre-computation has to be done just once, and does not have to be repeated each time a new $\mathbf{y}$ is received.
- A vector of least weight within each coset, called a coset leader, can also be identified in advance and stored.

Cosets — A Quick Review

Definition: A coset of $\mathcal{C}$ in $\mathbb{F}^n$ is a set of the form
$\mathbf{b} + \mathcal{C} := \{\mathbf{b} + \mathbf{c} : \ \mathbf{c} \in \mathcal{C}\}$, for some $\mathbf{b} \in \mathbb{F}^n$.

Cosets — A Quick Review

Definition: A coset of $\mathcal{C}$ in $\mathbb{F}^n$ is a set of the form
$\mathbf{b} + \mathcal{C} := \{\mathbf{b} + \mathbf{c} : \ \mathbf{c} \in \mathcal{C}\}$, for some $\mathbf{b} \in \mathbb{F}^n$.

Some basic facts:

1. $|\mathbf{b} + \mathcal{C}| = |\mathcal{C}| = q^k$, i.e., all cosets have the same size.

   Proof: The map $\mathbf{c} \mapsto \mathbf{b} + \mathbf{c}$ is a bijection between $\mathcal{C}$ and $\mathbf{b} + \mathcal{C}$. $\square$

Cosets — A Quick Review

Definition: A coset of $\mathcal{C}$ in $\mathbb{F}^n$ is a set of the form
$\mathbf{b} + \mathcal{C} := \{\mathbf{b} + \mathbf{c} : \ \mathbf{c} \in \mathcal{C}\}$, for some $\mathbf{b} \in \mathbb{F}^n$.

Some basic facts:

1. $|\mathbf{b} + \mathcal{C}| = |\mathcal{C}| = q^k$, i.e., all cosets have the same size.

   Proof: The map $\mathbf{c} \mapsto \mathbf{b} + \mathbf{c}$ is a bijection between $\mathcal{C}$ and $\mathbf{b} + \mathcal{C}$. $\square$

2. Each $\mathbf{b} \in \mathbb{F}^n$ lies in some coset of $\mathcal{C}$.

   Proof: Clearly, $\mathbf{b} \in \mathbf{b} + \mathcal{C}$, since $\mathbf{0} \in \mathcal{C}$. $\square$

Cosets — A Quick Review

Definition: A coset of $\mathcal{C}$ in $\mathbb{F}^n$ is a set of the form
$\mathbf{b} + \mathcal{C} := \{\mathbf{b} + \mathbf{c} : \ \mathbf{c} \in \mathcal{C}\}$, for some $\mathbf{b} \in \mathbb{F}^n$.

Some basic facts:

1. $|\mathbf{b} + \mathcal{C}| = |\mathcal{C}| = q^k$, i.e., all cosets have the same size.

   Proof: The map $\mathbf{c} \mapsto \mathbf{b} + \mathbf{c}$ is a bijection between $\mathcal{C}$ and $\mathbf{b} + \mathcal{C}$. $\square$

2. Each $\mathbf{b} \in \mathbb{F}^n$ lies in some coset of $\mathcal{C}$.

   Proof: Clearly, $\mathbf{b} \in \mathbf{b} + \mathcal{C}$, since $\mathbf{0} \in \mathcal{C}$. $\square$

3. $\mathbf{a}$ and $\mathbf{b}$ are in the same coset of $\mathcal{C}$ iff $\mathbf{a} - \mathbf{b} \in \mathcal{C}$.

   Proof: Suppose that $\mathbf{b} \in \mathbf{y} + \mathcal{C}$, so that $\mathbf{b} = \mathbf{y} + \mathbf{c}$ for some $\mathbf{c} \in \mathcal{C}$.
   Then, $\mathbf{a} \ = \ \mathbf{b} + (\mathbf{a} - \mathbf{b}) \ = \ \mathbf{y} + (\mathbf{c} + \mathbf{a} - \mathbf{b})$.
   Hence, $\mathbf{a}$ is also in $\mathbf{y} + \mathcal{C} \iff \mathbf{c} + (\mathbf{a} - \mathbf{b}) \in \mathcal{C}$
   $\iff \mathbf{a} - \mathbf{b} \in \mathcal{C}$. $\square$

Cosets — A Quick Review

Theorem: The distinct cosets of a linear code $\mathcal{C} \subseteq \mathbb{F}^n$ form a partition of $\mathbb{F}^n$.

