Lecture Notes in « INFORMATION THEORY»

October 2010

Bibliography

    http://www.cs.toronto.edu/~mackay/itprnn/ps/
    pédagogique de Télécommunication, MASSON, 1997
    Sciences et Technologies, éditions Technip, 1999
    http://www.montefiore.ulg.ac.be/~lwh/Info/
    MASSON et Cie, 1970
[Spa87] A. Spataru, « Fondements de la théorie de la transmission de l’information », presses polytechniques
    romandes, complément au traité d’électricité, 1987

Context

Information Theory for the (source and Noisy-channel) coding

I Information theory main tools: measure of information
   I.1 History and introduction
   I.2 Information content of an outcome (one emitted symbol)
   I.3 Entropy or average information content of an ensemble (discrete source)
   I.4 Various entropies between two random variables and Mutual Information
   I.5 Entropy of Markov sources : an introduction, Data processing theorem.

II Source Coding (or Data Compression):
   II.1 Introduction
   II.2 Characterization of an encoding
   II.3 The source coding theorem (first Shannon theorem)
   II.4 Source coding techniques: Shannon-Fano coding, Huffman coding.

III Channel Capacity and Noisy-channel coding theorem (discrete case):
   III.1 Channels » of information theory
   III.2 Capacity of a discrete memoryless channel
   III.3 The Noisy-Channel coding (or fundamental) theorem
       (second Shannon theorem)

IV Introduction to information theory in the case of a continuous channel
Case of a continuous channel with Additive White Gaussian Noise, entropy and capacity formula,
Appendix : geometric representation of signals and fundamental theorem in the continuous case,
performance of various modulations compared to the limits of the information theory.
Foreword

These lectures notes (and Problem Sessions) presents an introduction to Information Theory. The illustrated context especially turned to communication theory (initial context for which information theory was developed by Claude Shannon), but the scope of information theory is of course much wider.

The Figure 1 gives a block diagram of a (point-to-point) communication system:

Figure 1: Typical block diagram of a digital transmission via carrier modulation

In the emission stage, the aim is to transform a digital source (discrete symbols stream) into an analog (digitally-modulated) signal \( x(t) \):
- with a first CODING part, which can be separated in "source coding" and "channel coding" that transforms the pattern of discrete symbols \( S \) in a different pattern of discrete symbols \( B \)
- then a DIGITAL MODULATION part that transforms the pattern of symbols \( B \) in the analog signal \( x(t) \) matched to the support of transmission.

Information theory provides an insight on the coding/decoding (source and channel) parts, which will be especially developed in the course (with discrete random variables). Note that it makes also elegant lighting on the performance of digital modulations, as it will be briefly mentioned with the last part devoted to the continuous random variables and AWGN (Additive White Gaussian Noise) channel.
Notations:

**Symbol rate of a discrete source X:** \(D(X)\), in symb/sec or bauds

For a digital source X, the Symbol Rate or Baud-rate \(D(X)\) is the number of items (symbols) that is produced on average per unit of time (second). Hence, if X emits symbols once every T seconds, the Symbol Rate or Baud-rate is \((1/T)\) symbols per second. (Be careful, the symbol rate is a data-rate generally different to the "information rate" that will be defined in Information Theory).

Note that \(D(X)\) do not give all the knowledge about the Data flow, and we have also to specify the radix (size) \(Q\) of the alphabet \(X = \{x_1, x_2, \ldots, x_Q\}\) from which the symbols are taken. It can be sometimes useful to consider an *equivalent* bit rate (or binary rate) denoted \(D_b(X)\) and defined by :

\[
D_b(X) = D(X) \cdot \text{lb}(Q) \quad \text{in bit/sec}
\]

Remarks : \(\text{lb}(Q) = \ln(x) / \ln(2)\) is the base 2 logarithm, in bit/symb.

The definition for \(D_b(X)\) is always valid, that is to say even if \(\text{lb}(Q)\) is not an entire number.

And for binary symbols \((Q = 2)\), \(D_b(X) = D(X)\).

Appendix : Point to point communication or communication network?

Many digital transmission systems operate today in a network where cohabit several transmitters and receivers. In addition to the physical link between two elements, new issues appears as: how much information can be carried over a wireless network with a multiplicity of nodes, and how should the nodes cooperate to transfer information?

