Source Coding

C.Mailhes

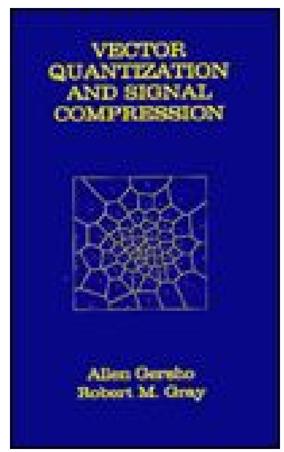
corinne.mailhes@n7.fr

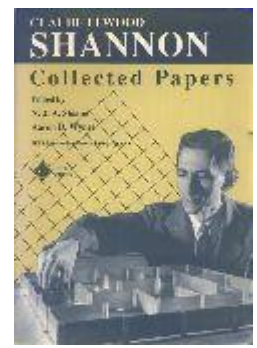
I - Source coding and communication system

II- Lossless Source Coding : Information Theory

III- Lossless Source Coding algorithms

- Huffman
- Lempel-Ziv (Welch)
- Arithmetic Coding

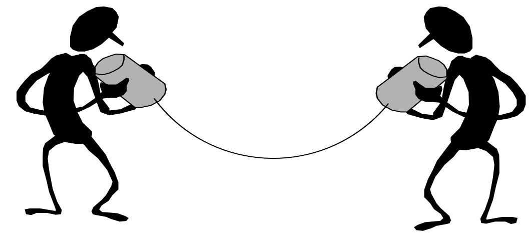




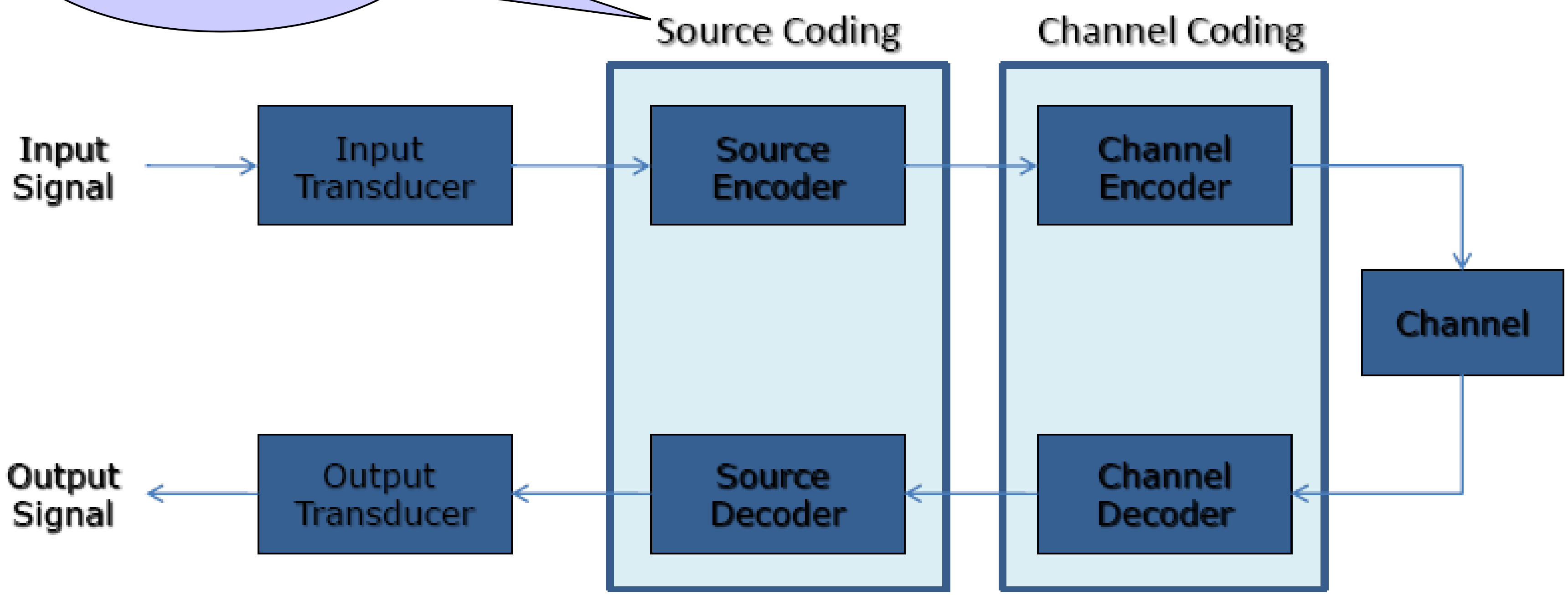
Version: October 2009

I –Source coding and communication system

Overview



What is it for ?



Basic Elements of a Digital Communication System

Economic Rôle ? → Source Coding

Fight against the noise (error control) ? → Channel Coding

Joint Source-Channel Coding ?

Source coding = data compression
To represent the source (data)
with the less code symbols as possible
and with the highest fidelity (lowest distorsion) as possible

Lossless compression:

Enables error free decoding

Unique decodability without ambiguity

Information theory

Lossy compression:

Distorsion and compression

Signal Processing

Claude Elwood Shannon (1916 – 2001), American electrical engineer and mathematician, has been called “the father of information theory”, and was the founder of practical digital circuit design theory.

II- Information Theory

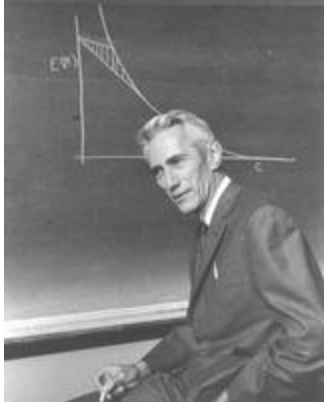
Définition of information quantity

➔ = doubt quantity, linked to the event probability:

