Coding Theory

↳ communication has 3 steps →

① ↳ encoding
↳ A source has a lot of redundancy
↳ to represent it in min k-bits

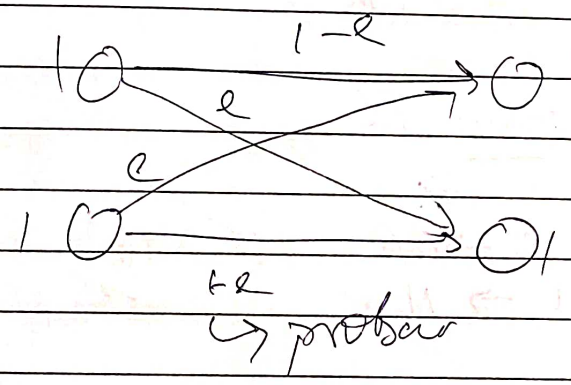
② Transmission over comm channel

③ Decoding at destination

I Theory → best & fundamental. lit on compression

our transmission channel

Transmission medium → channel.



Binary symmetric channel

Channel capacity → max rate we can communicate over a comm link

↳ low error probab possible if

information < channel capacity

hence good channel codes are possible

→ failed to specify its design

→ ECC

↓ (parity bits)

Adding redundant bits to message bits

↓

to detect & correct errors

eg → Repetition codes (Simple eg)

Rate = no. of informal bits / coded bits

= 1/2 → 1 IB / MB

2 CB

→ binary RC

0 → 00

1 → 11

Rate = 1/3

0 → 000

1 → 111

eg → 0 0 1 1 0 1 (IB)

coded bits using rate = 1/2

00 00 11 11 00 11

Received coded bits (single Error)

↳ 10 00 11 11 00 11

St detected as using repetition ← code both bits should be same but unsure if its 1/0

Received coded bits (double error):

↳ 11 00 11 11 00 11

↳ since its same could be undected

∴ Rate = 1/2 can detect single error

↳ can we correct it?

Rate = 1/2 repetition code can only detect single errors, but not correct them

→ inventions (prime factors & stuff)

→ Shor's algorithm

→ possible due to (abracadabra app.

→ Bipartite, challenge with

for Rate = 1/3

		000	000	111	111	000	111
Can detect S.E & D.E	S.E	100	000	"	"	"	"
	D.E	110	"	"	"	"	"

↳ and can correct S.E since majority of the bits → 0 its 0

Source encoder → source compression & coding
↳ represents source in num of bits.
↳ called data compression (Huffman coding)

its input & infor sequence

encryption
↳ make source bits transmission secure

Channel encoder
↳ ~~Correct~~ using additional redun. bits to detect & correct
Why not use inherent redundancy?
↳ we don't have control over them.

codeword → encoded sequence from channel encoder.

Modulation →

Channel
↳ physical transmission medium → wireless/wireless
⇒ corrupts signal, results in errors
eg (Binary erasure channel (BEC),
Binary symmetric channel (BSC)

Demodulation

Two types of ECC →
Block codes vs convolutional codes
↳ mapping of a block of k-bits into n-bit codeword

$u = (u_0, u_1, \ldots, u_{k-1})$ → Information bit sequence

n-bit codeword $V = (v_0, v_1, \ldots v_{n-1})$

encoder of block codes is memoryless
↳ output depends only on the current k-bits

code rate $(R) = k/n$
$n-k$ is no. of redundant bits (parity bits) added to each message to protect from errors

no. of codewords 2^k → each of length n

↳ this set is called binary codewords

(n, k)

eg → $k = 3, n = 6$

$R = \frac{1}{2}$

Message codewords

000 000

001 001

010 010

011 011

100 100

101 101

110 110

111 111

(1 1 1) 0 0 0 1 1 1

$u_0 u_1 u_2$

~~XOR~~

↳ modulo 2 addn

$v_0 \; v_1 \; v_2 \; v_3 \; v_4 \; v_5$

$v_5 = u_2$

$v_4 = u_1$

$v_3 = u_0$

$v_2 = u_0 + u_1$

$v_1 = u_0 + u_2$

$v_0 = u_1 + u_2$

Convolutional codes

↳ process in for sequence continuous while in block codes → it is done block by block

