Lec - 2 DC-2 - Kasun Sir

July - 05

Base: 2 → bits, e → nats, 10 → Hartley

* Measure of info : $I \propto \frac{1}{p} \Rightarrow \boxed{I = -\log_2(p) \text{ bits}}$

* Discrete – Memory less source :- $X \Rightarrow$ P. M. F

Symbols :- x_i

$\therefore I_X(x_i) = \log_b\left(\frac{1}{P(x_i)}\right)$ bits

($I_X(x_i)$ ← self information)

where, $b = 2, 10, e$.

Properties of self information :

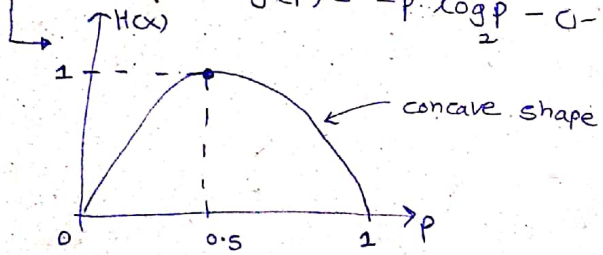
1. $I = 0$ for $p = 1$
2. $I \geq 0$ for $0 \leq p \leq 1$ ($0 \to \infty$, $1 \to 0$)
3. $I(x) > I(y)$ for $P(x) < P(y)$
4. $I(x,y) = I(x) + I(y)$ if x and y are statistically independant.

Entropy :- Average uncertainty resolved by observing the outcome of that source.

$$\boxed{H(x) = -\sum P(x) \cdot \log_2(P(x)) \text{ bits}}$$

e.g :- Binary source, $X = \begin{cases} 1, & \text{probability} = p \\ 0, & \text{probability} = (1-p) \end{cases}$

$\therefore H(x) = -\sum p \cdot \log(p) = -p \cdot \log_2 p - (1-p) \cdot \log_2(1-p)$ bits.



* → soure → $x_1, x_2, x_3, ----, x_N$

$p_1, p_2, p_3, -----, p_N$ are the probabilities.

$\therefore$ Max $H(x) = -\sum_{x_i} p_i \cdot \log(p_i)$ ⟶ objective function

constraint is, $\sum_{i=1}^{N} p_i = 1$

use, lagrange method ⇒ $f(p_1, p_2, --, p_N, \lambda) = -\sum p_i \cdot \log p_i + \lambda(p_1 + p_2 + --- + p_N - 1)$

$\frac{df}{dp_i} = -p_i \cdot \frac{1}{p_i} \cdot \log e - \log p_i + \lambda$

$\frac{df}{dp_i} = \lambda - \log e - \log p_i \Rightarrow \frac{df}{dp_i} = 0 \Rightarrow \boxed{p_i = 2^{\lambda}/e}$

2

$\therefore \sum p_i = 1 \Rightarrow \left(\frac{2\lambda}{e}\right) \cdot N = 1 \Rightarrow \boxed{p_i = \frac{1}{N}}$

$H(x) = -\sum \frac{1}{N} \cdot \log\left(\frac{1}{N}\right) = \log(N) > 0$:- Maximum Entropy

assume, $p_1 = 1, p_2 = p_3 = --- = p_N = 0$:- $H(x) = 0$

e.g :-

$X = \begin{cases} a, & p(a) = 1/2 \\ b, & p(b) = 1/4 \\ c, & p(c) = 1/8 \\ d, & p(d) = 1/8 \end{cases}$

$\Rightarrow H(x) = -\sum p_i \cdot \log(p_i)$

$= -\frac{1}{2} \cdot \log\left(\frac{1}{2}\right) - \frac{1}{4} \cdot \log\left(\frac{1}{4}\right) - \frac{1}{8} \cdot \log(1/8) - \frac{1}{8} \cdot \log(1/8)$

$= \frac{1}{2} + \frac{1}{2} + \frac{3}{8} + \frac{3}{8}$

$= 1 + \frac{3}{4}$

$\therefore H(x) = \frac{7}{4}$ bits.

Self → Information

Average → Entropy

(*)

$x \rightarrow x_i$
$y \rightarrow y_i$
⇒ Joint entropy

* Joint Entropy :→ $H(x,y) = -\sum_x \sum_y p(x,y) \cdot \log[p(x,y)]$

$H(x,y) = -E \log p(x,y)$

e.g :-

p	x = 0	x = 1	
y = 0	1/3	1/2	5/6
y = 1	1/6	0	1/6
	1/2	1/2	

$\Rightarrow H(x,y) = ?$

$H(x,y) = -\frac{1}{3} \cdot \log_2(1/3) - \frac{1}{6} \cdot \log_2(1/6) - \frac{1}{2} \cdot \log(0.5)$ bits

→ If x, y are independant random variable source then,

$H(x,y) = -\sum_x \sum_y p(x,y) \cdot \log[p(x,y)]$

$= -\sum_x \sum_y p(x) \cdot p(y) \cdot \log[p(x) \cdot p(y)]$

$= -\sum_x \sum_y p(x) \cdot p(y) [\log p(x) + \log p(y)]$

$= -\sum_x p(x) \cdot \sum_y p(y) \cdot [\log p(x) + \log p(y)]$

$= -\sum_x p(x) \cdot \log p(x) \sum_y p(y) + \left[-\sum_x p(x) \cdot \sum_y p(y) \cdot \log p(y)\right]$