This introductory course does not directly adress these issues in dealing mainly with the case of the point-to-point link, but we give below some references in "Network Information Theory" :

I. Information Theory main tools :

Measure of information

I.1 History and introduction

Information Theory (I.T.) was born in the context of the statistical theory of communications. Its methods, mainly mathematical, are useful to evaluate the performance of a digital communications system. It deals with the more fundamental aspects of the communications systems, from only probabilistic models of the physical sources or channels.

- 1928, Hartley: first attempt to scientific definition of a quantitative measure of information.
- 1948 C. Shannon: introduces the new concept of "quantitative measure of information " from a mathematical way, and deduces the main consequences about fundamental limits on compressing and reliably communicating data: real beginning of the "information theory".


- A scientific theory of information needs first starting with a scientific definition of the word "information" with so a precise meaning which may differ from common language. It seeks to assign a numeric quantity to measure information content of messages by using the emission probabilities of different messages. The idea of "information" is closely related to that of "uncertainty" or "surprise". The meaning of "information" is very restrictive, since it does not concern "message importance or meaning", nor personality of the recipient ("subjective" aspect).

- The scope of the theory exceeds that of the engineering problem of the transmission of information, it is also a considerable summary power to explain sometimes surprising experiences in areas as diverse as Physics (thermodynamic, optical, radiation,...), statistical inference, natural language processing, biology, or even as sociology, economy...

- Applied to communications, the (initial) aims of the I.T. was to characterize source, channel and recipient to evaluate the theoretical limits of transmission according to various settings and adequate coding / decoding processes. It provides measurable quantities on:
  1. the amount of information issued by a discrete symbols source
  2. maximum amount of information (or information rate) at which reliable communication can take place over a noisy channel

It should be noted that the knowledge of symbol rates (of the source, of the channel) is not enough to assess 1. or 2., since for example:
- a source which emits the same symbol 1000 times / second does not carry information,
- a channel that transmits 1000 symbols per second do not carry the same amount of information if the error probability \( Pe = 10^{-1} \) or \( Pe =10^{-9} \).

Today, a research field of I.T. concerns not only the point to point (Shannon) link but more the capacity and optimization of complex networks communication systems (Cf chapter "foreword").

I.2 Information content of an outcome (one emitted symbol)

The quantity of information associated with the realization of an event among \( N \) possible objectively reflects the “uncertainty” or the degree of “surprise” of the event. It is therefore especially large as the probability of the event was small (Note: large uncertainty before the event occurs ⇔ large information content (or surprise) after).
Context:
Suppose a probabilistic experiment that involves an outcome \( s \) chosen from a set of \( N \) finished elementary events (results) or possible alternatives: \( As = \{ s_1, s_2, ..., s_N \} \).

The sample space \( As \) is such that \( s_i \cap s_j = \emptyset, \forall i, j, i \neq j \) and \( E = (s_1 U s_2 U ... U s_N) \) is the certain event.

To each possible outcome is assigned a probability \( p(s_i) \in [0, 1] \) still noted \( p_i \), with of course \( \Pr(E) = \sum_{i=1}^{N} p_i = 1 \).

Most of the time in the course, the \( N \) elementary elements will represent the possible outputs of a discrete source \( S \) at a given time. \( As \) is then the source alphabet, and \( p_i \) is the probability that the output will be the symbol \( s_i \).

Notation:
- The source output \( S \) can be modeled by a discrete random variable (RV), where the events should be noted \( \{ S = s_i \} \) and the related probabilities \( \Pr (\{ S = s_i \}) \), or also \( p_d(s_i) \), for the probability mass function of \( S \). But we will most often use abbreviated notations \( s_i \) and \( p(s_i) \).
- In case of experiment with more than one outcome, \( (s_i, u_j) \) will still correspond to the event \( \{ S = s_i \} \cap \{ U = u_j \} \) taken in a joint sample space of dimension \( N_S \times N_U \), where \( S \) and \( U \) are 2 RV (with respective dimensions \( N_S \) and \( N_U \)).