→ with memory of order of m

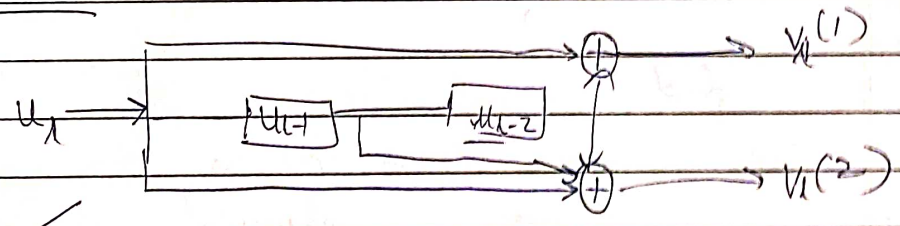
→ current output depends on current input & part input & output

m → no. of bits needed to give present output

(n, k, m) → convolutional code

→ Values of k, n of C.C << that of block codes

C.C



let k=1, n=2 & m=2

eg of → (2, 1, 2) → C.C

Outputs:

$$\begin{cases} v_l^{(1)} = u_l + u_{l-2} \\ v_l^{(2)} = u_l + u_{l-1} + u_{l-2} \end{cases}$$

Input : u_l

Decoder produces an estimate $\hat{v}$ of information sequence

Decoding rule → assignment of an estimate $\hat{v}$ to each of received sequence r

Av. prob of error → occurs when the decoded sequence $\hat{v} \neq v$ (received sequence)

$$P(E) = P(\hat{v} \neq v)$$
$$= \sum_r P(E|r) P(r)$$
$$= \sum_r P(\hat{v} \neq v/r) P(r)$$

find $\hat{v}$ → such that $P(\hat{v} \neq v/r)$ is minimized for each r

Minimizing $P(\hat{v} \neq v/r)$ is same as max $P(\hat{v} = v/r)$

For each $r_i \rightarrow$

$$P(v/r) = \frac{P(r/v)P(v)}{P(r)}$$ Bayes' rule

for every v & choose v that maximizes $P(v/r)$

$P(r)$ doesn't depend on v

$\Rightarrow \max P(v/r) = \max P(r/v)P(v)$

If all code words are equally likely to happen

$\Rightarrow \max P(v/r) = \max P(r/v)$

a maximum likelihood (ML) decoder chooses v that maximizes $P(r/v)$

Decoding strategies

↳ decoder produces

$r \rightarrow$ demodulated signal (input)

$\hat{u} \rightarrow$ ~~estimation of~~ info sequence (output)

~~estimation of~~ code sequence $\hat{v}$ (?)

& then use inverse encoder mapping to find $\hat{u}$ corresponding to $\hat{v}$.

Hamming distance bet^n 2 codewords

$d(r,v)$ → bet^n r & v
= no. of positions
for $r_i \neq v_i$

ML for BSC

$\log P(r/v) = -d(r,v) \log\left(\frac{p}{1-p}\right) + n \log(1-p)$

p → cond^n prob of error

eg → $r = 1\,1\,1\,0\,1\,1$
$v = 0\,1\,1\,1\,0\,1$
↳ 3

$d(r,v)$ → no. of bits got flipped
$n - d(r,v)$ → no. of bits not flipped

$P(r/v) = (1-p)^{(n-d(r,v))}\, p^{d(r,v)}$

FEC
↳ used in one-way system (transmitter & receiver)

Auto Repeat request (ARQ)
↳ used in 2 way system

Discrete
Date
Page

Generator Matrix & parity Check Matrix

(n,k) linear block code
k x n matrix → defines mapping of k-bits to n-bits codewords

G → rank k

gives mapping for IBut ✓

used for generating codewords

U → IB
V → EB
V = U G
(1 x n) (1 x k) (k x n)

How to find V?