$$i(x)=F(p(x)) \quad \text{with } F(1)=0$$

➔ additive quantity:

$$i(xy)=i(x)+i(y) \quad \text{if x and y independent}$$



Claude Elwood Shannon

(1916 -2001)

The « father » of Information Theory

Therefore

$$i(x)= - a.\log(p(x))$$ unity: ***Binary Unit***

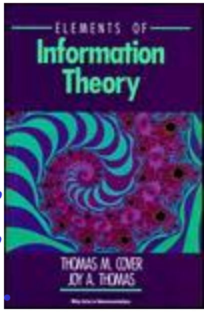
Related to the simpliest random experience: the equiprobable binary one

i(pile)=i(face)= - a.log(1/2) = 1 binary unit = 1 binit = 1 bit

By choosing to compute the function « log » in base 2, a=1 !

Thus

$$\boxed{i(x)= - \log_2 (p(x)) \text{ bits}}$$



Elements of Information Theory, Thomas M. Cover and Joy A.Thomas, John Wiley, 1991.

Other unities exist but « bits »: the most used

C. E. Shannon, « A mathematical theory of communication », *Bell System Technical Journal*, vol. 27, pp. 379-423 and 623-656, July and October, 1948.
See on web site http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf

Entropy

- It will be crucial to be able to quantify the amount of randomness of a probability distribution (a source !)

- Definition: The **entropy** H(X) of a discrete random variable X is defined by (also denoted H(p)):

$$H(X) = -\sum_{x} p(x) \log_2 p(x)$$

- The entropy of a distribution is expressed in *bits*.

You can view H as the expectation of -log (p(x)):

$H(X) = -\Sigma_x\, p(x) \log p(x) = E_p \{ -\log p(X) \}$.

