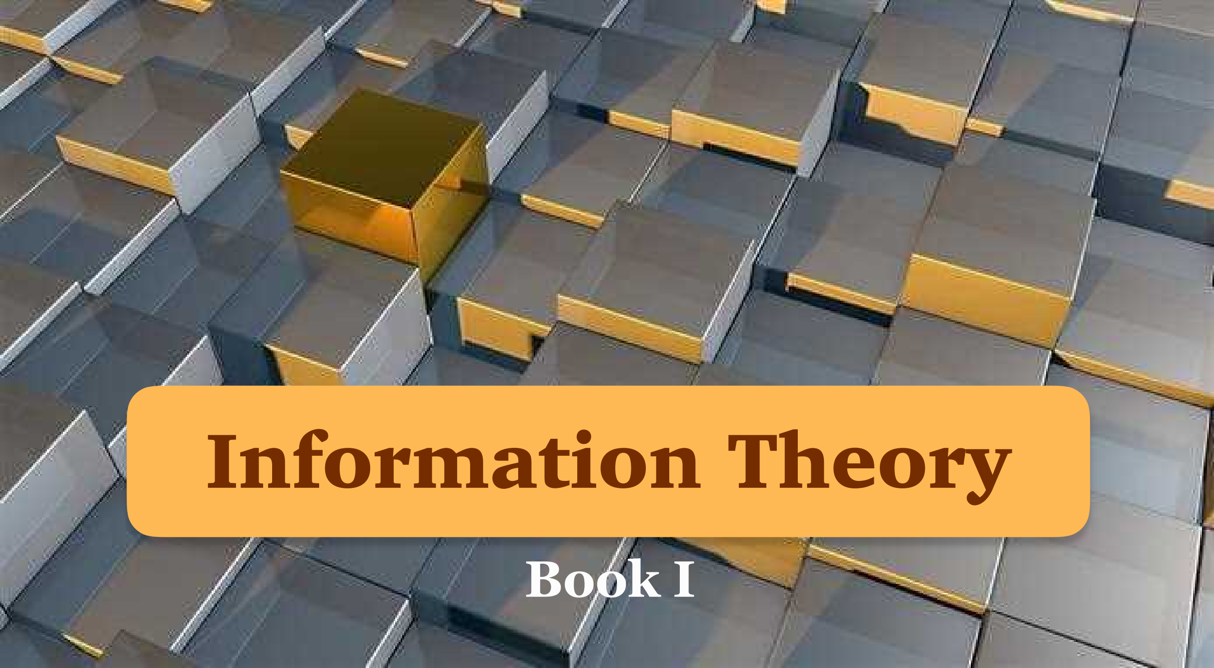


Information Theory

Book I

Information theory:

- Shannon information (statistical measurement of information content; classical information theory)
- Kolmogorov information (algorithmic information)
- Quantum information