↳ linear com of rows of G

gives a set of 2^k

it corresponding codeword → $V = UG = u_0 g_0 + u_1 g_1 + \cdots + u_{k-1} g_{k-1}$ (modulo-2)

$u = (u_0, u_1, u_2, \cdots u_{k-1})$

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix}$$

$$= \begin{pmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \\ \vdots & & & \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{pmatrix}$$

→ $V_1, V_2 \in$ codewords

$V_1 + V_2$

→ all zero vector

$0 = (0, 0, \ldots 0)$

↳ codeword in

valid every linear code.

→ (n, k) linear code

↳ k-d subspace of vector space V_n of all binary n-tuples

eg → $k = 3, n = 6$

(6, 3) linear block

Message u_0, u_1, u_2	Codeword $v_0, v_1, v_2, v_3, v_4, v_5$
0 0 0	0 0 0 0 0 0
1 0 0	0 1 1 1 0 0
0 1 0	1 0 1 0 1 0
1 1 0	1 1 0 1 1 0
0 0 1	1 1 0 0 0 1
1 0 1	1 0 1 1 0 1
0 1 1	0 1 1 0 1 1
1 1 1	0 0 0 1 1 1

How to find (G)?

$V_5 = u_0$

$V_4 = u_1$

$V_3 = u_2$

$V_2 = u_0 + u_1$

$V_1 = u_0 + u_2$

$V_0 = u_1 + u_2$

$$[v_0\ v_1\ v_2\ v_3\ v_4\ v_5] = [u_0\ u_1\ u_2] \begin{bmatrix} g_{0,0} & g_{0,1} & g_{0,2} & \cdots & g_{0,5} \\ g_{1,0} & & & & \\ \vdots & & & & \\ g_{2,0} & & & & \end{bmatrix}$$

$$G = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \end{bmatrix}$$

$v_0 = u_0 g_{0,0} + u_1 g_{1,0} + u_2 g_{2,0}$

$0 = 0\, g_{0,0} + 0\, g_{1,0} + 0\, g_{2,0}$

$0 = 1\, g_{0,0} + 0\, g_{1,0} + 0\, g_{2,0} \Rightarrow g_{0,0} = 0$

$1 = 0 + 0 + g_{2,0} \Rightarrow g_{2,0} = 1$

$$G = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \end{bmatrix}$$

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

codeword for message $u = (101)$ is

$$V = u \cdot G$$

$$= 1 \cdot (0\ 1\ 1\ 1\ 0\ 0) + 0 \cdot (1\ 0\ 1\ 0\ 1\ 0) + 1 \cdot (1\ 1\ 0\ 0\ 0\ 1)$$

$$= (0\ 1\ 1\ 1\ 0\ 0) + (0\ 0\ 0\ 0\ 0\ 0) + (1\ 1\ 0\ 0\ 0\ 1)$$

$$= (1\ 0\ 1\ 1\ 0\ 1)$$

(n, k) linear block code

↳ systematic if MB can be separated from PB directly

$$\begin{bmatrix} P_{0,0} & \cdots & P_{0,n-k-1} \\ P_{1,0} & & \\ \vdots & & \\ P_{k-1,0} & & \end{bmatrix}$$

$G = [P : I_k]$

or $[I_k : P']$

parity check eq'ns

$v_j = u_0 P_{0,j} + u_1 P_{1,j} + \cdots + u_{k-1} P_{k-1,j}$

$0 \le j \le n-k-1$

Message bits

$v_{n-k+i} = u_i \quad , 0 \le i \le k-1$

each parity bit v_j, $0 \le j \le n-k-1$, is a (modulo-2) sum of certain message bits

H (parity check matrix) $(n-k) \times n$ → properties

$VH^T = (0, 0, \cdots 0)$

eg → (7, 4) LBC

$G = \left[\begin{array}{ccc|cccc} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$

$= [P : I_n]$