[3]

$$= -\sum_x p(x) \cdot \log p(x) \cdot 1 - 1 \sum_y p(y) \cdot \log p(y)$$

$$\therefore H(X,Y) = H(X) + H(Y)$$

---

* **Conditional Entropy :→** $H(Y|X) = +\sum_x p(x) \cdot H(Y|X=x)$

$$H(Y|X) = -\sum_x p(x) \cdot \sum_y p(y|x) \cdot \log p(y|x)$$

$$= -\sum_x \sum_y p(x,y) \cdot \log p(y|x)$$

$$H(Y|X) = -E_{p(x,y)} \log p(Y|X) \; //$$

Note :- $H(Y|X) \neq H(X|Y)$

Prove; $\boxed{H(X) - H(X|Y) = H(Y) - H(Y|X)}$ → Mutual Information

* **Chain Rule of Entropy :-** $H(X,Y) = H(X) + H(Y|X)$

Proof 1 :- Rigorus method //

$$H(X,Y) = -\sum_x \sum_y p(x,y) \cdot \log p(x,y)$$

$$= -\sum_x \sum_y p(x,y) \cdot \log [p(x) \cdot p(y|x)]$$

$$= -\sum_x \sum_y p(x,y) \cdot [\log(p(x)) + \log(p(y|x))]$$

$$= -\sum_x p(x) \cdot \log(p(x)) - \sum_x \sum_y p(x,y) \cdot \log(p(y|x))$$

$$H(X,Y) = H(X) + H(Y|X) \; //$$

Proof 2 :- $p(x,y) = p(x) \cdot p(y|x)$

$$\Downarrow$$

$H(X,Y) = H(X) + H(Y|X)$; backwards of proof ①

e.g :- $H(X,Y|Z) = H(X|Z) + H(Y|X,Z)$

$$H(X|Z) = -\sum_x \sum_y p(x,z) \cdot \log p(x|z)$$

$H(Y|X,Z) = -\Sigma$

?

elements of Information Theory

PN

4

e.g:- $H(X,Y) = H(X) + H(Y|X)$

$H(X) = \frac{7}{4}$ bits,

$H(Y) = 2$ bits.

$H(Y|X) = H(X,Y) - H(X)$

$H(Y|X)$: $\frac{11}{8}$ bits; $H(X,Y)$: $\frac{25}{8}$ bits; $H(X)$: $\frac{7}{4}$ bits

* Relative Entropy :→ How dis-similar are (distance) two distributions

Kullback Leibler Distance (KL→ Distance)

$$D(p||q) = \sum_x p(x) \cdot \log\left(\frac{p(x)}{q(x)}\right)$$ → Inefficiency

$$D(q||p) = \sum_x q(x) \cdot \log\left(\frac{q(x)}{p(x)}\right)$$

example :- $\mathcal{X} = \{0, 1\}$ & p, q are distributions on $\mathcal{X}$

Let, $p(0) = 1-r$, $p(1) = r$.

Let, $q(0) = 1-s$, $q(1) = s$.

$$D(p||q) = (1-r) \cdot \log\left(\frac{1-r}{1-s}\right) + r \cdot \log\left(\frac{r}{s}\right),$$

$$D(q||p) = (1-s) \cdot \log\left(\frac{1-s}{1-r}\right) + s \cdot \log\left(\frac{s}{r}\right).$$

→ If $r = s$ then, $D(p||q) = D(q||p) = 0$

Note, In general $D(p||q) \neq D(q||p)$.

Lec-3 - July 8

Mutual information : $I(X;Y) = D(p(x,y) \| p(x)p(y))$

$= H(X) - H(X|Y)$

$= H(Y) - H(Y|X)$ //

$\therefore I(X;Y) = H(X) + H(Y) - H(X,Y)$ //

$\hookrightarrow I(X;Y) = I(Y;X)$ //

$\hookrightarrow I(X;X) = H(X) - \underbrace{H(X|X)}_{=0} = H(X)$ //

$\hookrightarrow I(X;Y) = 0$ when X & Y are independent //.

* If we design our system with high MI (Mutual Information) then, the system works with good capacity.

Source Coding $\Rightarrow$ Lossless $\Rightarrow$ Data Compression

$\hookrightarrow$ Zip

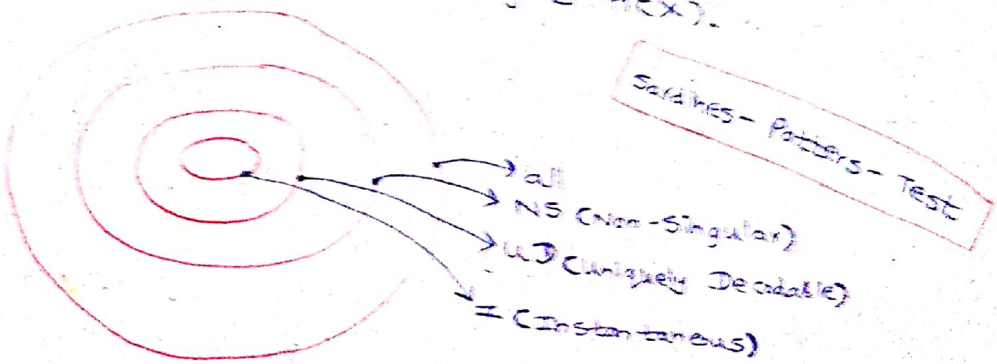
$\hookrightarrow$ based on dictionary coding

$C(x) \rightarrow$ Code word of x

$l(x) \rightarrow$ length of $C(x)$

$L(C) = \sum_{x \in X} p(x) \cdot l(x)$ bits / symbol.

* Our duty to check that, $L(C)$ is closely $\geq H(X)$.



Goal $\Rightarrow$ Design Instantaneous codes with minimum average length

$X \rightarrow x_1, x_2, x_3, ---, x_N$

$C(x_1) \rightarrow l_1 \rightarrow p_1$

$C(x_2) \rightarrow l_2 \rightarrow p_2$

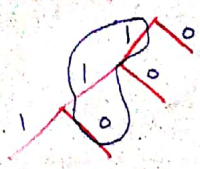
$C(x_3) \rightarrow l_3 \rightarrow p_3$

$C(x_N) \rightarrow l_N \rightarrow p_N$

Kraft's Inequality

* D-ary alphabet ; l_i :→ $\sum D^{-l_i} \leq 1 \Leftrightarrow$ Prefix code

$D = 2$:-
Binary tree //.



