Channel Code

Random code

- Only used for proof, not in practice.
- Only good for long block length.
- Difficult to decode, requiring exhaustive search in an exponentially large codebook

In practice

- Finite block length (for low latency system, eg. Auto drive)
- Simple decoder (relaxing as computing power increase)

Hamming Code.

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}_{3\times 7}.$$

→ parity check matrix

- block length: $n = 7$.
- H contains all set of non-zero binary vectors of length 3
- Null space of H: $\{ v : Hv = 0 \}$. $v \in \mathbb{R}^{7\times 1}$

  has dimension $4 = n - \text{rank}(H)$

These 2^4 codewords are

0000000	0100101	1000011	1100110
0001111	0101010	1001100	1101001
0010110	0110011	1010101	1110000
0011001	0111100	1011010	1111111

$Hc = 0$

↳ column vector

properties of 2^4 codewords.

Codeword

0000000	0100101	1000011	1100110
0001111	0101010	1001100	1101001
0010110	0110011	1010101	1110000
0011001	0111100	1011010	1111111

- Linear: any two codewords is also a codeword
- All codewords forms a linear subspace of dimension 4 in the vector space of length 7
- minimum weight: ① minimum # of 1's in any codeword, except all-0 codeword, e.g. 0011001 → 3
  ② no two columns of H ca add to 000
- minimum distance: minimum number of place in which two codewords differs.

Minimum Distance.

Codeword {

0000000	0100101	1000011	1100110
0001111	0101010	1001100	1101001
0010110	0110011	1010101	1110000
0011001	0111100	1011010	1111111

- Measure how far apart the codewords are.
- Determine how distinguishable the codewords will be at the output of channel.
- Aim to develop codes having large minimum distance
- For linear code, minimum weight = Minimum distance of 0000000 + any codeword with minimum weight

Decoder:

- Let

$$e_i = (00\cdots1\cdots00)$$

$\hookrightarrow$ i-th position

- Received vector if i-th position has error:

$$r = c + e_i \qquad r = c + e_i + e_j$$

- Check with H by

$$Hr = H(c + e_i) = Hc + He_i$$

e.g. $He_2 =$

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

And space

The $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ position has error.

info. bits → parity check bits

0000000	0100101	1000011	1100110
0001111	0101010	1001100	1101001
0010110	0110011	1010101	1110000
0011001	0111100	1011010	1111111

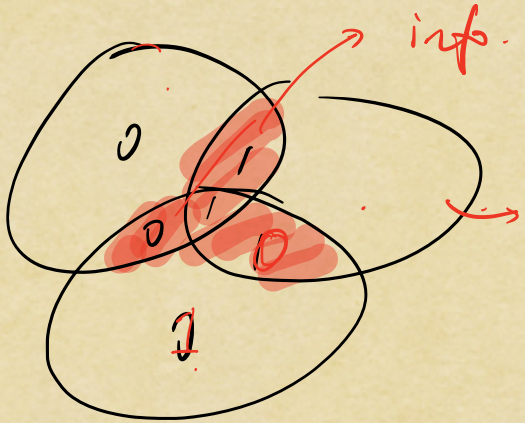
Systematic code

minimum distance

(n, k, d) or (n, k) code

n: codeword length

k: info. bits length



In general. Hamming code

$(n=2^l-1,\ k=2^l-l-1,\ d=3)$

$l=3.\ n=7.\ k=4.\ d=3.\quad R=\frac{4}{7}=\frac{60}{105}$

$l=4\quad n=15\quad k=15-4=11\quad d=3\quad R=\frac{11}{15}=\frac{77}{105}$

Linear Block code — 'Digital Communications' John G. Proakis

1) codeword $C_m = (C_{m1}, C_{m2}, \cdots C_{mn})$. n codeword length., $C_{mj} \in \mathbb{F}_{2^b}$

2) (n, k) code. k : length of information bits.