Some Properties of H

- always H(X)≥0.
- H(X)=0 iif X is a ‘deterministic variable’ with p(x)=1 for one specific value $x \in \mathcal{X}$.
- If p(x) = 1/D for D different values $x \in \mathcal{X}$, then H(X) = log D.
- H(X) ≤ log(number of $x \in H$ with p(x)>0) ;

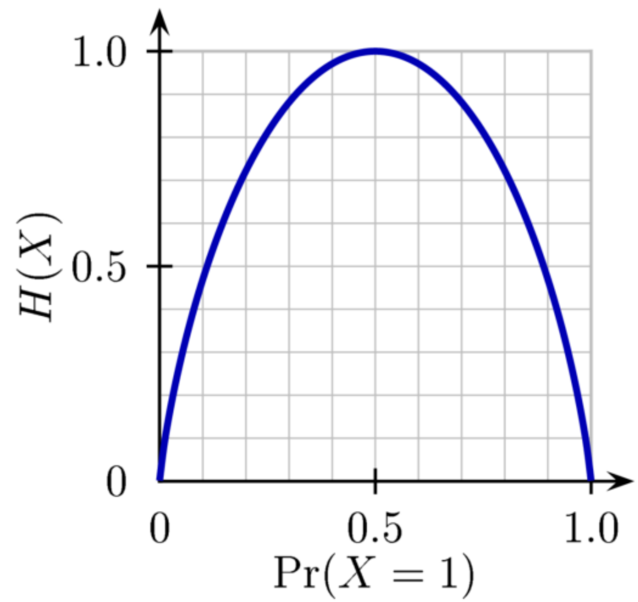
H(X) ***maximum for equiprobability statistics*****.**

- Decomposition increases entropy

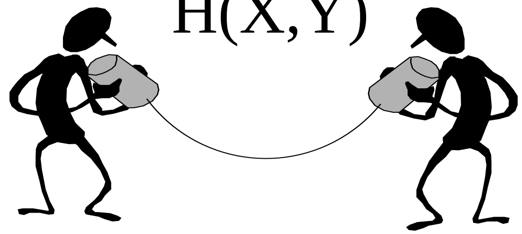
$$H(p1, p2, \ldots, pN) > H(P,Q)$$

Entropy of a Bit

- A completely random bit with p=(½,½) has
  H(p) = –(½ log ½ + ½ log ½) = –(–½ + –½) = 1.
- A deterministic bit with p=(1,0) has
  H(p) = –(1 log 1 + 0 log 0) = –(0+0) = 0.
- A biased bit with p=(0.1,0.9) has H(p) = 0.468996…
- In general, the entropy looks as follows as a function of 0≤Pr{X=1}≤1:



Entropies



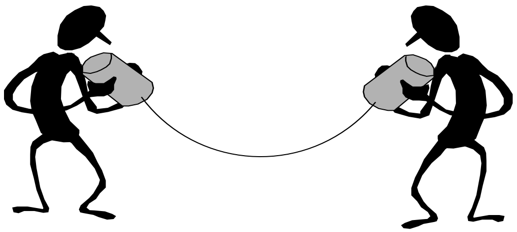
- The expected entropy of Y after we have observed a value $x \in X$, is called the *conditional entropy* H(Y|X)

$$
\begin{aligned}
H(Y|X) &= \sum_x p(x) \cdot H(Y|X = x) \\
&= -\sum_x p(x) \cdot \sum_y p(y|x) \log p(y|x) \\
&= -\sum_{x,y} p(x,y) \log p(y|x) \\
&= -E_{p(x,y)} \log p(Y|X)
\end{aligned}
$$

Also H(X/Y)

Chain rule: H(X,Y) = H(X)+H(Y|X) = H(Y)+H(X|Y).