Proof: Define a relation $\sim$ on $\mathbb{F}^n$ as follows: for $\mathbf{a}, \mathbf{b} \in \mathbb{F}^n$,

$$\mathbf{a} \sim \mathbf{b} \iff \mathbf{a} - \mathbf{b} \in \mathcal{C}$$

- It is easy to verify that $\sim$ is an equivalence relation.
- Hence, the equivalence classes of $\sim$ form a partition of $\mathbb{F}^n$.
- However, by Basic Fact 3, the equivalence classes of $\sim$ are precisely the cosets of $\mathcal{C}$. $\square$

Cosets — A Quick Review

Theorem: The distinct cosets of a linear code $\mathcal{C} \subseteq \mathbb{F}^n$ form a partition of $\mathbb{F}^n$.

Proof: Define a relation $\sim$ on $\mathbb{F}^n$ as follows: for $\mathbf{a}, \mathbf{b} \in \mathbb{F}^n$,

$$\mathbf{a} \sim \mathbf{b} \iff \mathbf{a} - \mathbf{b} \in \mathcal{C}$$

- It is easy to verify that $\sim$ is an equivalence relation.
- Hence, the equivalence classes of $\sim$ form a partition of $\mathbb{F}^n$.
- However, by Basic Fact 3, the equivalence classes of $\sim$ are precisely the cosets of $\mathcal{C}$. $\square$

Corollary: An $[n, k]$ linear code over $\mathbb{F}_q$ has q^{n-k} cosets.

Proof: Each coset has q^k words (by Basic Fact 1).
Put together, the cosets partition ${\mathbb{F}_q}^n$.
Hence, the number of cosets is $q^n/q^k = q^{n-k}$. $\square$

Back to MDD

$\mathcal{C}$ an $[n, k]$ linear code over $\mathbb{F}_q$.

Pre-computation (one-time):

- List out all q^{n-k} cosets of $\mathcal{C}$.
- Identify a coset leader (word of least weight) from each coset.

Given a received vector $\mathbf{y} \in \mathbb{F}_q{}^n$,

1. find the coset of $\mathcal{C}$ to which $\mathbf{y}$ belongs
2. retrieve the coset leader, $\mathbf{e}$, identified for that coset
3. decode to $\hat{\mathbf{c}} = \mathbf{y} - \mathbf{e}$

Back to MDD

$\mathcal{C}$ an $[n, k]$ linear code over $\mathbb{F}_q$.

Pre-computation (one-time):

- List out all q^{n-k} cosets of $\mathcal{C}$.
- Identify a coset leader (word of least weight) from each coset.

Given a received vector $\mathbf{y} \in {\mathbb{F}_q}^n$,

1. find the coset of $\mathcal{C}$ to which $\mathbf{y}$ belongs
2. retrieve the coset leader, $\mathbf{e}$, identified for that coset
3. decode to $\hat{\mathbf{c}} = \mathbf{y} - \mathbf{e}$

Example: Let $\mathcal{C}$ be the $[6, 3]$ binary linear code generated by

$$G = \left[\begin{array}{cccccc} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ \underbrace{0 \quad 0 \quad 1}_{I_3} & & & 0 & 1 & 1 \end{array} \right]$$

$\mathcal{C}$ has $2^{n-k} = 8$ cosets. It can be verified that $d_{\min}(\mathcal{C}) = 3$.

The Standard Array

A standard array for a linear code is a listing of the code and all its cosets in the form of an array.

$$
\begin{array}{rcl}
\mathcal{C} & = & \underbrace{\begin{array}{c}000000\\100000\\010000\\001000\\000100\\000010\\000001\\100001\end{array}}_{\text{coset leaders}}
\end{array}
$$

$\mathcal{C}$	$=$	000000	100110	010101	001011	111000	011110	101101	110011
$100000+\mathcal{C}$	$=$	100000	000110	110101	101011	011000	111110	001101	010011
$010000+\mathcal{C}$	$=$	010000	110110	000101	011011	101000	001110	111101	100011
$001000+\mathcal{C}$	$=$	001000	101110	011101	000011	110000	010110	100101	111011
$000100+\mathcal{C}$	$=$	000100	100010	010001	001111	111100	011010	101001	110111
$000010+\mathcal{C}$	$=$	000010	100100	010111	001001	111010	011100	101111	110001
$000001+\mathcal{C}$	$=$	000001	100111	010100	001010	111001	011111	101100	110010
$100001+\mathcal{C}$	$=$	100001	000111	110100	101010	011001	111111	001100	010010

coset leaders

The Standard Array

A standard array for a linear code is a listing of the code and all its cosets in the form of an array.