Information content of an outcome
The amount of information gained after observing the event \( S = s_i \) (the symbol emitted by the source \( S \) for example) which occurs with probability \( p_i \) is defined by:

\[
h(s_i) = - K \cdot \log(p_i)\]

where \( K \) is a real positive constant that depends on the chosen unit.

Properties: the function \( f = \log \) has been chosen because it only (among continuous monotonic functions) exhibits the two following important properties:

- \( h(s_i) > h(s_j) \) for \( p_i < p_j \), that is to say \( h(p_i) = f(p_i) \) is a decreasing function of the event probability \( p_i \).
- Additivity: for independent events: so that the observation of two statistically independent events (emission of two symbols for example) \( s_{i_1} \) and \( s_{i_2} \) provides a cumulative information:

\[
h(s_{i_1}, s_{i_2}) = h(s_{i_1}) + h(s_{i_2})
\]

Indeed, independence leads to \( p(s_{i_1}, s_{i_2}) = p_{i_1} \times p_{i_2} \) and then to \( f(p(s_{i_1}, s_{i_2})) = f(p(s_{i_1})) + f(p(s_{i_2})) \).

Moreover:
- The defined amount of information is always a positive quantity.
- If the source emits the symbol \( s \) with probability \( p = 1 \), the associated quantity of information is zero: \( h(s = 0) = 0 \) for the certain event.

Unit of information
Choose the constant \( K \) to choose one information Unit. This is equivalent to choose the base \( b \) of the logarithm:

\[
h(s_i) = - \log_b(p_i), \text{ with then } K = 1 / \ln(b),
\]

Different choices for different units: base \( b = e \Rightarrow \text{natural unit (nit): } b = 10 \Rightarrow \text{decimal unit or Hartley,...}

Standard choice today (used in the course) is to use a logarithm of base \( b = 2 \):

\[
\Rightarrow \log \text{binary: } \log_2(x) = \log \frac{x}{\ln(2)} = \ln(x) / \ln(2).
\]

\[
h(s_i) = - \log_b(p(s_i)) \text{, unit } Sh \text{ (Shannon)}
\]

The resulting Unit Information is called the “Shannon” (Sh), and is part of the International System of Units.

Note: initially (initial paper of C. Shannon, old books,...) the unit Sh was referred to as “bit” for “binary unit”, but it may be confused with the binary digit (binary alphabet symbol) that carries 1 Sh information only if the two possible states are equiprobable (\( p(0) = p(1) = \frac{1}{2} \)).

Examples and remarks:
- for a discrete binary source, with alphabet \( As = \{ 0, 1 \} \):
  - if \( p(0) = p(1) = \frac{1}{2} \) \( \Rightarrow \ h(0) = h(1) = 1 \text{ Sh} \)
  - if \( p(0) = 0.2 \) and \( p(1) = 0.8 \) \( \Rightarrow \) the observation of a 0 corresponds to an information content \( h(0) = 2.32 \text{ Sh} \).
  - the observation of a 1 leads to \( h(1) = 0.32 \text{ Sh} \)

- for a source \( S \), with \( N \) equiprobable messages \( : h(s_i) = - \log(1/N) = \log(N) \)
- information gained by a decimal number, when the 10 number are equiprobable: \( \log(10) = 3.32 \text{ Sh} \)
- if \( N = 2^n \) (built from n-bit) \( \Rightarrow h(s_i) = n \text{ Sh} \)
I.3 Entropy or Average information content of a Random Variable (or discrete memoryless source)

I.3.a) Discrete Memoryless Source (dms, or « simple » source) $S$ :

Each interval of time, a (stationary) discrete source produces one symbol taken in the N-size Alphabet $A_s = \{ s_1, s_2, \ldots, s_N \}$ with the fixed set of probabilities $\{ p_1, \ldots, p_N \}$. Moreover, we shall assume then that successive symbols are generated independently and with the same distribution. Such sources are called discrete memoryless sources (dms), or simple sources.

---

**Model** : a discrete (stationary) source can then be modeled as a sequence of random variables (with time index $i = 1, 2, \ldots, m$) called also stochastic process $\Sigma m = S_{[1]}, S_{[2]}, \ldots, S_{[m]}$, in which all RV $S_{[i]}$ are generated with the same distribution defined by the set $A_s$ and probability mass function (or set of probabilities) $\{ p_n \}_{n=1\ldots N}$. In the case of a dms, all $S_{[i]}$’s are moreover generated independently.

