Tutorial I

09 October 2020 13:05

Basics of coding theory

1.) NOISELESS CODING THEORY
- → Shannon entropies
- → Huffman coding

2.) Noisy coding theory
- → Error correcting codes
- → Block codes

Noisless coding theory

Random variable X $\{x_0, \dots, x_{n-1}\}$

$p_0, \dots, p_{n-1}$ $\sum_{i=0}^{n-1} p_i = 1$

Σ → alphabet $\Sigma = \{0, 1\}$

$x_0 \rightarrow$	00
$x_1 \rightarrow$	01
$x_2 \rightarrow$	10
$x_3 \rightarrow$	11

↓ code C

→ is this the most efficient way?

$\log_2 n$ bits to find codewords for n different messages

$$AVG(C) = \sum_{i=0}^{n-1} p_i \cdot |c_i|$$

$|c_i|$ ... lenght of codeword for message x_i.

1.) given a probability distribution (random variable)
what is the best achievable $AVG(C)$?

2.) How to construct such a code?

1.) for a random variable X w.p. $(p_0, \dots, p_{n-1})$

$$S(X) = -\sum_{i=0}^{n-1} p_i \log_2 p_i$$ Shannon entropy

I. Average lenght of any code C for v.v. $X \geq S(X)$

II. Encoding multiple messages helps

III. As the number of messages encoded together approaches infinity $S(X)$ is achievable.

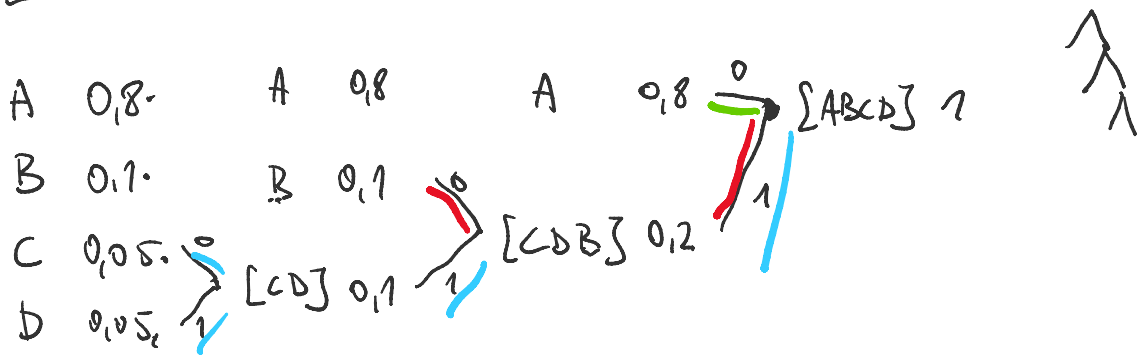
2.) Huffman coding

Alg.

INPUT: Probability distribution

OUTPUT: CODE

EX 1.2



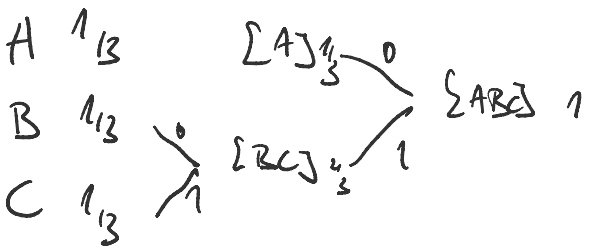
A → 0.
B → 10.
C → 110.
D → 111.

$$AVG(C) = 0{,}8 \cdot 1 + 0{,}1 \cdot 2 + 0{,}05 \cdot 3 + 0{,}05 \cdot 3$$

$$VL = 1.3$$

$$S(X) = -(0.8 \cdot \log_2 0.8 + 0{,}1 \cdot \log_2 0.1 + 0{,}05 \log_2 0.05 + 0.05 \log_2 0.05)$$