$$
\begin{array}{rcccccccccc}
\mathcal{C} & = & 000000 & 100110 & 010101 & 001011 & 111000 & 011110 & 101101 & 110011 \\
100000+\mathcal{C} & = & 100000 & 000110 & 110101 & 101011 & 011000 & 111110 & 001101 & 010011 \\
010000+\mathcal{C} & = & 010000 & 110110 & 000101 & 011011 & 101000 & 001110 & 111101 & 100011 \\
001000+\mathcal{C} & = & 001000 & 101110 & 011101 & 000011 & 110000 & 010110 & 100101 & 111011 \\
000100+\mathcal{C} & = & 000100 & 100010 & 010001 & 001111 & 111100 & 011010 & 101001 & 110111 \\
000010+\mathcal{C} & = & 000010 & 100100 & 010111 & 001001 & 111010 & 011100 & 101111 & 110001 \\
000001+\mathcal{C} & = & 000001 & 100111 & 010100 & 001010 & 111001 & 011111 & 101100 & 110010 \\
100001+\mathcal{C} & = & \underbrace{100001}_{\substack{\text{coset}\\ \text{leaders}}} & 000111 & 110100 & 101010 & 011001 & 111111 & 001100 & 010010
\end{array}
$$

Note: There may be multiple choices for coset leader: for example, in the last coset, we could have alternatively chosen 001100 or 010010 as coset leaders.

Unique Coset Leader

When is there a unique coset leader?

Note that any word of Hamming weight $\leq \lfloor \frac{d_{\text{min}}-1}{2} \rfloor$ is always the unique word of least weight within its coset.

[If two words $\mathbf{a}$, $\mathbf{b}$ of weight $\leq \lfloor \frac{d_{\text{min}}-1}{2} \rfloor$ were in the same coset, then their difference $\mathbf{a} - \mathbf{b}$ would be a word of weight $< d_{\text{min}}$ in $\mathcal{C}$.]

Unique Coset Leader

When is there a unique coset leader?

Note that any word of Hamming weight $\leq \lfloor \frac{d_{\min}-1}{2} \rfloor$ is always the unique word of least weight within its coset.

[If two words $\mathbf{a}$, $\mathbf{b}$ of weight $\leq \lfloor \frac{d_{\min}-1}{2} \rfloor$ were in the same coset, then their difference $\mathbf{a} - \mathbf{b}$ would be a word of weight $< d_{\min}$ in $\mathcal{C}$.]

Example: In the previous example of the $[6, 3, 3]$ linear code,

- All words of weight $\leq \lfloor \frac{d_{\min}-1}{2} \rfloor = 1$ are the unique coset leaders of their respective cosets.

Correctable Error Patterns

The coset leaders are precisely the error patterns that get corrected by the standard array implementation of MDD.

Example: Suppose $\mathbf{c} = 100110$ is transmitted.

- Suppose $\mathbf{y} = 110110$ is received. So, $\mathbf{e} = \mathbf{y} - \mathbf{c} = 010000$ is the error vector (but this is not *a priori* known to the decoder).

Correctable Error Patterns

The coset leaders are precisely the error patterns that get corrected by the standard array implementation of MDD.

Example: Suppose $\mathbf{c} = 100110$ is transmitted.

- Suppose $\mathbf{y} = 110110$ is received. So, $\mathbf{e} = \mathbf{y} - \mathbf{c} = 010000$ is the error vector (but this is not *a priori* known to the decoder).
- $\mathbf{y}$ is in the coset $010000 + \mathcal{C}$, for which 010000, being the unique word of least weight, is the coset leader.
- So, $\mathbf{y}$ gets decoded to $\hat{\mathbf{c}} = \mathbf{y} - \text{coset leader} = 100110 = \mathbf{c}$.
  The error vector gets corrected!
  Note that this $\mathbf{c}$ is the unique closest codeword to $\mathbf{y}$.

Correctable Error Patterns

Example: Suppose $\mathbf{c} = 100110$ is transmitted.

- Suppose $\mathbf{y} = 101010$ is received. Now, $\mathbf{e} = \mathbf{y} - \mathbf{c} = 001100$ is the error vector (again, not *a priori* known to the decoder).

Correctable Error Patterns

Example: Suppose $\mathbf{c} = 100110$ is transmitted.

- Suppose $\mathbf{y} = 101010$ is received. Now, $\mathbf{e} = \mathbf{y} - \mathbf{c} = 001100$ is the error vector (again, not *a priori* known to the decoder).

- $\mathbf{y}$ is in the coset $100001 + \mathcal{C}$, for which 100001 was chosen to be the coset leader in our standard array.

- So, $\mathbf{y}$ gets decoded to $\hat{\mathbf{c}} = \mathbf{y} - \text{coset leader} = 001011 \neq \mathbf{c}$. The error vector does <u>not</u> get corrected this time.

  Note that this $\hat{\mathbf{c}}$ is a closest codeword to $\mathbf{y}$, but it is not the unique such codeword:

  - There are three words of weight 2 in the same coset as $\mathbf{y}$, including the actual error vector $\mathbf{e} = 100001$.
  - Subtracting any of these weight-2 words from $\mathbf{y}$ would yield a codeword at distance 2 from $\mathbf{y}$.

