DIC15

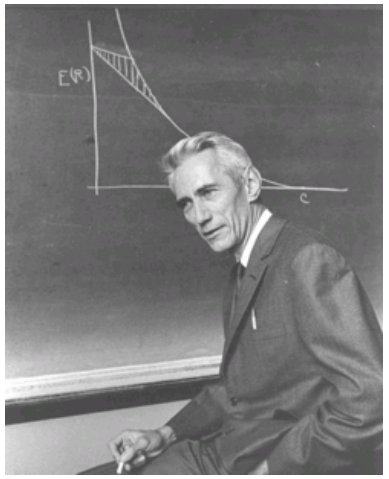
T1. Introduction

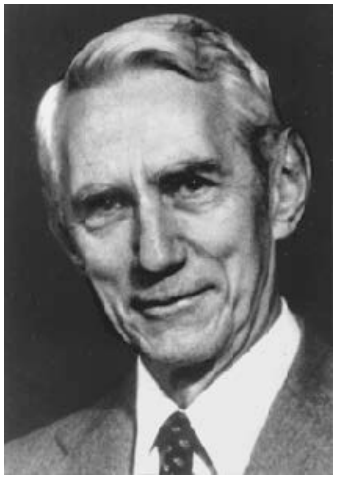
303 SXD

The digital era

A Mathematical Theory of Communication (Shannon, 1948)

- Mathematical foundations of *communication systems*
- Definition of *information*, of *channel capacity* and of *error-correcting codes*.
- Source and channel *coding theorems*

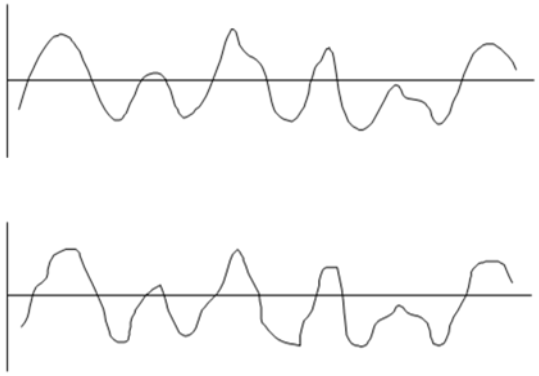




Claude Shannon (1916-2001)

- Master's thesis (1937): logic circuits
- II WWII: Cryptography

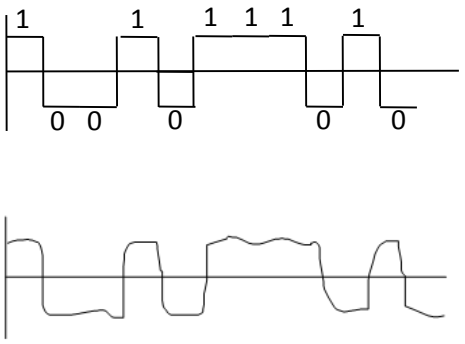
Analogical signal



Binary alphabet

$B = \{0,1\}$

Digital signal



Image

…and sound

Bach

[Dubliners](#)

J. Joyce

Rough orders of magnitude:

- Large textbook (e.g. Salomon et al., *Data compression*) $\simeq 40$ Mb.
- Large encyclopedia $\simeq 10$ Gb.