And for one given trial of the source, $\sigma m = s_\alpha_1, s_\alpha_2, \ldots, s_{\alpha_m}$ with $1 \leq \alpha_i \leq N$ and $s_{\alpha_i} \in A_s$, the probability of the given sequence $\sigma m$ is simply given by $p(s_{\alpha_1}, s_{\alpha_2}, \ldots, s_{\alpha_m}) = p_{\alpha_1} \times p_{\alpha_2} \times \ldots \times p_{\alpha_m}$, thanks to the independence.

Vocabulary Note: the word "symbol" refers here the elements of the source. Depending on context, these elements can be letters, words of several letters..., and we may use the appropriate vocabulary.

I.3.b) Entropy of the discrete memoryless source $S$ : average information content per source symbol :

$$H(S) = E\{ h(S) \} = - \sum_{n=1}^{N} p_n \cdot lb(p_n) \quad \text{in Sh/symb}$$

The entropy sets also the **average uncertainty** in terms of probability of the random experiment : $S \rightarrow \{ s_n \}$.

Notes :

- $p \times lb(p) = 0$ if and only if $p = 1$ or $p = 0$ (in limit).
- $H(S)$ depends only on the probabilities on the symbols in the alphabet $A_s$ of the source, and not on the specific values of the set $A_s$. Note that there exists a slight abuse of notation : $H(S)$ is a number, and not a function of the random process $S$, as one would expect.
- When it is convenient, we may interchangeably write the entropy $H(S)$ or $H(p_1, p_2, \ldots, p_N)$, or also $H_S(p_1, p_2, \ldots, p_N, i)$, where $(p_1, p_2, \ldots, p_N)$ is the set of probabilities, and $N$ is the radix of the alphabet.
- We have obviously the same computation formula for $H(S)$ if $S$ is not a dms source (a sequence of RV) but just one RV characterized by the ensemble $\{ A_s, \{ p_n \}_{n=1\ldots N} \}$. Entropy is then a measure of uncertainty of the RV. For memoryless sources, the (average) entropy $H(S)$ is merely the entropy of each (successive) symbol.
- **Appendix** : But in case of sources with memory, the (average) entropy (or entropy rate per channel use) of the source will be defined (in section I.5) as the joint entropy of the successive symbols normalized by the number of successive symbols, or equivalently (for a stationary source) as the conditional entropy of one symbol given all the previous generated symbols.

**Example for a binary (dms) source** : $p = \text{probability of "0"}, \ 1-p = \text{probability of "1"}$. 

$H(S) = H_2(p) = - p . lb(p) - (1-p) . lb(1-p)$

$H_2(p)$ is the binary entropy function (Cf figure 2), frequently encountered in IT problems.

- Entropy is maximized at 1 sh per trial (1 Sh/symb) when the 2 possible outcomes (binary symbols) are equally probable. One binary item (or bit) vehicle 1 Sh only if the 2 possible states are equiprobable ($p = 0.5$).
- The average information amount $H(S)$ tends to zero when one of the two symbols becomes much more frequent (or rare) than the other.
Other examples:
- Alphabet with \( N \) equiprobable letters: \( H(S) = H_{[N]}^{eq} = -\sum_{i=1}^{N} \frac{1}{N} \log_2 \left( \frac{1}{N} \right) \) Sh/symb
  \( \Rightarrow \) for a radix of the alphabet \( N = 4 \), we have \( \Rightarrow H = 2 \) Sh/symb
- French alphabet: assuming 27 equiprobable letters \{a, ..., z, _ (space character) \}, \( H. = \log_2(27) = 4.75 \) Sh/symb.
  But actually, in a French text, successive letters are not independent (not a dms) and not equiprobable. An appropriate measure of \( H. \) would lead about 1 to 2 Sh/symb: additional constraints \( \Rightarrow \) decrease entropy!