Some Observations

- Under a standard array implementation of MDD, an error vector **e** that is added in transmission gets corrected iff **e** is one of the coset leaders of the standard array.

  For example, for the $[6, 3, 3]$ binary linear code, the standard array decoder corrects <u>all</u> single-error patterns, but can only correct <u>one</u> other error pattern (which could be any one word from the last coset).

Some Observations

- Under a standard array implementation of MDD, an error vector $\mathbf{e}$ that is added in transmission gets corrected iff $\mathbf{e}$ is one of the coset leaders of the standard array.

  For example, for the $[6,3,3]$ binary linear code, the standard array decoder corrects <u>all</u> single-error patterns, but can only correct <u>one</u> other error pattern (which could be any one word from the last coset).

- Hence, the prob. of decoding error, given that $\mathbf{c}$ is the transmitted codeword, is

$$P_{\text{err}}(\mathbf{c}) = \sum_{\mathbf{y}:\ \mathbf{y}-\mathbf{c} \text{ is not a coset leader}} \Pr[\mathbf{y} \mid \mathbf{c}]$$

Storage Complexity

- Even the one-time pre-computation and storage of the standard array, which contains all the q^n words in ${\mathbb{F}_q}^n$, is infeasible for $n \sim 100$ or more.

- Storage of the entire array is not needed if we can find some means of identifying the coset to which a given $\mathbf{y} \in {\mathbb{F}_q}^n$ belongs.

  It would then suffice to store only the q^{n-k} coset leaders.

Syndromes

Let H be an $(n-k) \times n$ parity-check matrix for $\mathcal{C}$.

Definition: The syndrome of a vector $\mathbf{y} \in {\mathbb{F}_q}^n$ is $\mathbf{s} = H\mathbf{y}^T$.

Some facts:

1. $\mathbf{s}$ is a (column) vector belonging to ${\mathbb{F}_q}^{n-k}$
   $\Longrightarrow$ there are q^{n-k} possible syndromes.
2. The syndrome of $\mathbf{y} \in {\mathbb{F}_q}^n$ is $\mathbf{0}$ iff $\mathbf{y} \in \mathcal{C}$.

Syndromes

Let H be an $(n-k) \times n$ parity-check matrix for $\mathcal{C}$.

Definition: The syndrome of a vector $\mathbf{y} \in \mathbb{F}_q{}^n$ is $\mathbf{s} = H\mathbf{y}^T$.

Some facts:

1. $\mathbf{s}$ is a (column) vector belonging to $\mathbb{F}_q{}^{n-k}$
   $\implies$ there are q^{n-k} possible syndromes.
2. The syndrome of $\mathbf{y} \in \mathbb{F}_q{}^n$ is $\mathbf{0}$ iff $\mathbf{y} \in \mathcal{C}$.
3. Two vectors $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{F}_q{}^n$ are in the same coset of $\mathcal{C}$ iff they have the same syndrome.
   Proof:
   $$\begin{aligned} \mathbf{y}_1, \mathbf{y}_2 \text{ are in the same coset of } \mathcal{C} &\iff \mathbf{y}_1 - \mathbf{y}_2 \in \mathcal{C} \\ &\iff H(\mathbf{y}_1 - \mathbf{y}_2)^T = \mathbf{0} \\ &\iff H\mathbf{y}_1^T = H\mathbf{y}_2^T \quad \square \end{aligned}$$

Syndromes

Let H be an $(n-k) \times n$ parity-check matrix for $\mathcal{C}$.

Definition: The syndrome of a vector $\mathbf{y} \in \mathbb{F}_q{}^n$ is $\mathbf{s} = H\mathbf{y}^T$.

Some facts:

1. $\mathbf{s}$ is a (column) vector belonging to $\mathbb{F}_q{}^{n-k}$
   $\Longrightarrow$ there are q^{n-k} possible syndromes.
2. The syndrome of $\mathbf{y} \in \mathbb{F}_q{}^n$ is $\mathbf{0}$ iff $\mathbf{y} \in \mathcal{C}$.
3. Two vectors $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{F}_q{}^n$ are in the same coset of $\mathcal{C}$ iff they have the same syndrome.
   Proof:
   $$\begin{aligned} \mathbf{y}_1, \mathbf{y}_2 \text{ are in the same coset of } \mathcal{C} &\iff \mathbf{y}_1 - \mathbf{y}_2 \in \mathcal{C} \\ &\iff H(\mathbf{y}_1 - \mathbf{y}_2)^T = \mathbf{0} \\ &\iff H\mathbf{y}_1^T = H\mathbf{y}_2^T \quad \square \end{aligned}$$

   Thus, the syndrome of a word $\mathbf{y} \in \mathbb{F}_q{}^n$ uniquely determines the coset to which it belongs.