Proof ?

Optimal codes → optimization problem

Objective Function : $\sum p_i \cdot l_i = L(C)$ to be minimum

Subjective function : $\sum D^{-l_i} \leq 1$

l_i → Positive Integers

∴ Relax problem ; $\min \sum_i p_i l_i$

s.to ; $\sum D^{-l_i} = 1$

use lagrangian ,

$$J = \sum p_i \cdot l_i + \lambda \left(\sum D^{-l_i} - 1 \right)$$

$$\frac{dJ}{dl_i} = p_i - \lambda D^{-l_i} \cdot \log_e D = 0$$

$$\log_e D = \frac{p_i}{\lambda D^{-l_i}} \Rightarrow \therefore \boxed{D^{-l_i} = \frac{p_i}{\lambda \cdot \log_e D}}$$

$$\boxed{D^{-l_i} = p_i}$$

$$\sum D^{-l_i} = \sum \left\{ \frac{p_i}{\lambda \cdot \log_e D} \right\}$$

$$1 = \frac{1}{\lambda \cdot \log_e D}$$

$$\therefore \lambda = \frac{1}{\log_e D} \rightsquigarrow l_i = ?$$

$l_i^* = -\log_D p_i$ → Self Information

∴ Optimal average code word length :

$$L^* = H_D(X) \; //$$

$$\lceil -\log_D p_i \rceil \geq -\log_D p_i \quad \rightarrow \quad L(C) \geq H_D(X)$$

$$-\left\lceil \log_D \frac{1}{p_i} \right\rceil \leq -\log_D p_i$$

when equals, D-adic codes //.

$$\sum_i D^{-\lceil \log_D \frac{1}{p_i} \rceil} \leq \sum_i D^{-\log_D p_i} = \sum_i p_i = 1$$

∴ Kraft's inequality is satisfied by ~~taking~~ ceiling function //.

Lec 4 - 3rd July 24

1) Huffman code → Text compression
→ Image compression

2) Shannon - Fanon - Elias code

3) Arithmetic code

$$l_i^* = \lceil -\log_D p_i \rceil$$

$-\log_D p_i$

$$H_D(X) \leq L < H_D(X) + 1$$

$x \leq \lceil x \rceil \leq x + 1$

Proof:- $\log_D\left(\frac{1}{p_i}\right) \leq \lceil \log_D\left(\frac{1}{p_i}\right) \rceil < \log_D\left(\frac{1}{p_i}\right) + 1$

$$\sum_i p_i \log_D \frac{1}{p_i} \leq \sum_i p_i \lceil \log_D\left(\frac{1}{p_i}\right) \rceil < \sum_i \left(p_i \log_D \frac{1}{p_i} + p_i\right)$$

$$H_D(X) \leq L < H_D(X) + 1 \quad //$$

(condition for optimal code)

1) Huffman code.

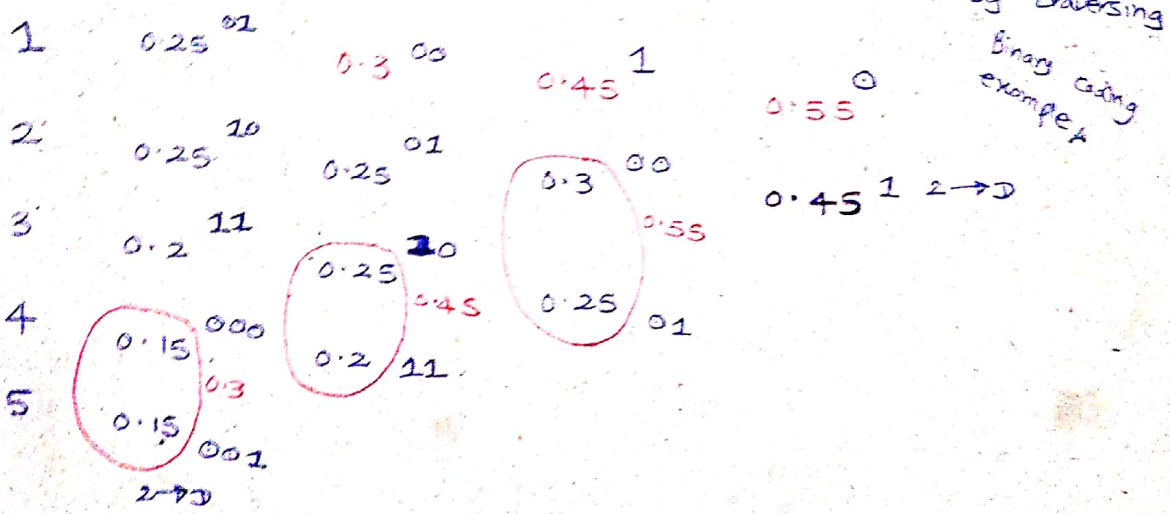
i) Arrange the codes in descending order

ii) Group last two probabilities and descend now

iii) Repeat step i) until getting D number of (2) probabilities

iv) assign 0 for up and 1 for down from last to front by traversing

Binary coding example



1 → 01 ; 2 → 10 , 3 → 11 ; 4 → 000 , 5 → 001

$$\therefore L(C) = \sum_i p_i l_i = \frac{1}{4}\times 2 + \frac{1}{4}\times 2 + \frac{1}{5}\times 2 + \frac{3}{20}\times 3 + \frac{3}{20}\times 3$$

$$= \frac{20 + 8 + 18}{20} = \frac{46}{20} = 2.3 \text{ bits/symbol}$$

Code efficiency $(\eta) = \frac{H_D(X)}{L}$,

Redundancy $(\gamma) = 1 - \eta = 1 - \frac{H_D(X)}{L} = \frac{L - H_D(X)}{L}$.

