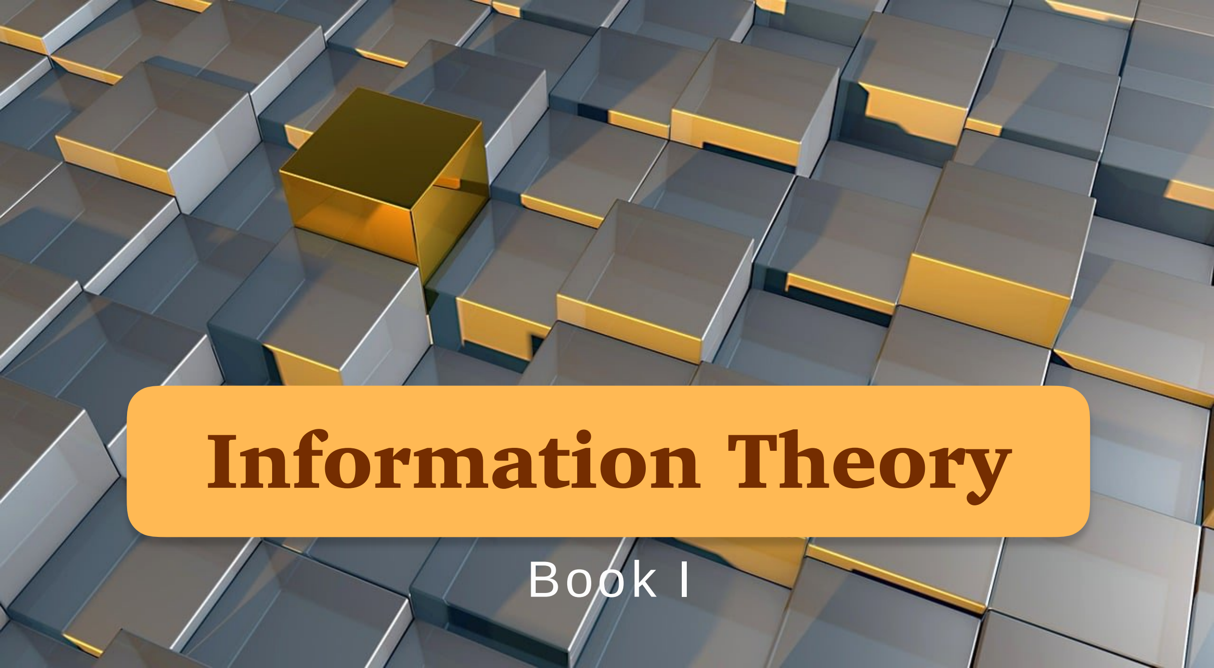


Information Theory

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

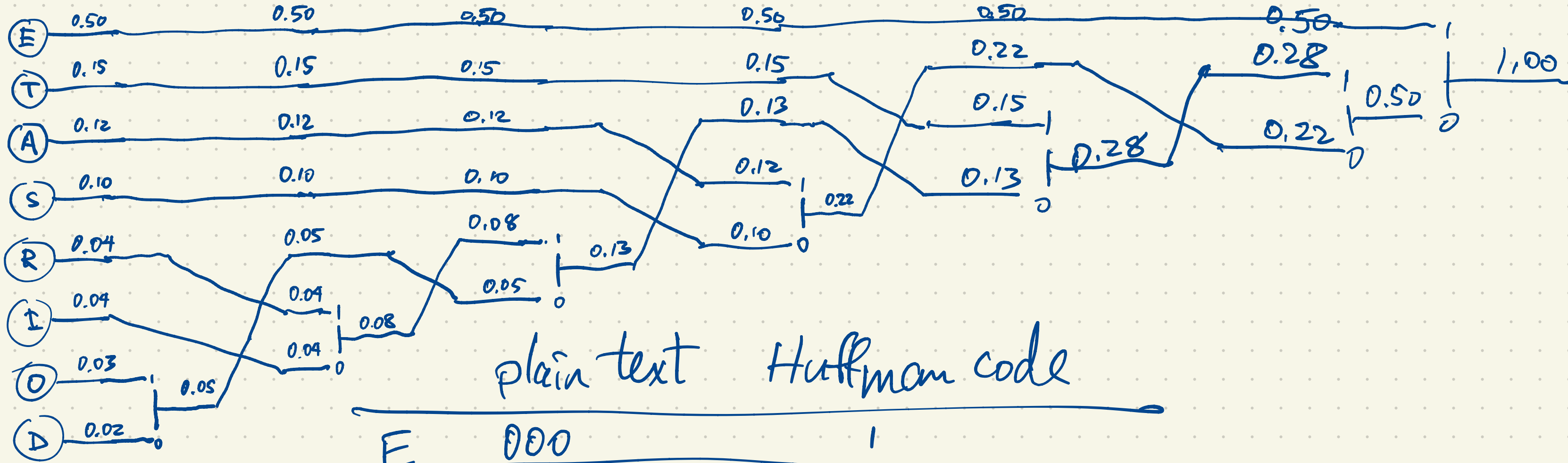
Text document

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 E, T, A, S, R, I, O, D

(stream of independent letters) 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 00001111101011
S T E E R
Decoding