0000000	0100101	1000011	1100110
0001111	0101010	1001100	1101001
0010110	0110011	1010101	1110000
0011001	0111100	1011010	1111111

Generator Matrix: G $k \times n$ dimension

$C_m = u_m \cdot G \qquad 1 \le m \le 2^k$

u_m : length $1 \times k$ binary vector.

$G = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{pmatrix}$ $\{g_i\}$ are the codewords for message $(1, 0 \cdots 0), (0100\cdots0)$ $(0, 0, \cdots 1)$

Thus: $C_m = \sum_{i=1}^{k} u_{mi} \cdot g_i$ GF(2) mod 2

Property: ① $G = [I_k \mid P]$

I_k is $k \times k$ identity matrix, P is a $k \times (n-k)$ matrix.

This code is systematic code.

② In systematic code

③ C : (n,k) code $\xrightarrow{\text{Null space}}$ $(n, n-k)$ code $C^\perp$ (dual code of C)

The generation matrix of $C^\perp$ is call parity check matrix: H

$C_m H^t = 0 \quad \forall C_m \in C$

④ If $G = (I_k \mid P)$ then $H = [P^t \mid I_{n-k}]$

Hamming code

$G = \begin{bmatrix} 1000 & 011 \\ 0100 & 101 \\ 0010 & 110 \\ 0001 & 111 \end{bmatrix} = [I_k \mid P] \rightarrow H = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$

① $\sum_{i=0}^{M} \binom{M}{i} = ? = 2^m$

$n = 2^l - 1$ ← $H = \begin{bmatrix} 0 & \cdots & 0 \\ 1 & & 1 \end{bmatrix}_{l \times n}$

$k = 2^l - l - 1$ ← $k = n - \text{Rank}(H) = n - l$

1. Reed-Muller Code: (with flexible parameter)

Recall Hamming code: $n = 2^m - 1$, $k = 2^m - m - 1$, $d_{min} = 3$

RM code: $n = 2^m$, $k = \sum_{i=0}^{r} \binom{m}{i}$, $d_{min} = 2^{m-r}$

1) G matrix:

$$G = \begin{pmatrix} G_0 \\ \vdots \\ G_k \end{pmatrix}$$

$m \gg r$

$$R_{Hamming} = \frac{2^m - m - 1}{2^m - 1}$$

$$R_{RM} = \frac{\sum_{i=0}^{r} \binom{m}{i}}{2^m}$$

where

$G_0 = (1\,1\,1 \cdots 1)_{1 \times n}$

$$G_1 = \begin{pmatrix} 0 & 0 & \cdots & 1 \\ 0 & \vdots & & \vdots \\ \vdots & 0 & & 1 \\ 0 & 1 & & 1 \\ \downarrow & & & \\ 0 & 1 & 2 & \cdots 2^m - 1 \end{pmatrix}_{m \times 2^m}$$

G_2 is a $\binom{m}{2} \times n$ matrix whose rows are obtained by bitwise multiplication of two rows of G_1

G_i: $2 < i \le r$, is $\binom{m}{r} \times n$ matrix, bitwise bitwise multiplication of r rows of G_2

EXAMPLE 7.3-2. The first-order Reed-Muller code with block length 8 is an (8, 4) code with generator matrix

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \quad (7.3\text{–}9)$$

This code can be obtained from a (7, 3) maximum-length code by adding one extra parity bit to make the overall weight of each codeword even. This code has a minimum distance of 4. The second-order Reed-Muller code with block length 8 has the generator matrix

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (7.3\text{–}10)$$

and has a minimum distance of 2.

2> Hadmard Codes.

$$M_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \qquad M_{2n} = \begin{bmatrix} M_n & M_n \\ M_n & \bar{M}_n \end{bmatrix}_{2n \times 2n}$$

$M_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$

$n = 2^m$

$\bar{M}_n$: 0s is replaced by 1 and vice versa.