The below fundamental Gibbs’ inequality will be a very useful lemma for the course of information theory:

**Gibbs’ inequality:**

Consider two probability distributions \( \{p_1, p_2, ..., p_K\} \) and \( \{q_1, q_2, ..., q_K\} \) that are defined over the same alphabet of \( K \) elements \( A = \{ a_1, ..., a_K \} \): (i.e. with \( \sum_{i=1}^{K} p_i = \sum_{i=1}^{K} q_i = 1 \), and \( p_i, q_i \in [0;1] \))

- there is inequality:
  \[ \sum_{k=1}^{K} p_k \cdot \log_2 \left( \frac{q_k}{p_k} \right) \leq 0 \]  
  (i.e. \( -\sum_{k=1}^{K} p_k \cdot \log_2 (p_k) \leq -\sum_{k=1}^{K} p_k \cdot \log_2 (q_k) \))

- where equality holds only if: \( p_k = q_k \; \forall k = 1 ... K \)

This can be proven by making use of the property of the logarithm: \( \forall x \in \mathbb{R}^+, \log(x) \leq x-1 \) (equality holds only at \( x = 1 \)) with \( x = q_k / p_k \), next multiplying by \( p_k \), and then summing for all \( k \).

**Some properties of entropy:**
- continuity: entropy \( H(S) = H(p_1, p_2, ..., p_N) \) is a continuous function over each variable \( p_i \) on \( [0, 1] \)
- symmetry: from all the variables \( p_i \): \( \forall i, j H(p_1, ..., p_i, p_j, ..., p_N) = H(p_1, ..., p_j, p_i, ..., p_N) \)
- lower and upper bounds: \( H(S) \) is always positive and upper-bounded:

\[
0 \leq H(S) \leq \log(N)
\]

Exercice: prove these lower and upper bounds and their relative necessary and sufficient conditions for the set of probabilities:

1- \( H(S) = 0 \iff p_k = 1 \) for some \( k \), and the remaining probabilities in the set are all zero.
2- \( H(S) = \log(N) \iff p_k = 1/N \) for all \( k = 1 ... N \), which corresponds to an alphabet with an uniform probabilities distribution.
(Demonstration for the upper-bound from Gibbs’ inequality applied to \( q_i = 1/N \) and to \( p_k \).

\( \Rightarrow \) an upper bound of \( H(S) \) is \( H_{[N]}^{eq} \) \( \stackrel{def}{=} \log(N) \): \( H(S) \) is therefore maximum when successive independent symbols of the (dms) source are equiprobable, \( p_k = 1/N \), which corresponds to maximum uncertainty.
I.3.c) Redundancy of a source $S$ : fractional difference between $H(S)$ and its maximum possible value $H_{eq}^{[N]}$ (that could be allowed with the same radix of alphabet, $N$)

$$R(S) = 1 - \frac{H(S)}{\log(N)}$$

We get $0 \leq R(S) \leq 1$. The redundancy allows to assess the degree of use of the alphabet by the source (full use when $R(S) = 0$).

I.3.d) Information Rate (per second) of a source $S$ :

*Entropy* expresses the average amount of information per symbol of a discrete source $S$. We can also define the average amount of information per unit of time, that is $Sh/second$, which is named in the course “Information Rate” (or also “Entropy per second”):

**Information Rate** :

$$H_t(S) = H(S) \cdot D(S) \text{ in } Sh/sec$$

where $D(S)$: symbol rate (symb/sec)

Notes: do not confuse the *information rate* $H_t(S)$ (in Sh/sec) with the (literal equivalent) bit rate $D_b(S)$ (in bit/sec).

We always have : $H_t(S) \leq D_b(S) = D(S) \cdot \log(N)$ since $H(S) \leq \log(N)$

**example information rate** : with binary alphabet ($N = 2$) and $D_b(S) = 34 \text{ Mbit/sec}$

- Equi-probable binary alphabet ($p_1 = p_2 = 0.5$) $=>$ $H(S) = 1 \text{ Sh/bit}$, $H_t(S) = 34 \text{ MSh/sec}$, redundancy $R(S) = 0$

- binary alphabet such as ($p_1 = 0.2$; $p_2 = 0.8$) $=>$ $H(S) = 0.72 \text{ sh/bit}$, $H_t(S) = 24.5 \text{ MSh/sec}$, redundancy $R(S) = 28%$