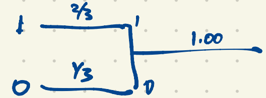
A string of n characters is represented as 3n bits (plain text) which the Huffman code compresses to 2.26n bits. (75.3% of original)

Example 2 of Huffman coding:

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

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

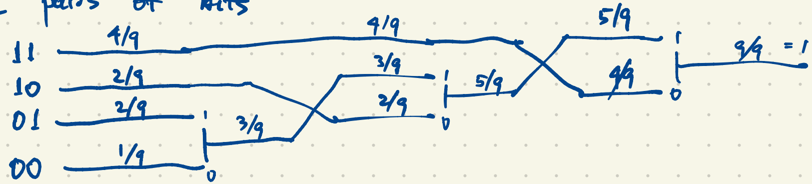
(binomial distribution)



Plain text	Huffman code
1	1
0	0

No compression

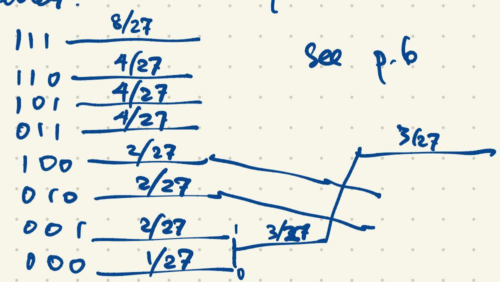
Take pairs of bits



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

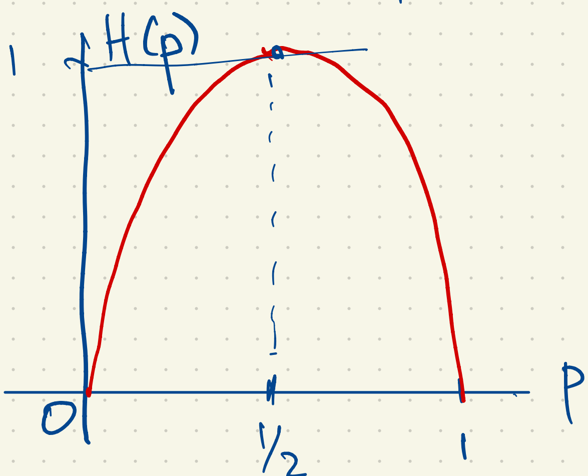
Entropy : $\sum_i p_i \log \frac{1}{p_i} \approx 1.55678$ bits per character

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

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.



Recall: If X is a random variable with outcomes $X = x_i$ $(1 \le i \le n)$ with prob. p_i $(\sum p_i = 1)$ then the binary entropy of X is $H_2(X) = \sum_{i=1}^{n} p_i \log_2 \frac{1}{p_i}$

= no. of bits on average required to express observed values of X.

When expressing information in base q, the q-ary entropy function $H_q(X) = \sum_{i=1}^{n} p_i \log_q \left(\frac{1}{p_i}\right) = \frac{1}{\log_2 q} H_2(X)$

Starting Friday, move to CR 144

Eg. A byte is 8 bits. $2^8 = 256$

If X can be encoded using N bits then it takes $\frac{N}{8}$ bytes.

If I buy a deck of cards its entropy is 0 in the sense that no information is required to express the order of the deck. After shuffling the deck, it takes 225.58 bits to express the order

↑ ignore jokers. $\log_2 52! \approx 225.58$ (about 68 decimals).

2nd Law of Thermodynamics

Watch the 7-8 minute video linked on course website

---

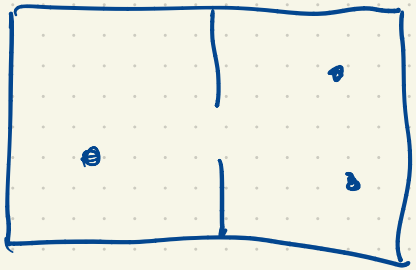
p. 49 Shannon's Source Coding Theorem (for channel without noise)

A channel is used to send a stream of symbols e.g. 0's and 1's reliably at a certain number C of bits per second. Information coming from a source X has finitely many outcomes with entropy $H(X) = H_2(X)$ bits per symbol eg. $x_1, \dots, x_n$ or $A, B, C, D, \dots$ This information can be reliably send and received at a maximum rate $\frac{C}{H} \; \frac{\text{bits/sec}}{\text{bits/symbol}} \Big\}$ symbols/sec.