Each row represent as a code, not systematic code, $d_{min} = \frac{n}{2}$

$n = 2^m$, $k = \log_2 2n = m+1$, $d_{min} = \frac{n}{2} = 2^{m-1}$

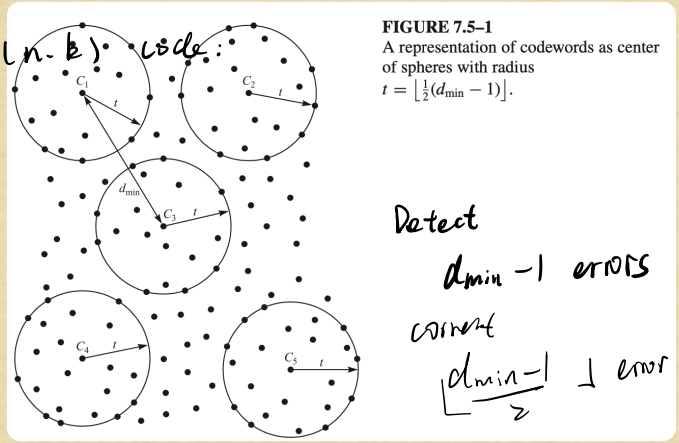


FIGURE 7.5–1
A representation of codewords as center of spheres with radius $t = \lfloor \frac{1}{2}(d_{min} - 1) \rfloor$.

(n, k) code:

Detect $d_{min} - 1$ errors

correct $\left\lfloor \frac{d_{min}-1}{2} \right\rfloor$ 1 error

2^k codewords is a point of n-dimension space; let t : the maximum # of det. error

최소 e 개 검출 $d \geq e+1$

t개 교정. $d > 2t+1$

LDPC （Low-density Parity-check）

- Robert Gallager, PhD Thesis, 1963
- A linear block code: Hc=0
- It can approche channel capacity
- Widely used in satellite communications, 5G, optic communications, etc
- Allow parallel decoding, and thus easy to implement.

Example:

Information bits: b=(c1,c2,c3)=(1 1 0)

LDPC code: C(s)=bG = (1 1 0 0 1 0)

$$C(s) = (c1 \quad c2 \quad c3) \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

- (c4, c5, c6) are parity check bits

$$\begin{aligned} c4 &= c1 \oplus c2 \\ c5 &= c2 \oplus c3 \\ c6 &= c1 \oplus c2 \oplus c3 \end{aligned} \rightarrow \begin{aligned} c1 \oplus c2 \oplus c4 &= 0 \\ c2 \oplus c3 \oplus c5 &= 0 \\ c1 \oplus c2 \oplus c3 \oplus c6 &= 0 \end{aligned} \rightarrow \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c1 \\ c2 \\ c3 \\ c4 \\ c5 \\ c6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

- Note: GH^T=0

$$c4 = c1 \oplus c2$$

$$c5 = c2 \oplus c3$$

$$c6 = c1 \oplus c2 \oplus c3$$

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c1 \\ c2 \\ c3 \\ c4 \\ c5 \\ c6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

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

Example:

- Each information bit will be checked by at least two parity check bits.

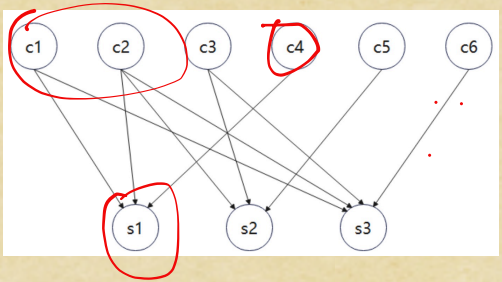
$$c4 = c1 \oplus c2$$

$$c6 = c1 \oplus c2 \oplus c3$$

$$c1 \oplus c2 \oplus c4 = 1 \neq 0$$

$$c2 \oplus c3 \oplus c5 = 0$$

$$c1 \oplus c2 \oplus c3 \oplus c6 = 1 \neq 0$$



- Decoding: Hard decoding and soft decoding
  - Hard Decoding: s=Hr, where r is the received code
  - Step 1: if s=0, correct decoding, other wise
  - Step 2: $f_n = \sum_{m=1}^{3} (s_m \cdot h_{mn}), n = 1, 2, \ldots, 6$ (

    if the f_n of r_n is larger than a threshold, then revert the bit

Cyclics Codes: C若为码字，则其所有循环位均为码字.

