

Linear Block Codes

Convolutional Codes:

How to make communication reliable?

Channel coding

Code rate:

$$\text{Defined by } R_c = \frac{k}{n}$$

Two types of channel codes

- Block codes
- Convolutional code

Block codes

- Binary seq. of length k mapped into seq. of length n

How do we send the n bits

via BPSK, QPSK, FSK

(Do we have to send the n bits together?)

$$n > k \Rightarrow R_c < 1.$$

represents the info bits sent in transmission
of a binary symbol over channel
(Given a seq. of k bits mapped into a
seq. of n bits, How does the entropy of the
codewords differ?)

(Do we have to send the n bits together?)

- Block codes are memoryless;

- Each set of k bits is independent

from the next sequence of k bits.

Sequence of codewords indep. of each other.

General Properties of Linear Block Codes

Why "Linear" block codes
linearly guarantees easy implementation

A binary block code C consists of a set of M

vectors of length n

$$C = \begin{bmatrix} \text{C}_{11} & \text{C}_{12} & \dots & \text{C}_{1n} \\ \text{C}_{21} & \text{C}_{22} & \dots & \text{C}_{2n} \\ \vdots & & & \\ \text{C}_{M1} & \text{C}_{M2} & \dots & \text{C}_{Mn} \end{bmatrix}$$

$$2^n \quad (\text{n bits})$$

How many possible codewords

$$2^k \quad (\text{k bits})$$

We choose only 2^k of them

(The all zero vector is a codeword — Why?)

$c_1 + c_2$ is a codeword

For a linear block code,
for any two code words c_1, c_2

$c_1 + c_2$ is a codeword

A block of ~~k~~ info bits is mapped into
a codeword of length n selected from the set
of 2^k codewords

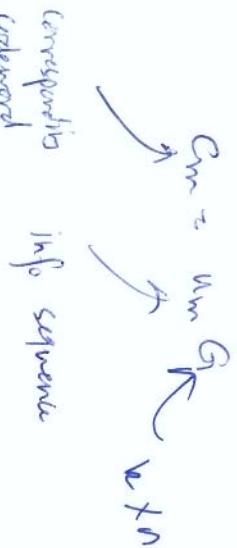
This is called an (n, k) block code with

$$\text{rate } R_c = \frac{k}{n}.$$

Generator & Parity Check Matrices

mapping from k bits to n bits can be represented by a $k \times n$ "Generator matrix" G

$$1 \leq m \leq 2^k$$



$$G = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{bmatrix}$$

\mathbf{g}_k is a codeword - why?

What info sequence "generates" it?

Thus, the code word corresponding to the

$$\text{info seq. } u_m = (u_{m1}, \dots, u_{mk}) \text{ is}$$

$$c_m = [u_{m1} \dots u_{mk}] G$$

$$= \sum_{i=1}^k u_{mi} g_i$$

ie info seq.

(k)

\Rightarrow codewords is the set of lin combination of rows of G

\Rightarrow Codewords is the row space of G

Systematic code

If G has the following structure size $k \times n-k$

$$G = \begin{bmatrix} I_k & P \end{bmatrix}$$

identifies matrix P

Resulting lin. block code is systematic

$$C_m = [u_{m1}, u_{m2}, \dots, u_{mk}] G$$

$$= [u_{m1}, u_{m2}, \dots, u_{mk}] [I_k | P]$$

$$= \underbrace{[u_{m1}, u_{m2}, \dots, u_{mk}]}_m \underbrace{[I_{k-m} | P]}_{\substack{\text{Parity check bits} \\ (k-m)}}$$

provides redundancy
against errors.

Rows of G are codewords

$$\Rightarrow G^t H^t = 0$$

Any linear block code has a systematic equivalent i.e. pub

$$G = [I_k \mid P]$$

by elementary row & column operations

Codewords are of dimension k in n -dimensional space

\Rightarrow orthogonal complement is of dim. $(n-k)$ in n -dim. space

matrix

Let H be corresponding

(of $\dim n-k \times n$)

now $n \times n$

Then for any codeword

$$C H^t = 0$$

Other systematic codes
~~or~~
 $G = [I_k \mid P]$

$$H = \begin{bmatrix} \bullet^{pt} & I_{n-k} \\ \hline \text{n-k} \times n-k \end{bmatrix}$$