Syndrome Decoding

Therefore, to implement MDD, it is enough to store a list of coset leaders along with the syndromes for their respective cosets. In all, this requires storing only q^{n-k} coset leader – syndrome pairs.

MDD then reduces to syndrome decoding: Given a rcvd $\mathbf{y} \in \mathbb{F}_q{}^n$,

1. compute $\mathbf{s} = H\mathbf{y}^T$
2. retrieve the coset leader, $\mathbf{e}$, corresponding to syndrome $\mathbf{s}$
3. decode to $\hat{\mathbf{c}} = \mathbf{y} - \mathbf{e}$

Example

Consider, once again, the $[6, 3, 3]$ code $\mathcal{C}$ generated by

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

A parity-check matrix for $\mathcal{C}$ is

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

(This can be verified by checking that the rows of H form a basis of $\mathcal{C}^{\perp}$.)

Example

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

We then form a table of syndromes and corresp. coset leaders:

Coset leader, $\mathbf{e}$	Syndrome $\mathbf{s} = H\mathbf{e}^T$
000000	$[0\ 0\ 0]^T$
100000	$[1\ 1\ 0]^T$
010000	$[1\ 0\ 1]^T$
001000	$[0\ 1\ 1]^T$
000100	$[1\ 0\ 0]^T$
000010	$[0\ 1\ 0]^T$
000001	$[0\ 0\ 1]^T$
100001	$[1\ 1\ 1]^T$

Example

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

We then form a table of syndromes and corresp. coset leaders:

Coset leader, $\mathbf{e}$	Syndrome $\mathbf{s} = H\mathbf{e}^T$
000000	$[0\ 0\ 0]^T$
100000	$[1\ 1\ 0]^T$
010000	$[1\ 0\ 1]^T$
001000	$[0\ 1\ 1]^T$
000100	$[1\ 0\ 0]^T$
000010	$[0\ 1\ 0]^T$
000001	$[0\ 0\ 1]^T$
100001	$[1\ 1\ 1]^T$

To illustrate syndrome decoding,

$$\begin{aligned} \mathbf{y} = 110110 \quad & \Longrightarrow \quad \mathbf{s} = H\mathbf{y}^T = [1\ 0\ 1]^T \\ & \Longrightarrow \quad \mathbf{e} = 010000 \quad \Longrightarrow \quad \hat{\mathbf{c}} = \mathbf{y} - \mathbf{e} = 100110 \end{aligned}$$

t-Error-Correcting Codes

Definition: A code is *t-error-correcting* (for some $t \in \mathbb{Z}_+$) if all error patterns $\mathbf{e}$ of weight $w_H(\mathbf{e}) \leq t$ can be corrected under minimum distance decoding (or equivalently, under syndrome decoding).

- Thus, a code is t-error-correcting iff all the distinct error vectors of weight $\leq t$ can be chosen to be coset leaders of distinct cosets (or equivalently, all these error vectors lie in distinct cosets).

t-Error-Correcting Codes

Definition: A code is *t-error-correcting* (for some $t \in \mathbb{Z}_+$) if all error patterns $\mathbf{e}$ of weight $w_H(\mathbf{e}) \leq t$ can be corrected under minimum distance decoding (or equivalently, under syndrome decoding).

- Thus, a code is t-error-correcting iff all the distinct error vectors of weight $\leq t$ can be chosen to be coset leaders of distinct cosets (or equivalently, all these error vectors lie in distinct cosets).

Lemma 1: A linear code with minimum distance d is t-error-correcting for any $t \leq \frac{d-1}{2}$.

Proof: Consider any $t \leq \frac{d-1}{2}$.
Distinct words of weight $\leq t$ cannot lie in the same coset; if they did, their difference would be a codeword of weight $\leq 2t < d$, which cannot happen. $\square$.

The Hamming Bound for Linear Codes

Proposition 2 (The Hamming bound for linear codes):

If an $[n, k]$ linear code over $\mathbb{F}_q$ is t-error-correcting, then

$$\sum_{i=0}^{t} \binom{n}{i} (q-1)^i \ \leq \ q^{n-k}.$$

The Hamming Bound for Linear Codes

Proposition 2 (The Hamming bound for linear codes):
If an $[n, k]$ linear code over $\mathbb{F}_q$ is t-error-correcting, then

$$\sum_{i=0}^{t} \binom{n}{i} (q-1)^i \ \leq \ q^{n-k}.$$

Proof:

- LHS = no. of error vectors of weight $\leq t$
- RHS = no. of cosets $\square$

The Hamming Bound for Linear Codes

Proposition 2 (The Hamming bound for linear codes):
If an $[n, k]$ linear code over $\mathbb{F}_q$ is t-error-correcting, then

$$\sum_{i=0}^{t} \binom{n}{i} (q-1)^i \;\leq\; q^{n-k}.$$

Proof:

- LHS = no. of error vectors of weight $\leq t$
- RHS = no. of cosets $\square$

Definition: A t-error-correcting linear code whose parameters satisfy the Hamming bound with equality is called a perfect code.