Digital Photograph

10 MB bitmap $\longrightarrow$ 350 KB jpg $\xrightarrow{\text{lossy compression}}$ 80 KB

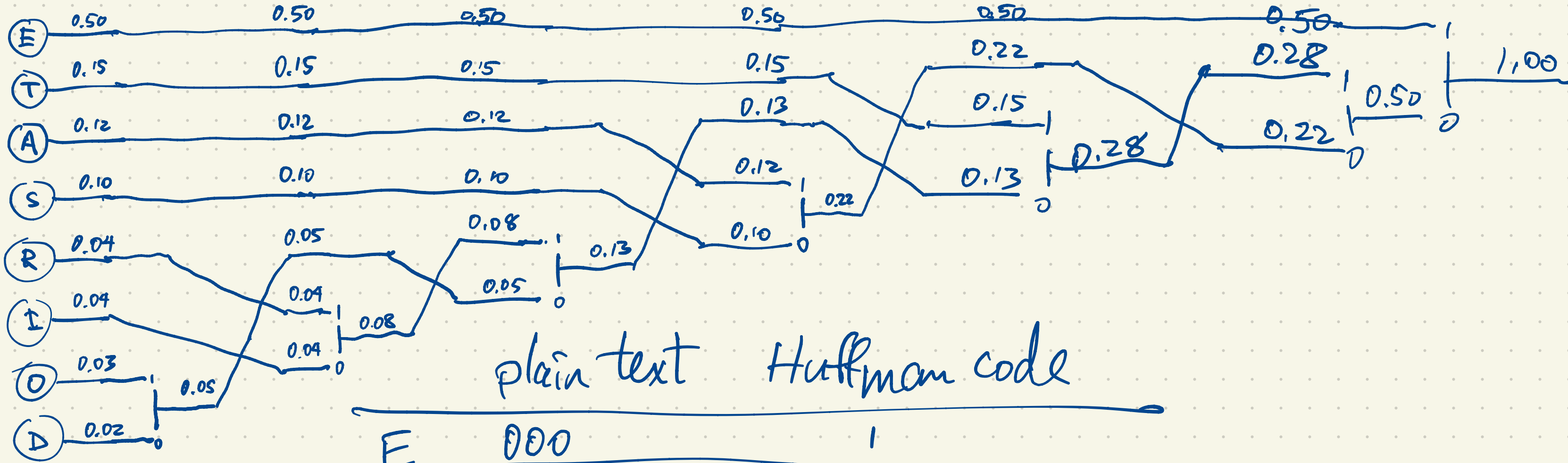
Text document

50KB $\xrightarrow[\text{loss-less compression}]{}$ 2.3 KB $\longrightarrow$ 2.4 KB

a perfect copy of the original can be extracted/recovered.

Consider an information stream composed of (stream of independent letters)

	E,	T,	A,	S,	R,	I,	O,	...,	D
freq.	0.50,	0.15,						...,	0.02



	plain text	Huffman code
E	000	1
T	001	011
A	010	001
S	011	000
R	100	01011
I	101	01010
O	110	01001
D	111	01000

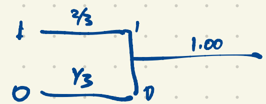
Huffman encoding:
Encode STEER as $\underbrace{000\,011\,1\,1\,01011}_{S\ T\ E\ E\ R}$
Decoding

A string of n characters is represented as 3n bits (plain text) which the Huffman code compresses to 2.26n bits. (75.3% of original)

Example 2 of Huffman coding:

Stream of $\underbrace{0\text{'s}}_{\frac{1}{3}}$ and $\underbrace{1\text{'s}}_{\frac{2}{3}}$

$X = \begin{cases} 1 & \text{with prob. } \frac{2}{3} \\ 0 & \cdots\cdots \frac{1}{3} \end{cases}$

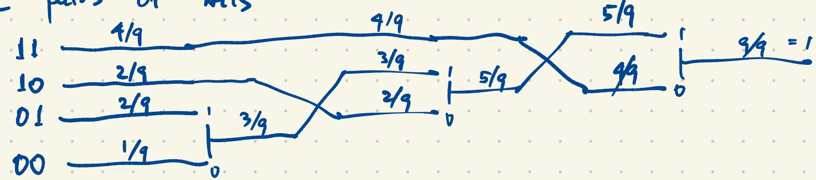


(binomial distribution)

Plain text	Huffman code
1	1
0	0

No compression

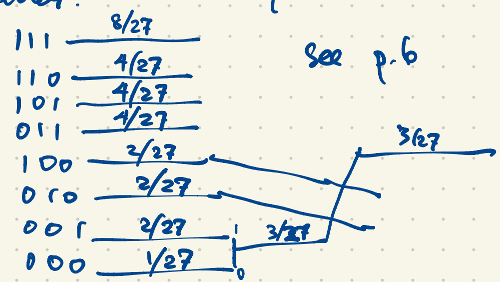
Take pairs of bits



Plain text	Huffman code
11	0
10	10
01	111
00	110

On average, a plain text file of n bits encodes as $\frac{17}{18}n \approx 0.9444\, n$ bits.

Better: encode triples of bits



see p.6

Plain text	Huffman code
111	11
110	00
101	101
011	0111
100	100
010	0110
001	0101
000	0100

On average, n bits is encoded as $\frac{76}{27}n$ bits $\approx 0.9383\, n$ bits.

What is the limit of the compression ratio (as the block size $\to \infty$)?

$0.9183\, n$ bits is the limit for compressing n bits from this stream

Shannon's first theorem showed that this stream has an entropy of

$$H(X) = \frac{1}{3}\log\frac{1}{1/3} + \frac{2}{3}\log\frac{1}{2/3} \approx 0.9183.$$

Example 1

Huffman code with blocksize 1 character gives n bits $\longrightarrow \frac{2.26}{3} n$ bits $\approx 0.753\, n$ bits

Entropy : $\sum_i p_i \log \frac{1}{p_i} \approx 1.55678$ bits per character

$p_i = 0.5, 0.15, \dots, 0.12 \quad (i = 1, 2, \dots, 8)$

Compare: plain text encoding of characters requires 3 bits.

Binary entropy function : A biased coin has heads with prob. p, tails ... $1-p$, $0 < p < 1$, with independent tosses

$$H(\text{coin}) = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}$$ = no. of bits (on average) to express the ~~outcome~~ of each coin flip.