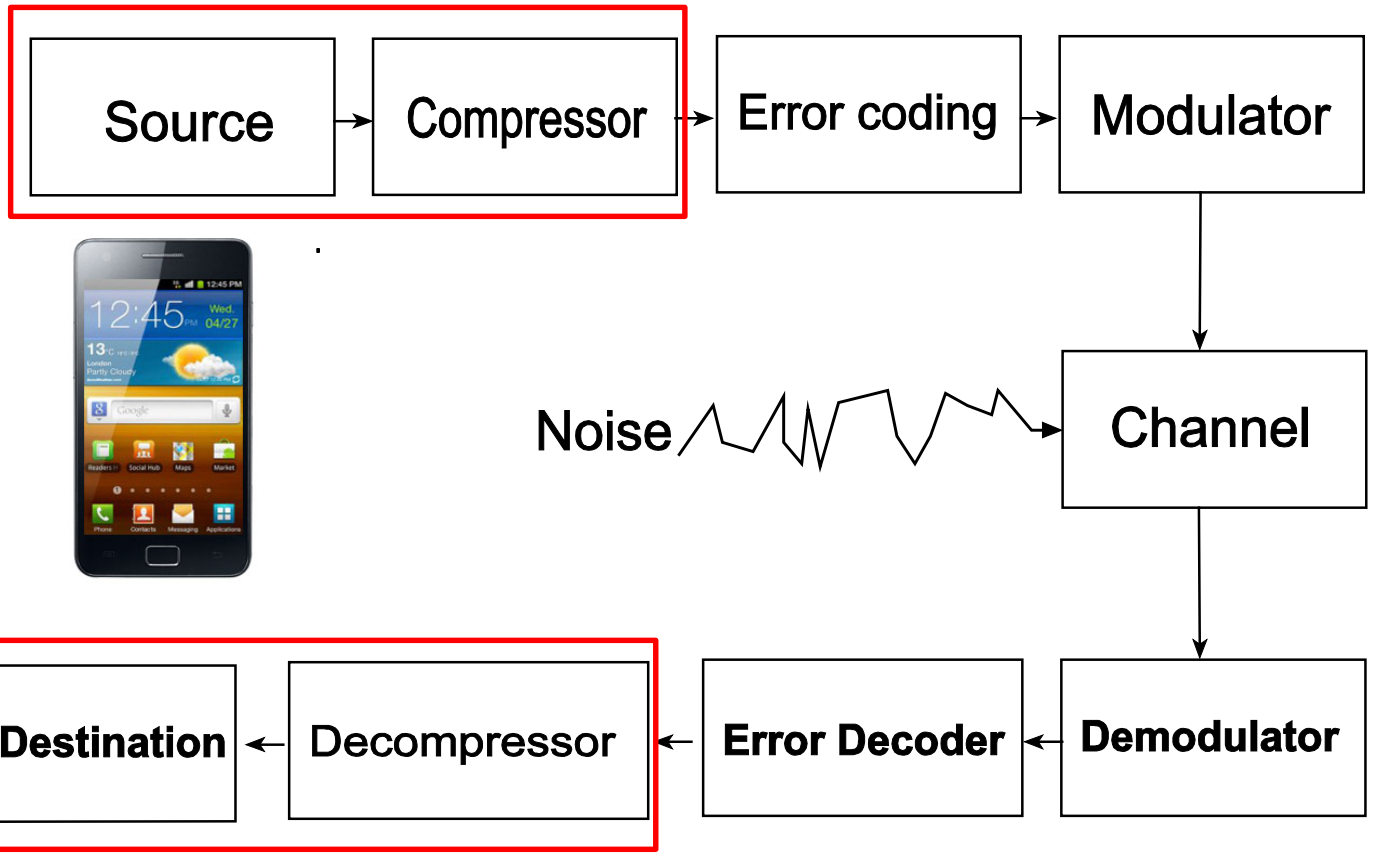
- 1 s of non-compressed digital audio $\simeq 1$ Mb.

- A non-compressed color picture $\simeq 25$ Mb
- 1 s of non-compressed digital video $\simeq 0.5$ Gb.

- Human genome $\simeq 6$ Gb.