- Such a code, under MDD, can correct **all** error patterns of weight $\leq t$, but **none** of weight $t+1$ or more.

Examples of Perfect Codes

- A trivial example of a perfect code is $\mathbb{F}^n$ (over any field $\mathbb{F}$), which has parameters $[n, n, 1]$. This is a perfect 0-error-correcting code: $\binom{n}{0}(q-1)^0 = 1 = q^{n-n}$.
- The $[n, 1, n]$ binary repetition code. It is a perfect $\frac{n-1}{2}$-error-correcting code, for odd values of n:

$$\sum_{i=0}^{\frac{n-1}{2}} \binom{n}{i} = \frac{1}{2} \sum_{i=0}^{n} \binom{n}{i} = 2^{n-1} = 2^{n-k}.$$

Examples of Perfect Codes

- A trivial example of a perfect code is $\mathbb{F}^n$ (over any field $\mathbb{F}$), which has parameters $[n, n, 1]$. This is a perfect 0-error-correcting code: $\binom{n}{0}(q-1)^0 = 1 = q^{n-n}$.

- The $[n, 1, n]$ binary repetition code. It is a perfect $\frac{n-1}{2}$-error-correcting code, for odd values of n:

$$\sum_{i=0}^{\frac{n-1}{2}} \binom{n}{i} = \frac{1}{2} \sum_{i=0}^{n} \binom{n}{i} = 2^{n-1} = 2^{n-k}.$$

- The $[7, 4, 3]$ binary Hamming code is a perfect 1-error-correcting code:

$$\binom{7}{0} + \binom{7}{1} = 8 = 2^{7-4}.$$

  We generalize this construction to obtain a family of perfect single-error-correcting codes.

Single-Error-Correcting Linear Codes

A linear code is single-error-correcting iff all distinct error vectors of Hamming weight ≤ 1 lie in distinct cosets, i.e., have distinct syndromes.

Single-Error-Correcting Linear Codes

A linear code is single-error-correcting iff all distinct error vectors of Hamming weight ≤ 1 lie in distinct cosets, i.e., have distinct syndromes.

Consider a linear code with parity-check matrix

$$H = [\mathbf{h}_1 \ \mathbf{h}_2 \ \cdots \ \mathbf{h}_n],$$

the $\mathbf{h}_i$'s being column vectors.

- Error vector of weight 0, i.e., $\mathbf{e} = [0 \ 0 \ \ldots \ 0]$, has syndrome $\mathbf{0}$.
- Error vector of weight 1, i.e., $\mathbf{e} = [0 \ \ldots \ 0 \ \alpha \ 0 \ \ldots \ 0]$ for some $\alpha \in \mathbb{F} \setminus \{0\}$ in the ith coordinate, has syndrome $H\mathbf{e}^T = \alpha \, \mathbf{h}_i$.

Single-Error-Correcting Linear Codes

A linear code is single-error-correcting iff all distinct error vectors of Hamming weight ≤ 1 lie in distinct cosets, i.e., have distinct syndromes.

Consider a linear code with parity-check matrix

$$H = [\mathbf{h}_1 \ \mathbf{h}_2 \ \cdots \ \mathbf{h}_n],$$

the $\mathbf{h}_i$'s being column vectors.

- Error vector of weight 0, i.e., $\mathbf{e} = [0\ 0\ \ldots\ 0]$, has syndrome $\mathbf{0}$.
- Error vector of weight 1, i.e., $\mathbf{e} = [0\ \ldots\ 0\ \alpha\ 0\ \ldots\ 0]$ for some $\alpha \in \mathbb{F} \setminus \{0\}$ in the ith coordinate, has syndrome $H\mathbf{e}^T = \alpha\,\mathbf{h}_i$.

Thus, all error vectors of weight ≤ 1 have distinct syndromes iff

- $\alpha\,\mathbf{h}_i \neq \mathbf{0}$ for any i and $\alpha \in \mathbb{F} \setminus \{0\} \quad \Longleftrightarrow \quad \mathbf{h}_i \neq \mathbf{0}$ for all i

Single-Error-Correcting Linear Codes

A linear code is single-error-correcting iff all distinct error vectors of Hamming weight ≤ 1 lie in distinct cosets, i.e., have distinct syndromes.