Mutual Information

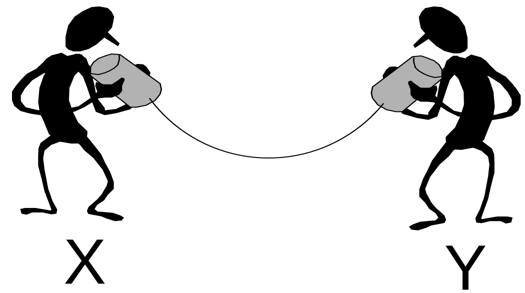


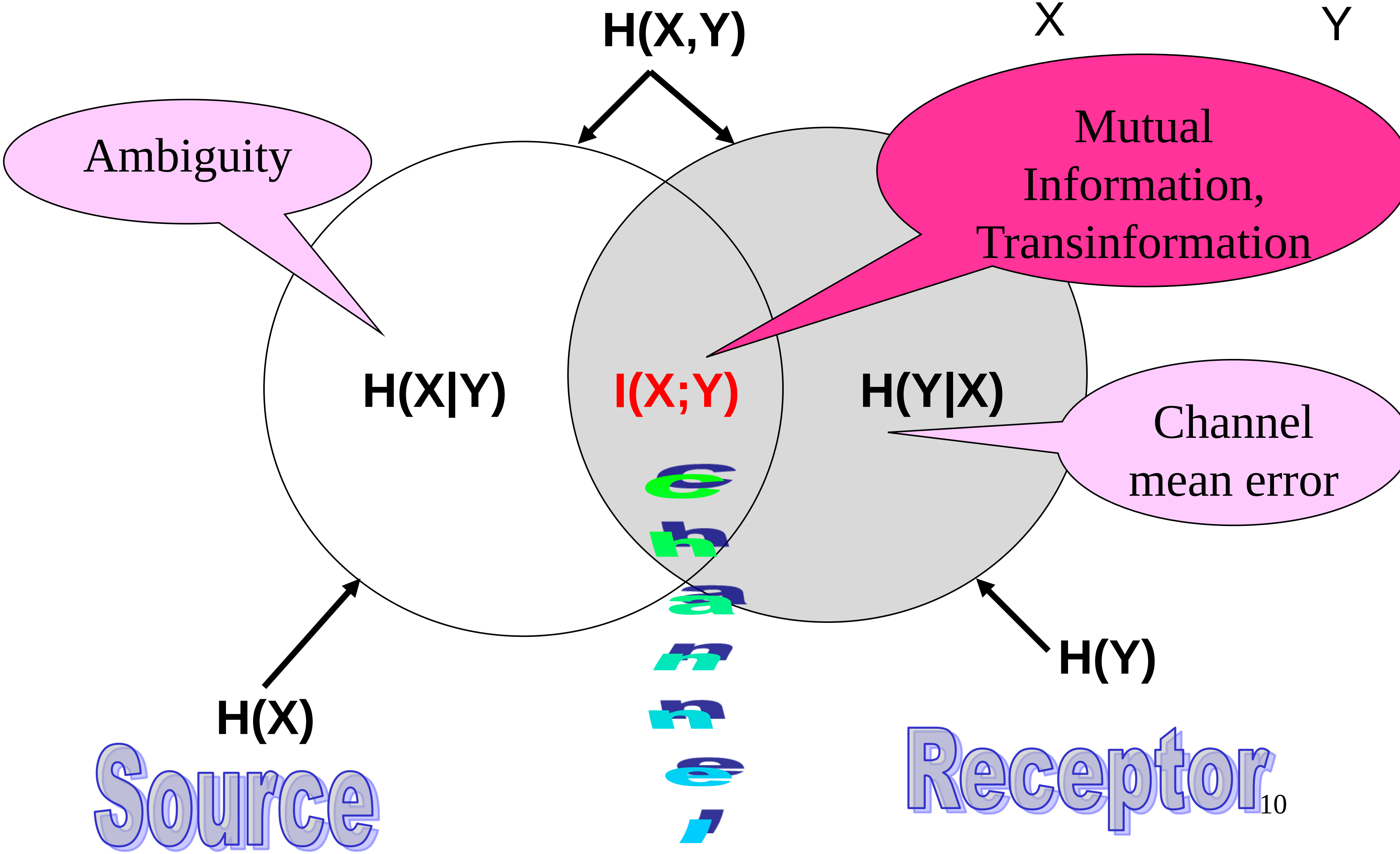
- For two variables X,Y the *mutual information* I(X;Y) is the amount of certainty regarding X that we learned after observing Y. Hence I(X;Y) = H(X)–H(X|Y).