one an important class of linear block code.

codeword: $C=(c_{n-1}, c_{n-2}, \cdots c_1, c_0) = \quad c_i \in \{0,1\}$.

$$C(X) = c_{n-1}X^{n-1} + c_{n-2}X^{n-2} + \cdots + c_1X + c_0$$

$$X \cdot C(X) = c_{n-1}X^n + c_{n-2}X^{n-1} \cdots + c_0X$$

$$\frac{XC(X)}{X^n+1} = c_{n-1} + \frac{C^{(1)}(X)}{X^n+1}, \qquad \text{where } C^{(1)}(X) = c_{n-2}X^{n-1} + \cdots + c_0X + c_{n-1}$$

① Here $C^{(1)}(X)$ represents codeword: $C^{(1)} = (c_{n-2}, c_{n-3}, \cdots, c_0, c_{n-1})$, which is one unit shift of codeword $C = (c_{n-1}, c_{n-2}, \cdots c_0)$

codeword: $C=(c_{n-1}, c_{n-2}, \cdots c_1, c_0) = c_i \in \{0,1\}$.

$$C(X) = c_{n-1}X^{n-1} + c_{n-2}X^{n-2} + \cdots + c_0 X + c_0$$

$$X \cdot C(X) = c_{n-1}X^n + c_{n-2}X^{n-1} \cdots + c_0 X$$

$$\frac{XC(X)}{X^n+1} = c_{n-1} + \frac{C^{(1)}(X)}{X^n+1}, \quad \text{where } C^{(1)}(X) = c_{n-2}X^{n-1} + \cdots + c_0 X + c_{n-1}$$

② $C^{(1)}(X) = X \cdot C(X) \mod (X^n+1) \Leftrightarrow XC(X) = c_{n-1}(X^n+1) + C^{(1)}(X)$

$C^{(i)}(X) = X^i C(X) \mod (X^n+1) \Leftrightarrow X^i \cdot C(X) = Q(X)\cdot(X^n+1) + C^{(i)}(X)$

codeword 循环右移 i 位　　↓ 商 (quotient)

$C^{(i)}(X)$ 为余式. remainder polynomial

③ Generator polynomial (生成多项式) of 1) degree $n-k$. 2) is factor of (X^n+1)

i.e. $(X^n+1) = g(X)\cdot b(X)$

$$g(X) = X^{n-k} + g_{n-k-1}X^{n-k+1} + \cdots + g_1 X + 1,$$

message polynomial with message bits $(u_{k-1}, \cdots u_0)$

$$u(X) = u_{k-1}X^{k-1} + u_{k-2}X^{k-2} + \cdots + u_1 X + u_0 \Rightarrow 2^k \{u_m(X)\}$$

$m = 1 \cdots 2^k$

degree $(u(x)\cdot g(x)) \le n-1$, $u(x)\cdot g(x)$ represents a codeword

$$c_m(x) = u_m(x)\cdot g(x) \quad m = 1,\cdots 2^k$$

(#) Enc:

Given a generator polynomial $g(x)$, and for each message $m \in [2^k]$

$c_m(x) = u_m(x)\cdot g(x)$ → its coefficients construct a codeword.