Huffman coding → $K(D-1) + D$ Eqns.

↑ Inefficient when skewed distribution

↑ Solution :- Sequence of length 2 ⇒ 2nd order extension of the source.

$M_1 M_1 \rightarrow 0.64$
$M_1 M_2 \rightarrow 0.15$
$M_2 M_1 \rightarrow 0.16$
$M_2 M_2 \rightarrow 0.04$

When $M_1 \rightarrow 0.8$, $M_2 \rightarrow 0.2$

$L = 1.56$ bits / 2 symbols $= 0.78$ bits/symbol

$\eta_{old} = \frac{0.72}{1} = 0.72 <$ (better) $\eta_{new} = \frac{0.72}{0.78} = 0.92$

→ Problems : Memory, computations.

$L < P_{max} + 0.086 + H_D(X)$

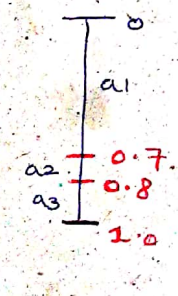
2> Arithmetic coding → Good sequence encoding technique

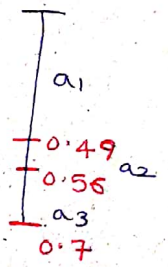
- ✓ Low memory
- ✓ Low computational complexity
- ✓ one shot code generation
- ✓ Prefix code with higher efficiency {$L \rightarrow H_D(X)$}

e.g :-

$P(X=1) = 0.7$
$P(X=2) = 0.8$
$P(X=3) = 0.2$

$F_X(1) = 0.7$
$F_X(2) = 0.8$
$F_X(3) = 1.0$





$a_1\ a_2\ a_3\ a_1\ a_3\ a_1\ a_2 \rightarrow$ Tag $\in [0, 1)$

Mid point

$T_X(a_i) = F_X(i-1) + \frac{1}{2} \cdot P(X=i)$

Lec 5 - Aug 24

e.g.:- 322

- l(322)
- T(322)
- u(322)

$$T^3(322) = \sum_{y<322} P(y) + \frac{1}{2}\cdot P(X=322)$$

$$F(32) = \underbrace{P(X=1) + P(X=2)}_{F(2)} + \underbrace{P(X=31)}_{P(X=3)\,P(X=1)} + \underbrace{P(X=32)}_{P(X=3)\cdot P(X=2)}$$

$$\therefore F(32) = F(2) + \underbrace{P(X=3)}_{F(3)-F(2)}\cdot[\underbrace{P(X=1)+P(X=2)}_{F(2)}]$$

$$\therefore F(32) = F(2)\cdot\{1 + F(3) - F(2)\} = F(2) + [F(3)-F(2)]\cdot F(2)$$

$$\hookrightarrow u^2 = l^1 + [u^1 - l^1]\cdot F(2)$$

$$l^2 = l^1 + [u^1 - l^1]\cdot F(1)$$

$$u^3 = l^2 + [u^2 - l^2]\cdot F(2)$$

$$l^3 = l^2 + [u^2 - l^2]\cdot F(1)$$

$$l^n = l^{n-1} + [u^{n-1} - l^{n-1}]\cdot F(x_n - 1)$$

$$u^n = l^{n-1} + [u^{n-1} - l^{n-1}]\cdot F(x_n)$$

s.t. $l^0 = 0,\ u^0 = 1$

$$\Rightarrow T^n(x_i) = \frac{u^n + l^n}{2}$$

e.g.:- $A = \{a_1, a_2, a_3\}$ $\Rightarrow$ $a_1\ a_3\ a_2\ a_1$ $\Rightarrow$ $T(1321) = ?$

$a_1 \to 0.8$, $a_2 \to 0.02$, $a_3 \to 0.18$

$l^0 = 0$, $u^0 = 1$

$F(0) = 0$,
$F(1) = 0.8$,
$F(2) = 0.82$,
$F(3) = 1$

$$l^1 = l^0 + [u^0 - l^0]\,F(0) = 0 + (1-0)\cdot 0 = 0,$$

$$u^1 = l^0 + [u^0 - l^0]\cdot F(1) = 0 + (1-0)\cdot 0.8 = 0.8.$$

$$l^2 = l^1 + [u^1 - l^1]\cdot F(2) =$$

$$u^2 = l^1 + [u^1 - l^1]\cdot F(3) =$$

$$l^3 = l^2 + [u^2 - l^2]\cdot F(1) =$$

$$u^3 = l^2 + [u^2 - l^2]\cdot F(2) =$$

$$l^4 = l^3 + [u^3 - l^3]\cdot F(0) =$$

$$\therefore T^4(1321) = \frac{l^4 + u^4}{2}$$

10

* When Sequence length ↑ ⇒ Tag value shrinks ⇒ Need high precision hardware

solution, use a D-ary function to change tag value to a D-ary string.

* Encoding : Sequence ⟶ Tag :- ✓

Decoding : Tag ⟶ Sequence :- ?

↳ Mimic the encoder in decoder (Reverse engineering)

↔ Tag → code ;

$0.8125 \times 2 = 1.625$ ↓

$0.625 \times 2 = 1.25$

$0.25 \times 2 = 0.5$

$0.5 \times 2 = 1.0$

$\therefore 0.8125_{10} = .0.1101_{2}$

Lec 6 - Aug 31

⇝ To avoid infinite length in codeword, truncating is suitable.