Eg. X is a stream of characters $E, T, \dots, D$ (first example) with prob. $0.50, 0.15, \dots, 0.02$, $H(X) = 1.55$ bits/char.

If I transmit info. from this source using a noiseless channel with capacity 31 bits/sec. then I can safely transmit less that $\frac{C}{H} = \frac{31 \text{ bits/sec}}{1.55 \text{ bits/char}} = 20$ char./sec.

We can get within any pos. ε of this optimal rate i.e. $20 - \varepsilon$.



Suppose X, Y are independent random variables each with finitely many possible values

X has value x_i with prob. $p_i \in (0,1)$ $(1 \leq i \leq m)$ $\sum_i p_i = 1$

Y has value y_j with prob. $q_j \in (0,1)$, $\sum q_j = 1$

$$H(X) = \sum_{i=1}^{m} p_i \log \frac{1}{p_i} , \quad H(Y) = \sum_{j=1}^{n} q_j \log \frac{1}{q_j}$$

The pair (X, Y) has value (x_i, y_j) with prob. $p_i q_j$

$$\sum_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}} p_i q_j = \sum p_i \cdot \sum q_j = 1 \cdot 1 = 1$$

$H(X,Y)$ joint entropy

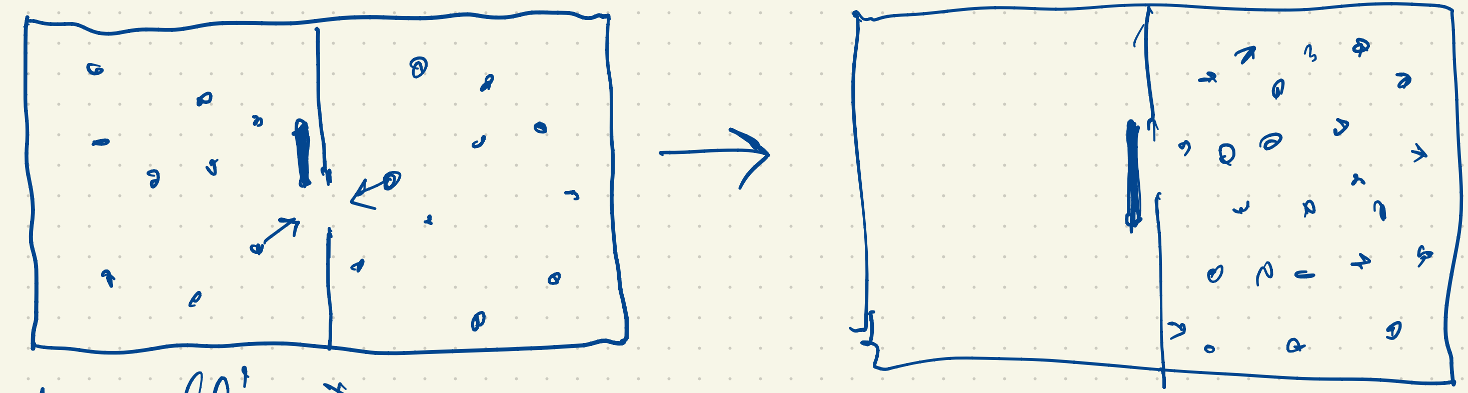
$$H(X,Y) = \sum_{i,j} p_i q_j \log\left(\frac{1}{p_i q_j}\right) = \sum_{i,j} p_i q_j \left(\log \frac{1}{p_i} + \log \frac{1}{q_j}\right)$$

$$= \sum_{i,j} p_i q_j \log \frac{1}{p_i} + \sum_{i,j} p_i q_j \log \frac{1}{q_j}$$

$$= \left(\sum_i p_i \log \frac{1}{p_i}\right) \underbrace{\sum_j q_j}_{1} + \underbrace{\left(\sum_i p_i\right)}_{1} \sum_j q_j \log \frac{1}{q_j} = H(X) + H(Y).$$

If X, Y are dependent

$$H(X,Y) \leq H(X) + H(Y)$$



Maxwell's Demon

Computation requires some minimal expenditure of energy when initializing memory registers, and when reading memory registers, resulting in the creation of entropy.

Sadi Carnot