Consider a linear code with parity-check matrix

$$H = [\mathbf{h}_1 \ \mathbf{h}_2 \ \cdots \ \mathbf{h}_n],$$

the $\mathbf{h}_i$'s being column vectors.

- Error vector of weight 0, i.e., $\mathbf{e} = [0\ 0\ \ldots\ 0]$, has syndrome $\mathbf{0}$.
- Error vector of weight 1, i.e., $\mathbf{e} = [0\ \ldots\ 0\ \alpha\ 0\ \ldots\ 0]$ for some $\alpha \in \mathbb{F} \setminus \{0\}$ in the ith coordinate, has syndrome $H\mathbf{e}^T = \alpha\,\mathbf{h}_i$.

Thus, all error vectors of weight ≤ 1 have distinct syndromes iff

- $\alpha\,\mathbf{h}_i \neq \mathbf{0}$ for any i and $\alpha \in \mathbb{F} \setminus \{0\} \iff \mathbf{h}_i \neq \mathbf{0}$ for all i
- $\alpha\,\mathbf{h}_i \neq \beta\,\mathbf{h}_j$ for distinct i, j, and any $\alpha, \beta \in \mathbb{F} \setminus \{0\}$

  $\iff \mathbf{h}_i \neq \gamma\,\mathbf{h}_j$ for distinct i, j, and $\gamma\,(= \alpha^{-1}\beta) \in \mathbb{F} \setminus \{0\}$

Single-Error-Correcting Linear Codes

A linear code is single-error-correcting iff all distinct error vectors of Hamming weight ≤ 1 lie in distinct cosets, i.e., have distinct syndromes.

Consider a linear code with parity-check matrix

$$H = [\mathbf{h}_1 \ \mathbf{h}_2 \ \cdots \ \mathbf{h}_n],$$

the $\mathbf{h}_i$'s being column vectors.

- Error vector of weight 0, i.e., $\mathbf{e} = [0\ 0\ \ldots\ 0]$, has syndrome $\mathbf{0}$.
- Error vector of weight 1, i.e., $\mathbf{e} = [0\ \ldots\ 0\ \alpha\ 0\ \ldots\ 0]$ for some $\alpha \in \mathbb{F} \setminus \{0\}$ in the ith coordinate, has syndrome $H\mathbf{e}^T = \alpha\,\mathbf{h}_i$.

Thus, all error vectors of weight ≤ 1 have distinct syndromes iff

- $\mathbf{h}_i \neq \mathbf{0}$ for all i
- $\mathbf{h}_i \neq \gamma\,\mathbf{h}_j$ for distinct i, j, and $\gamma \in \mathbb{F} \setminus \{0\}$

Single-Error-Correcting Codes

Theorem: A linear code with parity-check matrix H can correct all single-error patterns iff

- the columns of H are all non-zero; and
- no column is a (non-zero) multiple of any other column

Single-Error-Correcting Codes

Theorem: A linear code with parity-check matrix H can correct all single-error patterns iff

- the columns of H are all non-zero; and
- no column is a (non-zero) multiple of any other column

In particular, a binary linear code is single-error-correcting iff it has a parity-check matrix with all columns distinct and non-zero.

- The $[7, 4, 3]$ binary Hamming code illustrates this idea.

Binary Hamming Codes

Let $r \geq 1$ be an integer.

The binary Hamming code $\mathcal{H}_r$ is the binary linear code specified by an $r \times n$ parity-check matrix whose columns are all the distinct, non-zero binary r-tuples.

- Since there are $2^r - 1$ distinct, non-zero binary r-tuples, we have $n = 2^r - 1$.
- One way of specifying an $r \times (2^r - 1)$ parity-check matrix is

$$H = [\mathbf{h}_1 \ \mathbf{h}_2 \ \cdots \ \mathbf{h}_{2^r-1}],$$

  where $\mathbf{h}_i$ is the r-bit binary representation of i.

Binary Hamming Codes

Let $r \geq 1$ be an integer.

The binary Hamming code $\mathcal{H}_r$ is the binary linear code specified by an $r \times n$ parity-check matrix whose columns are all the distinct, non-zero binary r-tuples.

- Since there are $2^r - 1$ distinct, non-zero binary r-tuples, we have $n = 2^r - 1$.
- One way of specifying an $r \times (2^r - 1)$ parity-check matrix is

$$H = [\mathbf{h}_1 \ \mathbf{h}_2 \ \cdots \ \mathbf{h}_{2^r-1}],$$

  where $\mathbf{h}_i$ is the r-bit binary representation of i.