But, what is the truncating length ?

$$l(x) = \left\lceil \log \frac{1}{P(x)} \right\rceil + 1$$

Proof, Code is non-singular; ⟶ Tag is singular ⇒ code is singular

Code is prefix free,

Efficient.

① Non-singular ; $F(x-1) \le T(x) < F(x)$

$n = \left\lceil \log \frac{1}{P(x)} \right\rceil + 1$ $\quad \lfloor T(x) \rfloor_n \le T(x)$

Need to show that, $F(x-1) \le \lfloor T(x) \rfloor_n$

$$T(x) - \lfloor T(x) \rfloor_n < \frac{1}{2^n} = \frac{1}{2^{\lceil \log \frac{1}{P(x)} \rceil + 1}} < \frac{1}{2^{\log \frac{1}{P(x)} + 1}} = \frac{P(x)}{2}$$

$$\therefore T(x) - \lfloor T(x) \rfloor_n < \frac{P(x)}{2} = T(x) - F(x-1)$$

$$F(x-1) < \lfloor T(x) \rfloor_n$$

∴ Arithmetic coding is non-singular //.

Lec 7 - Sep 07

* Huffman & Arithmetic coding ~~(6)~~ Source P.M.F ~~Skewed~~
* Dictionary coding
  - Static D.C
  - Dynamic D.C → LZ77, LZ78, LZW

1) LZ77

<o, l, c>

- o: offset
- l: length of the symbol
- c: codeword for the next symbol

* Amount of bits transmitting in LZ77 per tuple =

$$\lceil \log_2 S \rceil + \lceil \log_2 W \rceil + \lceil \log_2 |A| \rceil$$

Issues : Limited view (past and future), 3-tuple

2) LZ78 → 2-tuple <i, c>

e.g :- b a w w a / b a w w a / b a w w a / b a w w a / b o o / b o o

ASCII

index	entry	output
1	b	<0, C(b)>
2	a	<0, C(a)>
3	w	<0, C(w)>
4	wa	<3, C(a)>
5	/	<0, C(/)>
6	ba	<1, C(a)>
7	ww	<3, C(w)>
8	a/	<2, C(/)>
9	baw	<6, C(w)>
10	wa/	<4, C(/)>
11	baww	<9, C(w)>
12	a/b	<8, C(b)>
13	o	<0, C(o)>
14	o/	<13, C(/)>

* Good when recurring pattern but bad for random occur

RTX 2080 GM 5 TX ↔ RTX

1984

3) LZW (Lempel - Ziv - Welch) <i>

* Dictionary
  - Static
    - characters
    - Strings of length
  - Dynamic

* used in GIF compress
* Example in wikipedia.

e.g:- TOBEORNOTTOBEORTOBEORNOT

{Lecture 7 - Sep 07 - Bomins →}

Summary - Source Coding

1) Basic Info theory
2) lossless compression → Huffman, Arithmetic, De
3) optimal codes

CHANNEL CODING

(Error Correction codes)

1. Variations from: block and convolutional codes
2. optimal hard-decision decoding of linear block codes using the Syndrome decoding method
3. Optimal hard-decision decoding of convolutional codes using the Viterbi algorithm

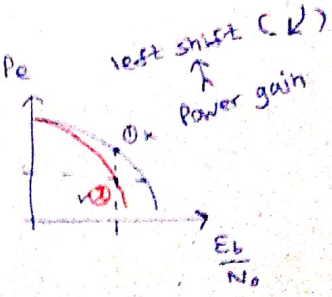
Why? → efficient, reliable communication system (cost effective)
- Hardware
- Power
- Bandwidth

FEC → Forward Error correction

For a given, $\frac{E_b}{N_0}$ → BER ↓

For a given, BER → $\frac{E_b}{N_0}$ ↓

Save power



* Error detection & error correction (when retransmission not ...)
  - ↓ ARQ
  - ↓ FEC → saves power

1) Parity bits — [bits | Parity]
→ error detection

1.1) odd parity — | 1 0 1 1 0 1 1 | → [1]

1.2) even parity → [0]

* Simple parity bits issue is that even number of bits are error then can't tell any whether any error in data.

2) Checksum → 1's complement addition — [Hex data | checksum]
→ Error detection

```
  1101
+ 1001
 10110
  ↳  1
  0111
```

* Hexadecimal digits 1 word → 4 hex digits (16 bits)

```
  F 2 0 4
+ F 3 2 1
1 E 5 2 5
  ↳     1
  E 5 2 6
```

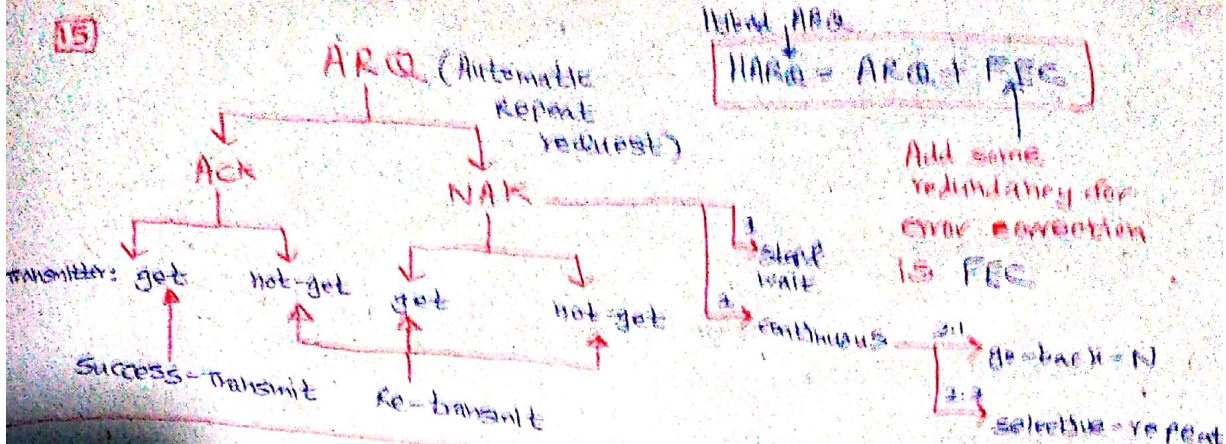
e.g.: 0 0 F E | C 5 2 3 | F D A 1 | D 6 8 A | A F 0 2

```
  0 0 F E
  C 5 2 3
  C 6 2 1
  F D A 1
1 C 3 C 2
        1
  C 3 C 3
  D 6 8 A
1 9 A 4 D
        1
  9 A 4 E
  A F 0 2
1 4 9 5 0
        1
  4 9 5 1
```