- Note that now X and Y can be interchanged using the chain rule: $$\begin{aligned} I(X;Y) &= H(X) - H(X \mid Y) \\ &= H(X,Y) - H(Y \mid X) - H(X \mid Y) \\ &= H(Y) - H(Y \mid X) \\ &= I(Y;X) \end{aligned}$$

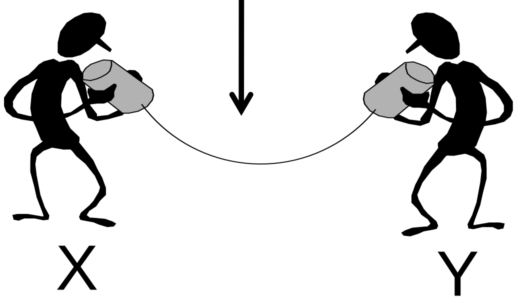
- Think of I(X;Y) as the 'overlap' between X and Y.

All Together Now





Channel Capacity



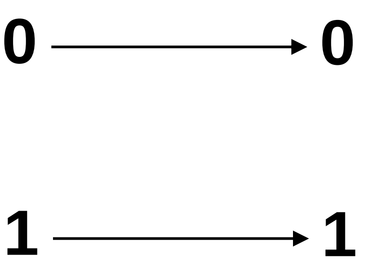
The channel capacity C is the maximum over all possible p(x):
$C = \max_{p(x)} I(X;Y)$.

[Cover & Thomas, Section 8.3]:
$C \geq 0$ and
$C \leq \log|X|$ and $C \leq \log|Y|$ as $I(X,Y) \leq \log|X|, \log|Y|$.

Some Example Capacities

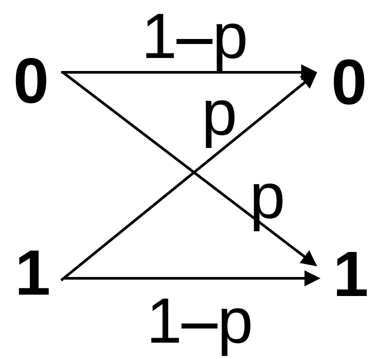
- A **noiseless binary channel** has $H(X|Y)=0$, hence for the mutual information $I(X;Y)=H(X)$, which is maximized by $p(0)=p(1)=½$.

Hence $C = \max_p I(X;Y) = 1$ bit.



- For a **noisy, symmetric binary channel** we have $H(Y|X=x)=H(p)$, hence $I(X;Y) = H(Y)–H(p)$.

Hence $C = \max_p I(X;Y)$ is obtained for $H(Y)=1$ (again $p(0)=p(1)=½$), such that $C = 1–H(p)$.



III- Coding algorithms

Discrete source alphabet: X={$x_1, x_2, \ldots, x_N$} N messages

Entropy H(X) (bits)

Discrete channel with alphabet: U={$u_1, u_2, \ldots, u_D$}

Capacity C (bits) D symbols

Often N > D

Coding : x_k => codeword : $m_k = u_{n_1} u_{n_2} \ldots u_{n_k}$ n_k : length of the codeword

Code mean length $\bar{n} = \sum_k p_k n_k$

As small as desirable ?...

Entropy of codewords

Source with H(X) delivers messages with $\bar{n}$ symbols of code:

$$H(X) / \bar{n} \leq \log_2(D)$$

Entropy = minimum mean length of binary code

Efficacity $E = H(X) / \bar{n} \log_2(D)$ Redundancy $\rho = 1 - E$

CODE PROPERTIES

- A code is **non-singular** if every element of S_X maps into a different string in D^*, i.e., $x_i \neq x_j \Rightarrow C(x_i) \neq C(x_j)$.

- A code is a **uniquely decodable code** if its extension is non-singular.

- A code is a **prefix code** or an **instantaneous code** if no codeword is a prefix of any other codeword.