- $\mathrm{rank}(H) = r \implies \dim(\mathcal{H}_r) = n - r = 2^r - 1 - r$.
- all columns are distinct and non-zero, but some three columns sum to 0 $\implies d_{\min}(\mathcal{H}_r) = 3$.
- Thus, $\mathcal{H}_r$ is an $[2^r - 1,\ 2^r - 1 - r,\ 3]$ binary linear code.

Hamming Codes are Perfect Codes

$\mathcal{H}_r$ is an $[2^r - 1,\ 2^r - 1 - r,\ 3]$ binary linear code.

$\mathcal{H}_r$ is a perfect 1-error-correcting binary linear code:

- LHS of Hamming bound: $\binom{n}{0} + \binom{n}{1} = 1 + (2^r - 1) = 2^r$
- RHS of Hamming bound: $2^{n-k} = 2^r$

q-ary Hamming Codes

Let $\mathbb{F}_q$ be a finite field, and $r \geq 1$ an integer.

- From each one-dimensional subspace of ${\mathbb{F}_q}^r$, pick exactly one non-zero vector.
- Doing this yields N vectors $\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N$, which we take to be the columns on an $r \times N$ matrix H.
- The q-ary Hamming code $\mathcal{H}_{r,q}$ is equal to $\mathrm{nullspace}_{\mathbb{F}_q}(H)$.

q-ary Hamming Codes

Let $\mathbb{F}_q$ be a finite field, and $r \geq 1$ an integer.

- From each one-dimensional subspace of ${\mathbb{F}_q}^r$, pick exactly one non-zero vector.
- Doing this yields N vectors $\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N$, which we take to be the columns on an $r \times N$ matrix H.
- The q-ary Hamming code $\mathcal{H}_{r,q}$ is equal to $\text{nullspace}_{\mathbb{F}_q}(H)$.
- $\text{rank}_{\mathbb{F}_q}(H) = r$, so $\dim_{\mathbb{F}_q}(\mathcal{H}_{r,q}) = N - r$.
- By construction, distinct columns of H span distinct 1-D subspaces, so no column is a multiple of another.

  Thus, $\mathcal{H}_{r,q}$ is single-error-correcting $\implies d_{\min} \geq 3$.
- Take any two columns $\mathbf{h}_i$ and $\mathbf{h}_j$ $(i \neq j)$ of H; their sum must lie in some 1-D subspace, so $\mathbf{h}_i + \mathbf{h}_j \in \text{span}(\mathbf{h}_k)$ for some k.

  Thus, there exist three lin. dep. cols in $H \implies d_{\min} = 3$.

Determining N

N is the number of distinct 1-D subspaces of ${\mathbb{F}_q}^r$.

- Each 1-D subspace contains $q - 1$ non-zero vectors.
- Any pair of distinct 1-D subspaces has only $\mathbf{0}$ in the intersection.
- Hence,

$$\begin{aligned} N \cdot (q-1) &= \text{no. of non-zero vectors in } {\mathbb{F}_q}^r \\ &= q^r - 1 \end{aligned}$$

Thus, $N = (q^r - 1)/(q - 1)$.

Determining N

N is the number of distinct 1-D subspaces of ${\mathbb{F}_q}^r$.

- Each 1-D subspace contains $q-1$ non-zero vectors.
- Any pair of distinct 1-D subspaces has only $\mathbf{0}$ in the intersection.
- Hence,

$$\begin{aligned} N \cdot (q-1) &= \text{no. of non-zero vectors in } {\mathbb{F}_q}^r \\ &= q^r - 1 \end{aligned}$$

Thus, $N = (q^r - 1)/(q-1)$.

In summary, $\mathcal{H}_r$ is a $\left[\frac{q^r-1}{q-1},\ \frac{q^r-1}{q-1} - r,\ 3\right]$ linear code over $\mathbb{F}_q$.

- It is easy to verify that it is a perfect single-error-correcting linear code over $\mathbb{F}_q$.

Other Perfect Linear Codes

There are only two other perfect linear codes:

- a $[23, 12, 7]$ triple-error-correcting code over $\mathbb{F}_2$
- an $[11, 6, 5]$ double-error-correcting code over $\mathbb{F}_3$

These are known as Golay codes.

The fact that there are no other perfect codes was proved by Tietäväinen (1973), building on the work of van Lint (early 1970s).

What Next?

- Constructions of t-error-correcting codes for $t \geq 2$ requires the full machinery of finite fields.
- But before getting into that, we squeeze more out of elementary combinatorial and linear-algebraic arguments to explore the trade-offs between the code parameters n, k, d.
- This allows us to give meaningful answers to questions such as what is the "best" possible code for a given set of parameters.