Check that

Ex

$$G = [I_4 \mid P] = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Let $u = (u_1, u_2, u_3, u_4)$ be the info

$$c_5 = u_1 + u_2 + u_3$$

$$c_6 = u_2 + u_3 + u_4$$

$$c_7 = u_3$$

$$c_8 = c_1 + c_2 + c_4$$

$$c_9 = u_4$$

Weight & Distance for Lin. Block Codes

→ weight of a codeword $w(c)$ is the no. of nonzero components of the codeword

$$w(0) = 0 \quad 0 \text{ is a codeword}$$

- Hamming distance bet. $c_1 \& c_2$ is no. of components at which $c_1 \& c_2$ differ

$$d(c_1, c_2) = w(c_1 - c_2)$$

$$w(c) = d(c, 0)$$

- The set of all distances from c ~~is $\{d(c, c)\}$~~ is the same as the set of weights of codewords

$$d(c, c_i + c) = w(c_i + c - c) \\ = w(c_i)$$

- Min distance

$$d_{min} = \min_{c \neq 0} d(c, 0)$$

$$w_{min} = \min_{c \neq 0} w(c)$$

$$= d_{min}$$

Relation bet. min weight & columns of parity check matrix

Necessary & sufficient condition for c to be a codeword is

$$c^T H = 0$$

choose c with min weight then

$$c^T H = 0$$

⇒ d_{min} represents the min no. of columns of H that are d_{min} dependent

Since there are no columns with weights less than d_{min} no fewer columns of H are d_{min} dependent

\Rightarrow column span of H has dimension $d_{min} - 1$.

Error Detection & Correction Capabilities of

Block Codes

- Once an error is detected, we can ask for retransmission

d_{\min} is min separation between a pair of codewords

$\Rightarrow d_{\min}$ errors can transform of the 2^k codewords

into another

When this happens, we have an undetected error

a code word

Each codeword is like the center of a sphere, with radius t

t represents the no. of errors that affect

The largest t such that two spheres with not intersect or become tangent

$$t = \left\lfloor \frac{1}{2} (d_{\min} - 1) \right\rfloor$$

If no. of errors is less than d_{\min} , it is not possible for the error to transform one codeword into another

Another way of seeing this is that

($d_{\min} - 1$ lin.
indep. columns)

Column span of H is $d_{\min} - 1$

\Rightarrow any error of weight $\leq d_{\min} - 1$ can not result

$$\text{in } e^H t = 0$$

Hard decision decoding of linear block codes

Min. distance decoding:

We can pursue soft decoding of linear block codes. However, we will not do that here. Instead, we assume that the analog samples are quantized.

Decoding is then done on the decoded bits. This results in loss in performance but reduces the computational complexity.

Decoder decides in favor of codeword c_i closest to y .

Min-dist. decoding is optimum : results in a min. probability of codeword error for the binary symmetric channel (BSC).

Another way to do it

$$y = c_i + e$$

Add y to all codewords g_j

$$y + g_j = c_i + g_j + e$$

The codeword g_j that result in the min weight of $(y + g_j)$ is the most probable codeword that must have been received.

So compute M errors

$$e_m = y + c_m$$

Choose c_m which e_m has min weight

Syndrome & Standard Array Decoding

Make use of Hard Decision Decoding:

Let \mathbf{c}_m be the transmitted codeword

y

be

the received codeword

$$y = \mathbf{c}_m + \mathbf{e} \text{ error binary vector}$$

There are a total 2^{n-k} error patterns

Let's calculate y^H

$$y^H = \mathbf{c}_m^H + \mathbf{e}^H$$

$$\Rightarrow \mathbf{s} = \mathbf{e}^H$$

\mathbf{s} = syndrome vector (of dim. $(n-k)$)
of the error pattern

error

\mathbf{s} is a characteristic of the error
pattern (syndrome)

If $\mathbf{s} = 0 \Rightarrow$ error pattern = code word
 \Rightarrow undetected error

Error remains undetected if \mathbf{t}_b is equal
one of the codewords

There are 2^{k-1} such patterns

There are a total 2^{n-k} error patterns

of which 2^{k-1} are undetected because they correspond to actual codewords. Patterns can be detected by comparing 2^{n-k} nonzero patterns with \mathbf{c}_n . Remaining 2^{n-k} patterns are detected.

$\mathbf{c}_1 = 0$	$\mathbf{c}_3 = \dots$	\mathbf{c}_{2^k}
$\mathbf{c}_{2+\mathbf{e}_1}$	$\mathbf{c}_{3+\mathbf{e}_2}$	$\dots \mathbf{c}_{2^k+\mathbf{e}_2}$

because there are only 2^{n-k} syndromes

\Rightarrow different error patterns result in the same syndrome

For ML decoding we are looking for the error pattern of least weight among all

Construct a decoding table Also: all error patterns

By construction: coset leader has lowest weight among all coset members.