(*no codeword is the beginning of an other*)

There exists almost one instantaneous code such that

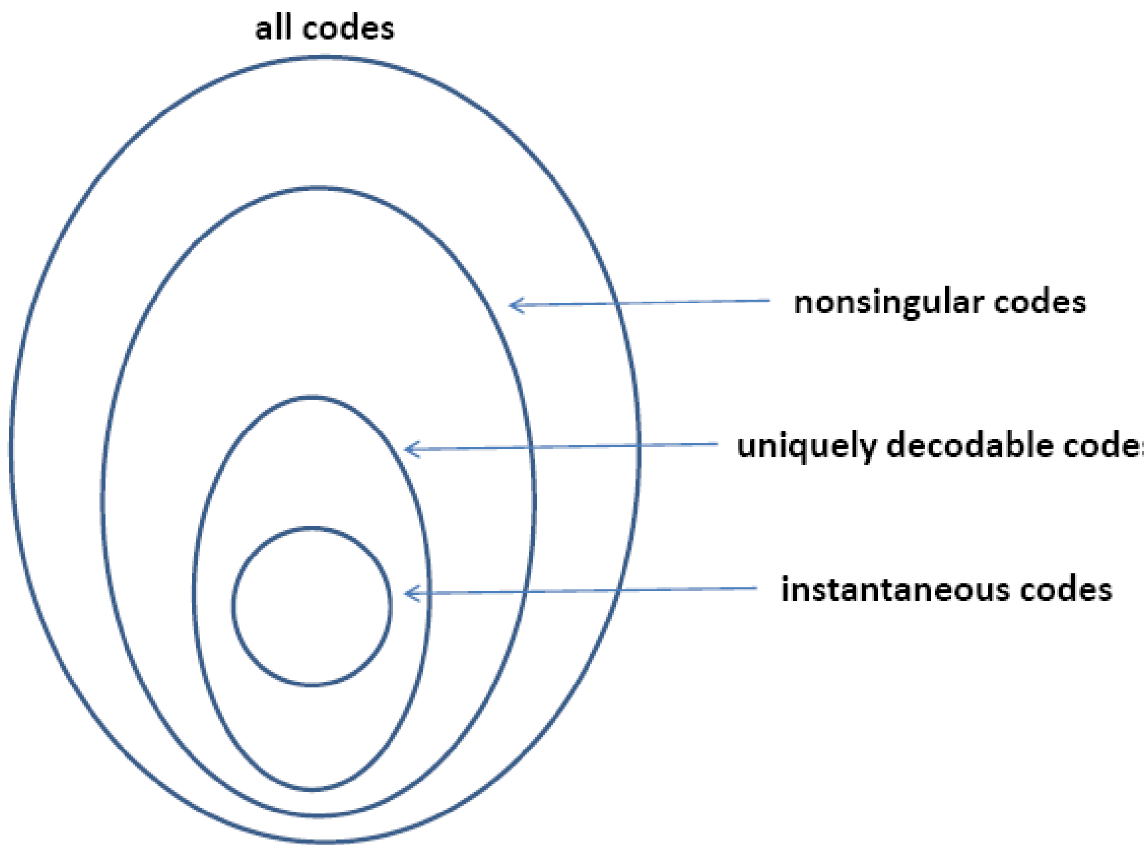
$$H(X) / \log_2(D) \leq \bar{n} \leq H(X) / \log_2(D) + 1$$

INTERNATIONAL MORSE CODE

1. A dash is equal to three dots.
2. The space between parts of the same letter is equal to one dot.
3. The space between two letters is equal to three dots.
4. The space between two words is equal to five dots.