$$= 1.02 \ldots$$



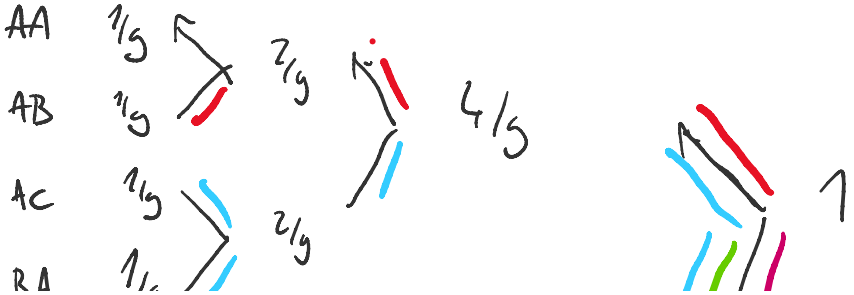
A → 0
B → 10
C → 11

$$AVG(C) = 1/3 \cdot 1 + 2 \cdot (1/3 \cdot 2)$$

$$VL = 5/3 = 1.666$$

$$S(X) = -3 \cdot 1/3 \cdot \log_2 1/3$$

$$= -\log_2 1/3 = 1.58$$



$$AVG(C) = 7 \cdot (1/9 \cdot 3) + 2 \cdot (1/9 \cdot 4)$$

$$= 3{,}2222 \ldots$$

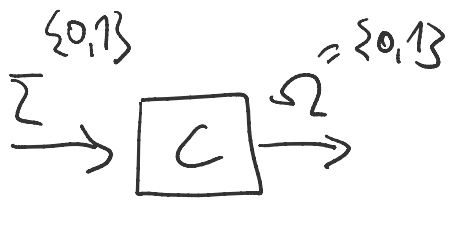
AVG (C) = 1.6111

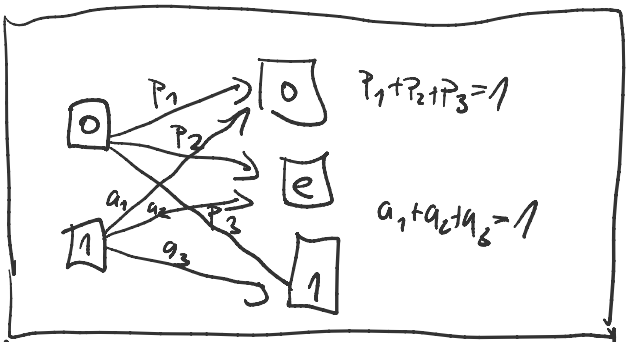
$$\frac{AVG(C)}{2} = 1.6111$$

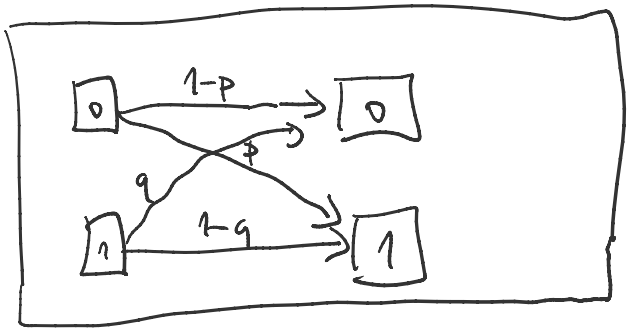
IF you encode k symbols together

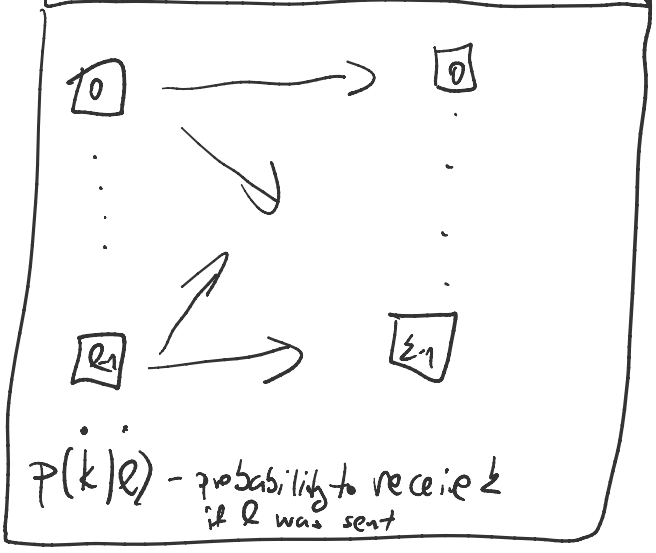
then $\lim_{k\to\infty} \frac{AVG(Huff(k))}{k} = S(X)$

Noisy coding theory (Error correcting codes)