I.3.e) Extension of a discrete source :

Consider an original source $S$ characterized by an $N$-size Alphabet $A_S = \{s_1, s_2, ..., s_N\}$ and a set of probabilities $\{p_1, ..., p_N\}$. It will be useful to consider blocks rather than individual symbols, with each block consisting of $k$ successive symbols. We may view each such block as being produced by a $k$-th order extended source denoted $S_k$, with an alphabet $A_{S_k}$, that has $N^k$ distinct blocks (or word of $k$ letters): $x_j = s_{j1} s_{j2} ... s_{jk}$, for $j = 1 ... N^k$ and all $s_{ji}$ in $A_S$.

**Extension of a discrete memoryless source and Entropy** :

for a dms, the probabilities of the words are: $p(x_j) = p(s_{j1}) \cdot p(s_{j2}) ... p(s_{jk})$.

the entropy of the $k$-th order extension of the dms is : $H(S^k) = k \cdot H(S)$, in $Sh$ / word of $k$ letters

This property will be immediately deducted from the upcoming results about dependence between 2 sources (I.4).

**exercise**: We consider alphabet with 3 letters A, B, C and respective probabilities : $p_A = 0.7$; $p_B = 0.2$; $p_C = 0.1$;

* source S1: emits successively independent letters,
* source S2: emits successively independent words, each word is built from 2 independent letters,
* source S3: emits successively independent words, each word is built from 2 non-independent letters.

The probabilities of the 9 possible words are given :

$p_{AA} = 0.6$; $p_{AB} = 0.1$; $p_{AC} = 0$; $p_{BA} = 0.06$; $p_{BB} = 0.1$; $p_{BC} = 0.04$; $p_{CA} = 0.04$; $p_{CB} = 0$; $p_{CC} = 0.06$.

Compute the entropies $H(S1), H(S2)$ and $H(S3)$.
I.4 Various entropies between two Random Variables and mutual information:
Advertisement: we consider in this part the entropies and related functions between two discrete R.V. X and Y, to measure their degree of resemblance. These concepts are especially important when dealing with one source with memory (X and Y are then not independent and represents symbols of two different time). In other hand, X and Y might be interpreted as input / output of a noisy channel, as in Ch2.III. More basically, as for the entropy, the definition of the new functions will directly hold if X and Y are two discrete memoryless source (sequence of independent RV), disseminating messages more or less similar.

Let X and Y be 2 discrete Random Variable with respectively two sample space or alphabets Ax = \{x_i, x_2, ..., x_N\} and Ay = \{y_1, y_2, ..., y_M\}, and two set of probabilities Px = \{p(x_1), p(x_2), ..., p(x_N)\} and Py = \{p(y_1), p(y_2), ..., p(y_M)\} .

(X, Y) may be regarded as a virtual source emitting at one instant a two-letter word (x_i, y_j).

I.4.a) Background : joint, marginal, and conditional probabilities

• joint probabilities: the couple of RV (X,Y) is defined on a joint sample space Ax∩Ay (cardinal product) with cardinal N×M, with a set of N×M joint probabilities joint p(x_i, y_j) .

=> Pr (X = x_i; Y = y_j) = Pr (X = x_i ∩ Y = y_j), denoted in abbreviated form p(x_i, y_j).

probability that the word (x_i, y_j) is at the output of the virtual source.

• marginal probabilities (or also a prior probabilities): p(x_i) = \sum_{j=1}^{M} p(x_i, y_j) and p(y_j) = \sum_{i=1}^{N} p(x_i, y_j)

=> p(x_i): probability that the letter x_i be the first letter of the word (regardless of what happens on the second letter).

we have therefore relationships: \sum_{i=1}^{N} \sum_{j=1}^{M} p(x_i, y_j) = \sum_{i=1}^{N} p(x_i) = \sum_{j=1}^{M} p(y_j) = 1

• conditional probabilities: p(x_i / y_j) (or also a posteriori or transition probabilities)

p(x_i / y_j) is the probability that the event X = x_i will occur given the knowledge that the event Y= y_j has already occurred (also denoted Pr (X = x_i | Y= y_j) in more rigorous notation). When we have the knowledge that Y= y_j is happened, the set of possible events (word (x_i, y_j), originally cartesian product Ax × Ay of cardinal N×M has been reduced to the set of N events Ax × \{ y_j \}.