* Sending: do like this for payload
* Receiving: do like this for whole data (payload + check sum) → FFFF

FFFF − 4951 → B6AE (checksum)

15



Lec 8 - Sept 14

HARQ

Type 1

Data + ED + FEC

good channel → ✓

bad channel → HARQ

Type 2

Data + ED → Done

Feedback → Data + FEC

Chase combining

Incremental redundancy

Modulo-2 Arithmetic { XOR }

$0 + 0 = 0$

$1 + 0 = 1$

$0 + 1 = 1$

$1 + 1 = 0$

$0 \times 0 = 0$

$0 \times 1 = 0$

$1 \times 0 = 0$

$1 \times 1 = 0$

$0 \div 1 = 0$

$1 \div 1 = 0$

Subtraction = Addition

10111011 → Hamming weight = 6 { Number of ones in sequence }

1011001
1010110

Hamming distance = 4 { Number of different locations in 2 sequences }

∴ Hamming weight of addition of 2 sequence = Hamming distance

1) Block codes → block-by-block
→ Memoryless

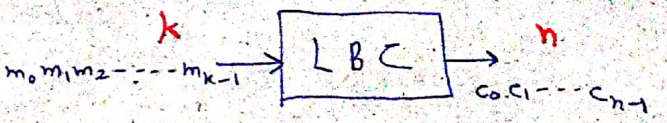
Subclass1
→ Linear - Block codes

Property: $c_i + c_j \rightarrow c_k$

Subclass1
→ Systematic LBC

* (n, k) — n: length of codeword; k: length of data block

$R_d = \left(\frac{k}{n}\right) \cdot R_s$ — R_d: channel data rate; R_s: information rate



$\Rightarrow$ Tx power ↑ B.W ↑

⇊

Ensure, improvement in BER

(n, k) → $n-k$ redundant bits

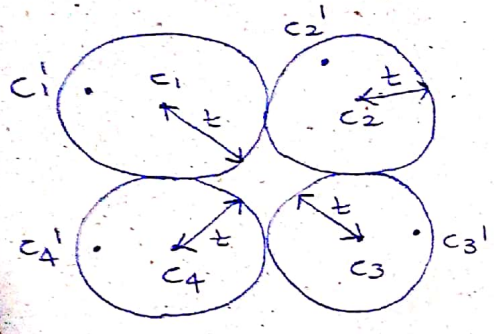
n: codewords; k: messages

$\Rightarrow 2^k$ codewords required

2^n codewords available

Hypercube → n-D space

$HD(c_j \text{ and } c_j') \leq t$



; Minimum HD between any two codewords $> 2t$

$= 2t+1$

Vector Notations

$\underline{m} = [m_0\ m_1 - - - - m_{k-1}]_{1\times k}$

$\underline{b} = [b_0\ b_1 - - - - b_{n-k-1}]_{1\times(n-k)}$

$\underline{c} = [c_0\ c_1 - - - - c_{n-1}]_{1\times n}$

$\underline{b} = \underline{m} \cdot \underline{P}$

$(1\times(n-k)) \quad (1\times k) \quad (k\times(n-k))$

; $\underline{P} = \begin{bmatrix} p_{00} & p_{01} & p_{02} & - - - & p_{0,n-k-1} \\ \vdots & & & & \\ p_{k-1,0} & p_{k-1,1} & - - - & - - - & p_{k-1,n-k-1} \end{bmatrix}$

$\underline{c} = [\underline{b} \mid \underline{m}]_{1\times n}$

$= [\underline{m}\cdot\underline{P} \mid \underline{m}]_{1\times n}$

$= \underline{m}_{1\times k} [\underline{P} \mid \underline{I}]_{k\times n}$ — $\underline{I}$: unit matrix

Define generator matrix; $[P \mid I]_{k\times n} = G_1$

$\underline{c} = \underline{m} \cdot G_1$

k rows must be linearly independant

Codebook ⇒ set of all codes

SLBC ← G_1 is fixed

17

$H \rightarrow$ Parity check matrix

$$H = [I_{n-k} \mid P^T]$$

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

$$H \cdot G^T = [I_{n-k} \mid P^T] \begin{bmatrix} P^T \\ I_k \end{bmatrix} = P^T + P^T = 0$$

Error detection & Error correction

$$C \cdot H^T = m \cdot G \cdot H^T = m \cdot 0 = 0$$

$m \rightarrow [G] \rightarrow c \rightarrow [H^T] \rightarrow 0$

k rows must be linearly not dependant

receive

$r = c + e$ ← error

↓ transmit

goal $\Rightarrow$ recover c from r //

Syndrome decoding mechanism

$\Rightarrow$ Syndrome: $\underset{1\times n-k}{S} = \underset{1\times n}{r} \cdot \underset{n\times n-k}{H^T}$

$$S = r \cdot H^T = (c+e) \cdot H^T = c \cdot H^T + e \cdot H^T = 0 + e \cdot H^T$$

$$S = e \cdot H^T$$

* properties of S

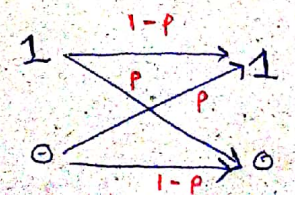
- contains information about e
- depends only on e

$$S = e \cdot H^T$$

$$r = c + e \Rightarrow c = r - e = r + e$$

$\Rightarrow$ Maximum-Likelihood rule

2^k possibles



$$P(r \mid c_i) = p^d \cdot (1-p)^{n-d} = (1-p)^n \left(\frac{p}{(1-p)}\right)^d$$