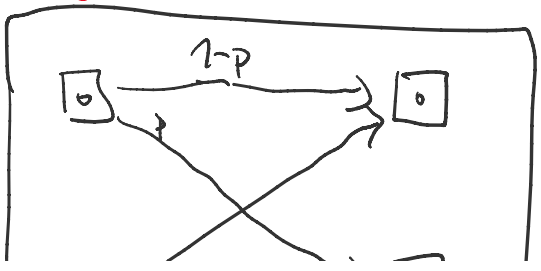


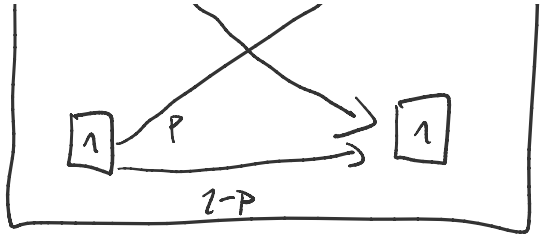






Binary symmetric channel





$p < 1/2$

Principle of maximum likelihood

You recieve 0 from a binary symmetric channel

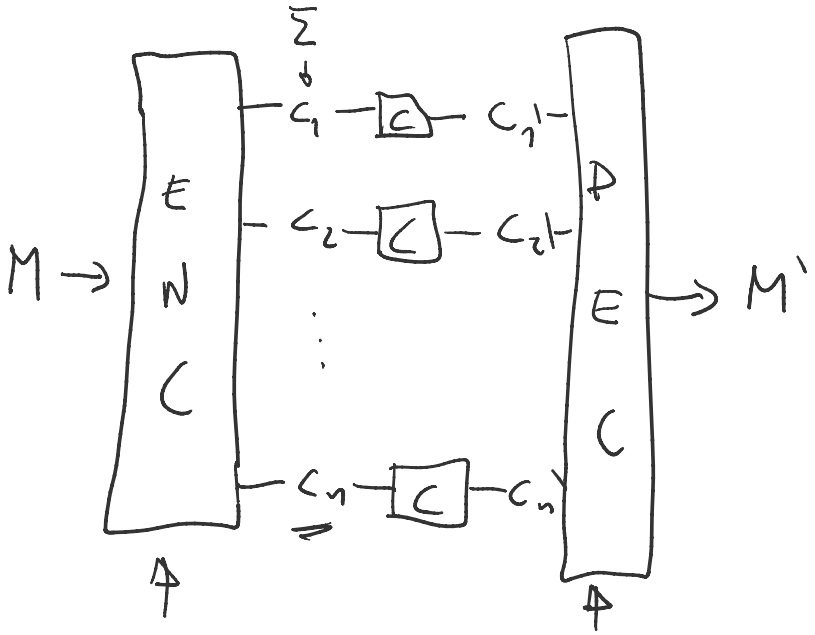
How do you decode it?

$0 \xrightarrow{1-p} 0$

$1 \xrightarrow{p} 0$

$p < 1/2 \Rightarrow p < 1-p \Rightarrow$ it is more likely 0 was sent

---



	ENCODER	DECODER	
$M \in \{0,1\}$	$0 \to 000$	$\#1 \geq 2$	decode as 1
$\Sigma \in \{0,1\}$	$1 \to 111$	$\#1 < 2$	decode as 0
	,in	,out	

$\Sigma \in \{0,1\}$ $\quad 1 \to 111$ $\quad \#1 < 2$ $\quad$ decode as 0

$\Sigma \in \{0,1\}$ $\quad \cdot P(000 \mid 001) = (1-p)(1-p) \cdot p \quad p < 1/2$

$P(111 \mid 001) = p \cdot p \cdot (1-p)$ (VI)

$\vdots$

Repetition code can achieve arbitrarily low probability of decoding

$n \in \{0,1\}$	ENC	DEC
$\Sigma \in \{0,1\}$	$0 \to \overbrace{0 \dots 0}^{2k+1}$	$\#1 \geq k+1 \Rightarrow 1$
$\Sigma \in \{0,1\}$	$1 \to 1 \dots 1$	$\#1 < k+1 \Rightarrow 0$

$000000\dots0 \to$ less than $k+1$ errors, the decoding is correct

$\to$ more than k errors, the decoding is incorrect

$$Pr(\text{correct decoding}) = \sum_{i=0}^{k} \binom{2k+1}{i} p^i (1-p)^{2k+1-i}$$

$$\lim_{k \to \infty} \quad Pr(\text{correct decoding}) = 1$$

$$\frac{\#\text{Messages}}{\text{length codewords}} = \frac{2}{2k+1} \underset{k \to \infty}{=} 0$$