A • —
B — • • •
C — • — •
D — • •
E •
F • • — •
G — — •
H • • • •
I • •
J • — — —
K — • —
L • — • •
M — —
N — •
O — — —
P • — — •
Q — — • —
R • — •
S • • •
T —
U • • —
V • • • —
W • — —
X — • • —
Y — • — —
Z — — • •

1 • — — — —
2 • • — — —
3 • • • — —
4 • • • • —
5 • • • • •
6 — • • • •
7 — — • • •
8 — — — • •
9 — — — — •
0 — — — — —



Huffman codes

- Huffman codes are special prefix codes that can be shown to be optimal (minimize average codeword length)

Huffman Algorithm:

1) Arrange source letters in decreasing order of probability ($p_1 \ge p_2 .. \ge p_k$)

2) Assign '0' to the last digit of X_k and '1' to the last digit of X_{k-1}

3) Combine pk and pk-1 to form a new set of probabilities

$$\{p_1, p_2, ..., p_{k-2}, (p_{k-1} + p_k)\}$$

4) If left with just one letter then done, otherwise go to step 1 and repeat

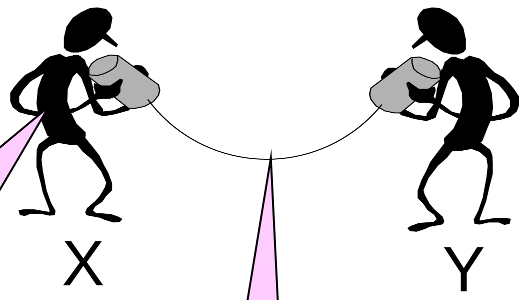
Huffman code: an example

Codewords	Source messages	Probas						
1	A	0.4	0.4	0.4	0.4	0.4	0.4	0.6 0
001	B	0.18	0.18	0.18	0.19	0.23	0.37 0	0.4 1
011	C	0.1	0.1	0.13	0.18	0.19 0	0.23 1	
0000	D	0.1	0.1	0.1	0.13 0	0.18 1		
0100	E	0.07	0.09	0.1 0	0.1 1			
0101	F	0.06	0.07 0	0.09 1				
00010	G	0.05 0	0.06 1					
00011	H	0.04 1						

Huffman mean length: 2.61

H(X)=2.55 bits thus an efficiency of E=97.8%

What else ?



Source:
Entropy **H(X)**
(bits)

$d_S = 1/T_s$ messages/s

Channel:
Capacity **C**
(bits)

$d_C = 1/T_C$ symbols/s

Can we use
any channel
for
any source ?

NCC theorem

Shannon's noisy channel-coding theorem shows that unreliable channels can be used for reliable communication if we code our messages cleverly.

More specifically, the theorem states that each (discrete, memoryless) channel has a *capacity* C'≥0, such that each "bits per transmission" *rate*

R < C' (bits / s)

is achievable if we use long enough codes.

(Achievable here means that the error probability $\lambda^{(n)}$ tends to zero as the length n of the codes grows.)

Once we have a « good » channel, can we use any code ?

source → coder → $\bar{n} d_S = \bar{n}/T_s$ symbols/s → $d_C = 1/T_C$ symbols/s → channel

$\bar{n} < d_C / d_S = T_S / T_C$

Code has to be efficient enough !... And if not ?

Noiseless coding theorem

Source $X=\{x_1,x_2,\ldots,x_N\}$. It can be shown:

There exists a « good » code with mean length $\bar{n}$ such that

$$H(X)/\log_2(D) \leq \bar{n} \leq H(X)/\log_2(D) + \mathbf{1}$$

Now code the « source extension » $X_k = \{ \underbrace{x_1x_1\ldots x_1}_{\text{Bloc of k messages}}, \ldots, x_Nx_N\ldots x_N\}$

$$H(X)/\log_2(D) \leq \bar{n} \leq H(X)/\log_2(D) + \mathbf{1/k}$$

Huffman Coding: The Retired Champion

- Replacing an input symbol with a codeword
- Need a probability distribution
- Hard to adapt to changing statistics
- Need to store the codeword table
- Minimum codeword length is 1 bit

1952

JPEG
MP3
…

Huffman Coding (1952) : optimal code if source statistics known

If unknown, no more optimal ...

Arithmetic Coding: The Rising Star

- Replace the entire input with a single floating-point number
- Adaptive coding is very easy
- Fractional codeword length

1981

Dictionnary-based coding (Ziv-Lempel and &): another alternative

- No statistics estimation !

1977