$p < 0.5 \Rightarrow$ reasonable communication channel

17

$H \longrightarrow$ Parity check matrix

$$H = [I_{n-k} \mid P^T]$$

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

$$H \cdot G^T = [I_{n-k} \mid P^T] \cdot \begin{bmatrix} P^T \\ I_k \end{bmatrix} = P^T + P^T = 0$$

Error detection & Error correction

$$c \cdot H^T = m \cdot G \cdot H^T = m \cdot 0 = 0$$

$$m \longrightarrow \boxed{G} \longrightarrow c \longrightarrow \boxed{H^T} \longrightarrow 0$$

k rows must be linearly not dependant

$r = c + e$ (receive, error, transmit)

goal $\Rightarrow$ recover c from r //

Syndrome decoding mechanism

↔ Syndrome: $S = r \cdot H^T$

$1 \times n-k \quad 1 \times n \quad n \times n-k$

$$S = r \cdot H^T = (c+e) \cdot H^T = c \cdot H^T + e \cdot H^T = 0 + e \cdot H^T$$

$$S = e \cdot H^T$$

* properties of S

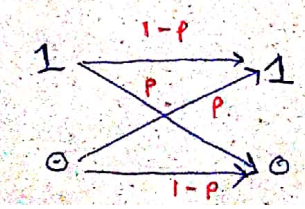
- contains information about e
- depends only on e

$$S = e \cdot H^T$$

$$r = c + e \Rightarrow c = r - e = r + e$$

$\Rightarrow$ Maximum-Likelihood rule

(2^k possibles)



$$P(r \mid c_i) = p^d \cdot (1-p)^{n-d} = (1-p)^n \left(\frac{p}{(1-p)}\right)^d$$

$p < 0.5 \Rightarrow$ reasonable communication channel

$P(\underline{r} \mid \underline{c}_i)$ ⇒ pick the $\underline{c}_i$ which has smallest HD from $\underline{r}$

⇒ Pick the $\underline{e}$ which has the smallest HW

$S = r \cdot H^T$ ⇒ 2^k possible $\underline{e}$ ⇒ choose $\underline{e}$ which has smallest H.W.

Coset leader

MAP ← ML

Gaussian noise

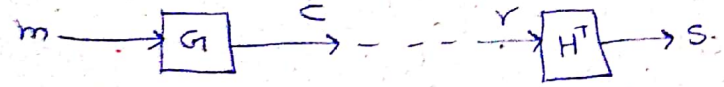
Encoding cyclic BC//

$$C(x) = \text{reminder}\left(\frac{x^{n-k} \cdot m(x)}{g(x)}\right) + x^{n-k} \cdot m(x)$$

(n, k)
g(x)
m(x)

necessary condition :- $x^n + 1$ has a factor $g(x)$

LBC//



* $c = m \cdot G$
* $S = r \cdot H^T \Rightarrow e \cdot H^T$
* $r = c + e$ / $c = r + e$
* $c \cdot H^T = 0$
* $G \cdot H^T = 0$
* $G_{k \times n} = [I_k \mid P_{k \times n-k}] \Rightarrow H^T_{n \times n-k} = \begin{bmatrix} P_{k \times n-k} \\ I_{n-k} \end{bmatrix}$

$\gamma = n \times \sum_{t=1}$ Hamming code

$m = n - x$

↳ Memory for LBC = $2^{n-k} \cdot (2n-k)$

LBC ; Error correcting capability $t = \sum_{j=0} \binom{n}{j}$ ⇒ $x \to m$, $t \to 2$

$0 \to x$ number of bit errors can be corrected

Lec - 12 → Oct - 22

page 31

- Encoder ⇒ FSM ⇒ Trellis

$$C(D) = m(D) \cdot g(D)$$

State diagram → tedious

Trellis → helpful than state diagram
Trellis → useful in decoding

e.g :-

input ↓ memory

$g^{(1)} \rightarrow 110$

$g^{(2)} \rightarrow 101$ ⇒

$g^{(3)} \rightarrow 100$

$g^{(4)} \rightarrow 111$

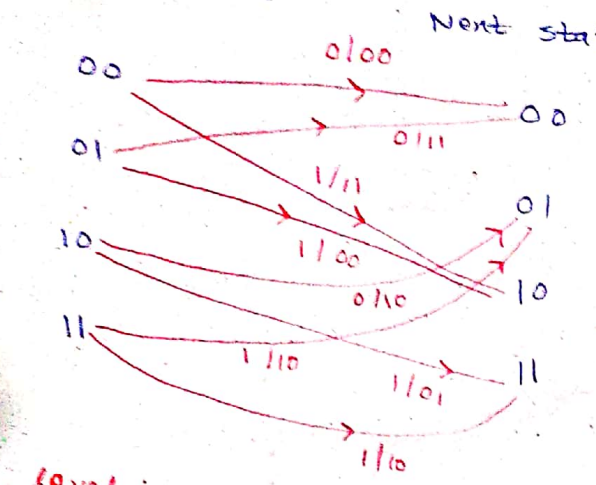
* Code rate $= \frac{1}{4}$ $\left\{\frac{1}{\text{Number of generators}}\right\}$

* Constraint length $= K = M + 1 = 3$

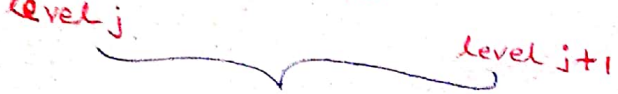
{length of a generator}

* Number of states in, state diagram $= 4$

$(2^{3-1} = 2^2 = 4 \leftarrow 2^{K-1})$



* ↔ Impulse response :- Output of the path when input is 1.