Code rate

---

2 Homework exercise

#4

#5

#4

#2

Hamming distance

$c_i \in \{0,1\}^n \qquad c_i \in C \qquad C \subseteq \{0,1\}^n$

C - code

c_i - codeword

$Ham(c_i, c_j)$ the number of positions in which c_i and c_j differ

Ex 1.6 $\quad C = \{10001, 00110, 11010, 01101\}$

$Ham(10001, 00110) = 4 \qquad Ham(00110, 11010) = 4$

$Ham(10001, 11010) = 3 \qquad Ham(00110, 01101) = 3$

$Ham(10001, 01101) = 3 \qquad Ham(11010, 01101) = 4$

$n = 5$	length of the codewords
$M = 4$	number of codewords
$d = 3$	minimum distance

Error detection → if at most d-1 errors happen they detected (output is not a codeword)

Error correction ⇔ $d = 2t+1$ code can correct up to t errors

1 2 3 4

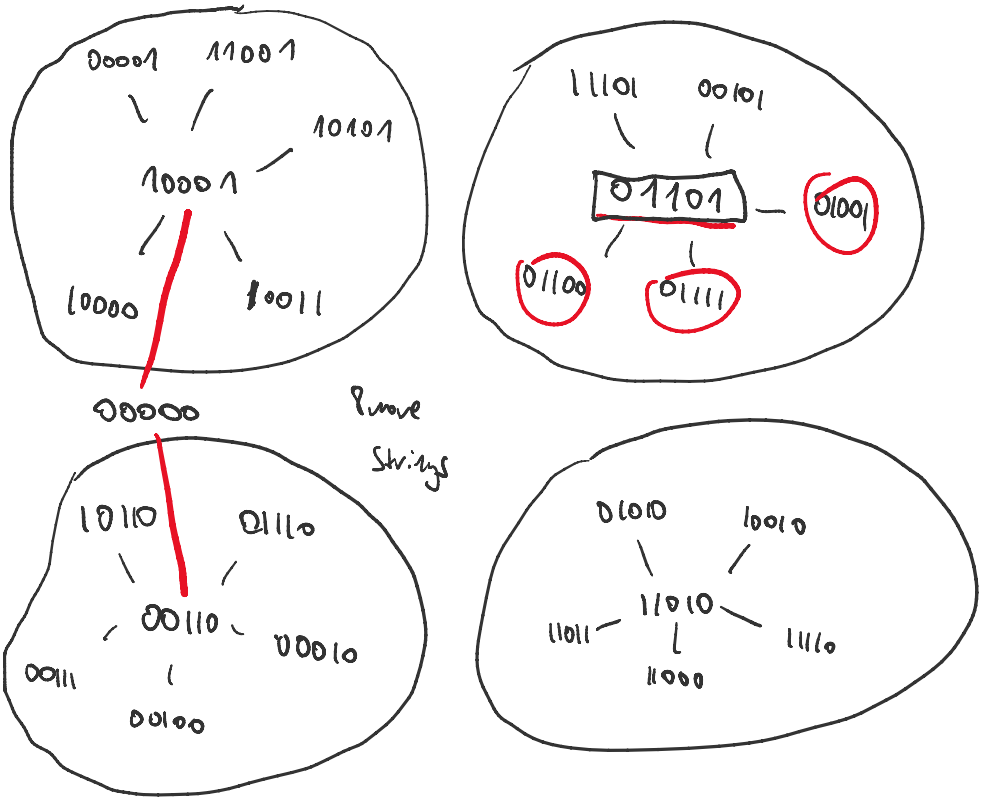
$C = \{10001, 01101, 00110, 11010\}$

$Ham(10101, 10001) = 1$

$Ham(01101, 10101) = 2$ ⇒ MLP ⇒ decode as 10001

$Ham(00110, 10101) = 3$

$Ham(11010, 10101) = 4$



$A_q(n, M)$ = the largest d of a code with codewords of length n (over q-ary alphabet)

alphabet size (usually $q=2$)

↓ n ↑ M ↑ d

Code equivalence (slide 53)

C_1 from C_0 by

1.) permutations of the position of the code

2.) permutation of symbols appearing in a fixed position

C_1 is equivalent to C_0

$C_0 = \{10001, 01101, 00110, 11010\}$

$C_1 \{01001, 10101, 00110, 11010\}$

$C_2 \{00001, 11101, 10110, 01010\}$

More efficient

Linear codes

Cyclic codes