1 Mb = 10^6 bit, 1 Gb = 10^9 bit

Model of a communication system



Binary symmetric channel (BSC):

$$\begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix} \end{array}$$

Compression is (ordinarily) possible

Example: *An instance of a Huffman code*

Alphabet: $A = \{a, b, c, d, e\}$
Message: $M = aaabccedabdb \in A^*$ (length 12).
Frequencies:

$$p(a) = 1/3\,,\ p(b) = 1/4\,,\ p(c) = p(d) = 1/6\,,\ p(e) = 1/12.$$

With a fixed-length binary encoding (e.g., ASCII-like):

$$a \to 000;\ b \to\ 001;\ c \to\ 010;\ d \to\ 011;\ e \to\ 100.$$

we need 36 bits to encode the message M. But we can do better:

$$a \to 00;\ b \to\ 01;\ c \to\ 10;\ d \to\ 110;\ e \to\ 111.$$

Note: Since no member of $\{00,01,10,110,111\}$ is a prefix of another member (for example, no one begins with 00 except 00), there is no ambiguity in 'decoding' the compressed message:

$$00\mathbf{00}00\mathbf{01}10\mathbf{10}111\mathbf{110}00\mathbf{01}110\mathbf{01}$$ (27 bits).

Ratio of compression: 27/36=3/4.

Example: 7-zip

One of my tex files, X.tex, has 76716 B. The file X.zip obtained with the compressor 7-zip has 25781 B. This is a compression factor 25781/76716=0.336.

Mathematical formulas and recipes may also provide a way to compress.

Street/mechanical organs

What is data compression?

$\mathcal{A}$ *source alphabet*

Examples

$\boldsymbol{\mathcal{A} = B = \{0,1\}}$ *binary alphabet.*

$\boldsymbol{\mathcal{A} = \{a, \dots, z, A, \dots, Z\}}$.

$\boldsymbol{\mathcal{A} = \{32, \dots, 128\}}$.

$\boldsymbol{\mathcal{A} = \{0\text{x}0000, \dots, 0\text{xFFFF}\}}$

M *message*:

a sequence/string/stream/word/vector of symbols from $\mathcal{A}$

Notation: $\mathcal{A}^*$ set of messages.

$\mathcal{C}$ *target alphabet*

$M \xrightarrow{\text{Compressor}} M'$, $M' \in \mathcal{C}^*$ *compressed message*.

$\rho = \text{bit\_length}(M')/\text{bit\_length}(M)$ *compression ratio*

(compression occurs for M when $\rho < 1$)

Lossless and lossy compression

If M can be recovered exactly from M', we say that the compressor is *lossless*. If M' only allows to obtain M approximately, we say that the compressor is *lossy*.

The text compressors we have seen (and also the mechanical organs) are examples of lossless compressors.

What data compression is not:

- ***Data encryption***
  In data compression we do not care who decompresses the data.
- ***Error-correcting/detecting-coding***
  One also assumes that the data will be transmitted/stored with no distortion.
- ***Watermarking/Stenography***
  We regard the source as it is, with no regard for its identity or hidden added content.

Remark. Most messages cannot be losslessly compressed. Assume, for simplicity, that messages are binary strings of length N. There are 2^N of them. Suppose we are aiming at a compression rate $0 < \rho < 1$ and let $M = \lceil \rho N \rceil$ (the smallest integer greater or equal to ρN).

Then the number of binary strings of length M or less is

$$2^0 + 2^1 + \cdots + 2^M = 2^{M+1} - 1 \simeq 2^{M+1}.$$

For large N, this is a rather small part of the number of messages, as

$$2^{M+1}/2^N = \frac{2}{2^{N-M}} \leq 4/2^{(1-\rho)N},$$

for $M \leq \rho N + 1$ and $2^{N-M} \geq 2^{N-\rho N-1} = 2^{(1-\rho)N}/2$.

Note that even if ρ is close to 1, the exponent $(1-\rho)N$ becomes very large when N is large, and hence the fraction of possible ρ-compressed files is negligible. For example, if $N = 1000$ (rather small in practical terms) and $\rho = 0.9$, then $(1-\rho)N = 100$ and $2^{(1-\rho)N} = 1.268 \times 2^{30}$.