* L inputs ⇒ L+M levels in trellis ; $0 \rightarrow M$, $L \rightarrow L+M$ } May not attain all States

$M \rightarrow L$ } Attain all States

; start ≡ 00,
Finish ≡ 00.

Decoding of Convolutional codes

$m \rightarrow c \rightarrow r \rightarrow c \rightarrow m$; $r \rightarrow c \rightarrow m$ (or)

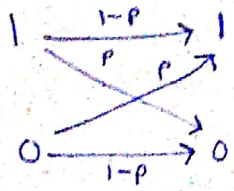
likelihood function : $P(\underline{r} | \underline{c})$

* Pick $\hat{c}$ to maximize the likelihood function.

log-likelihood function $\Rightarrow \log P(\underline{r} | \underline{c})$

$$\log P(\underline{r} | \underline{c}) = \log \left[\prod_{i=1}^{N} P(\underline{r}_i | \underline{c}_i) \right]$$

$$= \sum_{i=1}^{N} \log P(\underline{r}_i | \underline{c}_i)$$



If channel caused d errors

$\Rightarrow d$ = H.D between $\underline{r}$ and $\underline{c}$

$$= \log \left[p^{d} (1-p)^{N-d} \right]$$

$$= d \log p + (N-d) \cdot \log(1-p)$$

$$= d \left[\log p - \log(1-p) \right] + N \cdot \log(1-p)$$

$$= d \cdot \log\left(\frac{p}{1-p}\right) + N \cdot \log(1-p)$$

($d \cdot \log\left(\frac{p}{1-p}\right)$: variable; $N \cdot \log(1-p)$: constant)

$p < 0.5 \Rightarrow \frac{p}{1-p} < 1$

maximized when d is minimal

Pick $\hat{c}$ which has smallest H.D with received vector $(\underline{r})$

* Block codes $\Rightarrow$ Syndrome decoding $\Leftarrow$ Search space is limited
* Large N $\Rightarrow$ Very complexive search
* Sequence detection problem
* Viterbi algorithm (Dijkstra's algorithm)
  - Did not prove the optimality
* Path metric $\rightarrow$ Hamming distance
  Expected output and Actual output

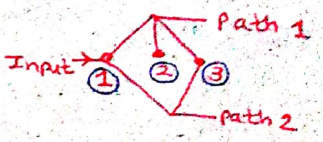
2^N M → states L+M → Levels

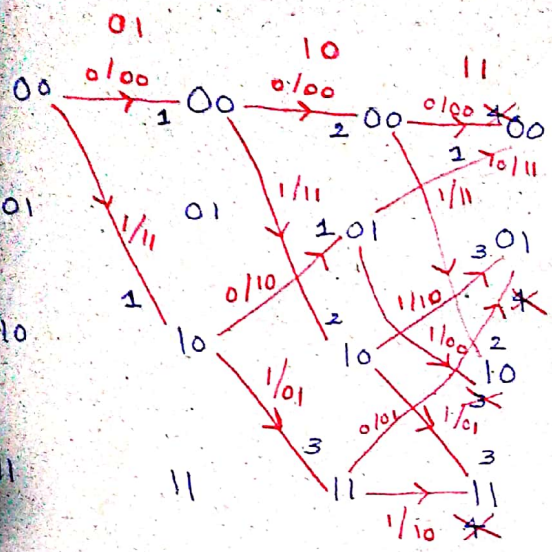
01 | 10 | 11 | 00 | 00 | 00 = r

c, m

Code rate = 1/2

No. of states





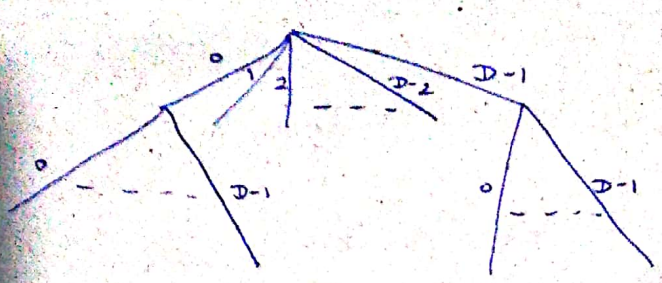
1 0 1 0 0 0

With the cost 1

Kraft's Inequality Proof

$\sum D^{-l_i} \leq 1 \Leftrightarrow$ prefix code

* Consider a D-ary tree ⇒



Possible code words by observing / traversing through branches of tree ⇒ 0, 1, 2, 3, 4, ---, D-1

length: 1, 1, 1, 1, 1, ---, 1

$\sum D^{-1} = D \cdot D^{-1} = 1 \leftarrow$ maximum

Binary :-

0
1 0
1 1 0
1 1 1 0
1 1 1 1 0
1 1 1 1 1 0

Taking sum of length :-

$= 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + \cdots$

$= \frac{1}{2^1} + \frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^4} + \cdots$

$\text{Sum} = \frac{1}{2} + \frac{1}{2}\left[\frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \cdots\right]$

$\text{Sum} = \frac{1}{2} + \frac{1}{2} \cdot \text{Sum}$

$\therefore$ Sum = 1 { Infinite length or Finite length with last two same length

$\therefore$ Sum < 1 { Finite length where last code length > previous last code