Check: given any $c(c_{n-1}, c_{n-2}, \cdots c_0)$, see 计算其循环移位是否为一个码字

$c(x) = u(x)\cdot g(x)$

$c^{(1)}(x) = x\cdot c(x) + c_{n-1}(x^n+1)$, since $g(x)$ 整除 x^n+1 和 $c(x)$, we can write

$c^{(1)}(x) = x(u(x)g(x) + c_{n-1}g(x)\cdot b(x)) = \underbrace{(xu(x) + c_{n-1}\cdot b(x))}_{u_1(x)}\cdot g(x)$ $c^{(1)}(x)$ 也可由 $u_m(x)\,g(x)$ 表示

⑤ How to find $g(x)$?

degree $(g(x)) = n-k$, and $\frac{g(x)}{x^n+1} = h(x)$, $g(x)$ 为 x^n+1 的因式.

eg $x^7+1 = (x+1)(x^3+x^2+1)(x^3+x+1)$. 记 $g(x) = x^3+x^2+1$ 或 $g(x) = x^3+x+1$

TABLE 7.9–2
The (7, 4) Cyclic Code with Generator Polynomial
$g_1(X) = X^3 + X^2 + 1$

Information Bits				Codewords						
X^3	X^2	X^1	X^0	X^6	X^5	X^4	X^3	X^2	X^1	X^0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	1	0	1
0	0	1	0	0	0	1	1	0	1	0
0	0	1	1	0	0	1	0	1	1	1
0	1	0	0	0	1	1	0	1	0	0
0	1	0	1	0	1	1	1	0	0	1
0	1	1	0	0	1	0	1	1	1	0
0	1	1	1	0	1	0	0	0	1	1
1	0	0	0	1	1	0	1	0	0	0
1	0	0	1	1	1	0	0	1	0	1
1	0	1	0	1	1	1	0	0	1	0
1	0	1	1	1	1	1	1	1	1	1
1	1	0	0	1	0	1	1	1	0	0
1	1	0	1	1	0	1	0	0	0	1
1	1	1	0	1	0	0	0	1	1	0
1	1	1	1	1	0	0	1	0	1	1

⑤ Decoder: $x^n+1 = g(x) \cdot h(x)$

$h(x)$ is parity check polynomial 奇偶校验多项式.
or reciprocal

BCH code: 高效译码. n 中小值常用

$n = 2^m - 1$, $n - k \le mt$, $d_{min} \ge 2t+1$ $\frac{t}{n}$

$$g(X) = \text{LCM}\{\phi_\alpha(X), \phi_{\alpha^3}(X), \phi_{\alpha^5}(X), \ldots, \phi_{\alpha^{2t-1}}(X)\} \quad (7.10\text{–}3)$$

LCM: 最小公倍式. $\alpha, \alpha^3, \alpha^5 \in GF(2^m)$

$c(\alpha^i) = 0 \quad 1 \le i \le 2t$ monic

$\phi_\beta(X)$: β 的最小多项式. 即 满足 $\varphi_\beta(\beta)=0$ 的最低次首一多项式

minimal polynomial $\Rightarrow \phi_\beta(X) = \prod_{i=0}^{l-1} (X+\beta^{2^i})$ l s.t. $\beta^{2^l-1}=1$ 求解的最小整数

eg: $\beta = \alpha^5$, $\Rightarrow l=2$ $l=2$

$\varphi_\beta(X) = (X+\beta)(X+\beta^2) = (X+\alpha^5)(X+\alpha^{10})$

$= X^2 + (\alpha^5+\alpha^{10})X + \alpha^{15} = X^2 + X + 1$

LDPC （Low-density Parity-check

Reed-Solomon code:

$$g(x) = (x+a)(x+a^2)\cdots(x+a^{2t})$$

$$= x^{2t} + g_{2t-1}x^{2t-1} \cdots + g_1 x + g_0$$

$g_i \in GF(2^m)$

$n = 2^{m-1}$, $n-k = 2t$, $d_{min} = 2t+1 = n-k+1$

数据存储系统