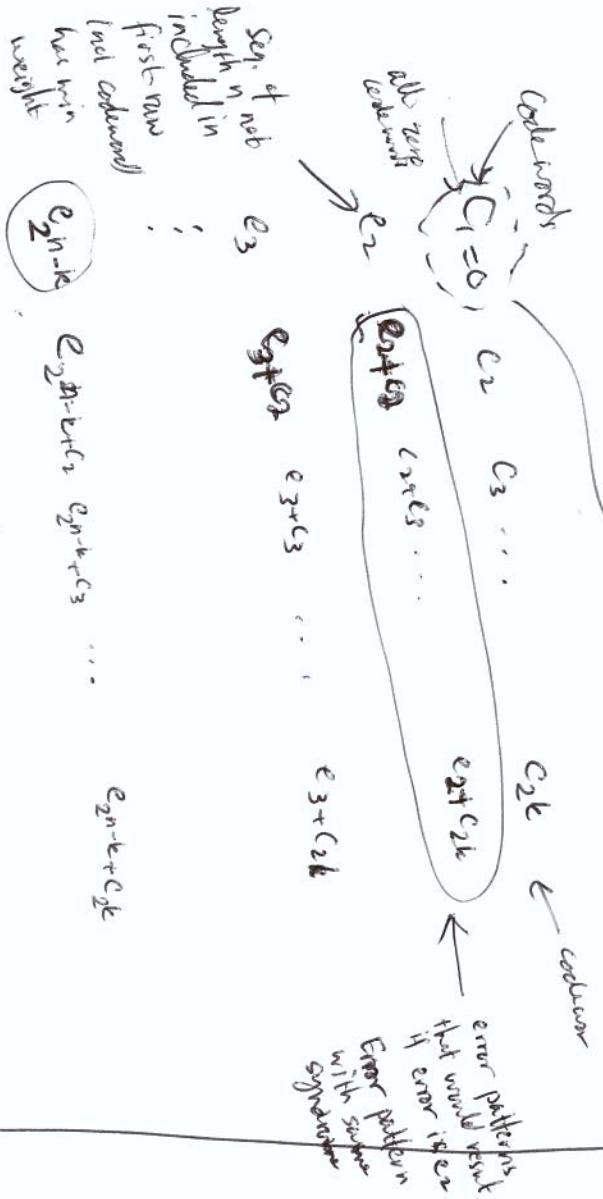


Table of size $2^{n-k} \times 2^k$

The table is called Standard Array

Each row consists of k possible received sequences

that would result from first error

Each row is a coset under left most column (or error pattern)

Cost: all possible received sequences resulting from an error pattern

All received seq.'s in same coset result in one-to-one correspondence bet. const & synd rows.

Same syndromes

Each syndromes

correspondence

To decode:

find the member with the lowest weight
(in this case coset leader) & add it to

received y .

Coset leaders have the only error patterns
that are correctable by \mathcal{C}

These are $2^{n-k}-1$ non-zero error patterns

2^{k-1} (codewords) not detectable

2^{n-k} are detectable

2^{n-k} are correctable

of which 2^{n-k-1} are correctable

Example:

Construct the standard array for the $(5,2)$ systematic code with the Generator matrix

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

$$2^k = 2^2 = 4$$

↖ ↘

00000	01011	11110
00001	01010	10111
00010	01001	10001
00100	01111	10101
01000	11101	10110

$n-k$

$2 =$

$2^3 = 8$

00000

00001

00010

00100

01000

01100

10000

10100

11000

11100

00011

01111

11111

00111

01011

11011

10011

11101

00101

01101

10101

11110

00010

01010

10001

11001

00110

01110

10110

11110

00001

01001

10001

11001

00100

01100

10100

11100

00000

01010

10001

11001

00110

01110

10110

11110

00001

01001

10001

11001

00100

01100

10100

11100

00000

01010

10001

11001

00110

01110

10110

11110

00001

01001

10001

11001

00100

01100

10100

11100

00000

01010

10001

11001

00110

01110

10110

11110

00001

01001

10001

11001

00100

01100

10100

11100

00000

01010

10001

11001

00110

01110

10110

11110

00001

01001

10001

11001

00100

01100

10100

11100

00000

01010

10001

11001

00110

01110

10110

11110

00001

01001

10001

11001

00100

01100

10100

11100

00000

01010

10001

11001

00110

01110

10110

11110

00001

01001

10001

11001

00100

01100

10100

11100

00000

01010

10001

11001

00110

01110

10110

11110

00001

01001

10001

11001

00100

01100

10100

11100

00000

01010

10001

11001

00110

01110

10110

11110

00001

01001

10001

11001

00100

01100

10100

11100

00000

01010

10001

11001

00110

01110

10110

11110

00001

01001

10001

11001

00100

01100

10100

11100

00000

01010

10001

11001

00110

01110

10110

11110

00001

01001

10001

11001

00100

01100

10100

11100

00000

01010

10001

11001

00110

01110

10110

11110

00001

01001

10001

11001

00100

01100

10100

11100

00000

01010

10001

11001

00110

01110

10110

11110

00001

01001

10001

11001

00100

01100

10100

11100

00000

01010

10001

11001

00110

01110

10110

11110

00001

01001

10001

11001

00100

01100

10100

11100

00000

01010

10001

11001

00110

01110

10110

11110

00001

01001

10001

11001

00100

01100

10100

11100

00000

01010

10001

11001

00110

01110

10110

11110

00001

01001

10001

11001

00100

01100

10100

11100

00000</