The new set of probabilities are the N conditional probabilities: \{ p(x_1 / y_j), p(x_2 / y_j), ..., p(x_N / y_j) \}.

The relationship between joint and marginal probabilities is: p(x_i / y_j) = \frac{p(x_i, y_j)}{p(y_j)}

Exercise: set the fundamental total probability axiom (p(y_j) in function of p(y_j / x_i) and p(x_i))

I.4.b) Joint entropy H(X, Y)

The average uncertainty (or amount of average information by word) of the pair (X, Y) is given by joint entropy:

H(X,Y) = E\{ h(X,Y) \} = -\sum_{i=1}^{N} \sum_{j=1}^{M} p(x_i, y_j) . lb \left( p(x_i, y_j) \right)

• If X and Y are independent, sum of the marginal entropies: H(X, Y) = H(X) + H(Y)
• if X = Y, then H(X, Y) = H(X) = H(Y)
• General case: joint observation of (X, Y) brings less information that the sum of the information made by separated observations:

0 ≤ H(X,Y) ≤ H(X) + H(Y)

Proof: using Gibbs’inequality: with p_k = p(x_i, y_j), q_k = p(x_i).p (y_j), for the K = N×M values of k. Actually, H(X,Y) ≥ Max \{H(X) ; H(Y)\}
I.4.c) Conditional entropy $H(X / Y)$:
Average remaining uncertainty (or ambiguity) in $X$ after knowledge of $Y$ (or amount of information remains to acquire for $X$ when $Y$ is known) is given by the conditional entropy of $X$ given $Y$:

$$H(X / Y) = - \sum_{i=1}^{N} \sum_{j=1}^{M} p(x_i, y_j) \cdot \log(p(x_i / y_j))$$

It corresponds to the expected value of $g(X, Y) = - \log\{ p(X | Y) \}$, which can be computed in summing with the joint probability mass function $p(x_i, y_j)$ of the pair $(X,Y)$.

This expression can also be obtained by returning to the elementary information associated with $x_i$ conditionally to $y_j$:

$$h(x_i / y_j) = - \log(p(x_i / y_j))$$

- uncertainty in $X$ when $Y$ is known and equal to $Y = y_j$:
  $$H(X / y_j) = \sum_{i=1}^{N} p(x_i, y_j) \cdot h(x_i / y_j) = - \sum_{i=1}^{N} p(x_i, y_j) \cdot \log(p(x_i / y_j))$$

- uncertainty in $X$ knowing $Y$, averaged over all possible values of $Y$:
  $$H(X / Y) = \sum_{j=1}^{M} p(y_j) \cdot H(X / y_j)$$

Using the definition of conditional probabilities leads to the first definition of $H(X / Y)$.

Relationships between the entropies:

$$H(X / Y) = H(X, Y) - H(Y)$$

proof from the definition of $H(X, Y)$ and using $p(x_i, y_j) = p(y_j) \cdot p(x_i / y_j)$.

Interpretation as a chain rule, and generalize to the case of $n$ R.V. with $n > 2$ ?

Special Cases:
- If $X$ and $Y$ are independent, then $H(X / Y) = H(X)$
- If $X = Y$, then $H(X / Y) = 0$

General case: from the bounds on $H(X, Y)$, we can deduce the following bounds:

$$0 \leq H(X|Y) \leq H(X)$$

$\Rightarrow$ conditional entropy $H(X|Y)$ is less than or equal to the information content provided by $X$, since knowledge of $Y$ may reduce uncertainty on $X$.

I.4.d) mutual information $I(X ; Y)$: \textit{(average) amount of information shared by $X$ and $Y$ in Sh/symb} That is the amount of information that one RV contains about the other. $I(X; Y)$ is a measure of the gap to the independence between $X$ and $Y$, with the equivalent definitions:

$$I(X ; Y) = H(X) + H(Y) - H(X,Y)$$

$$I(X ; Y) = H(X) - H(X/Y) ; \quad I(X ; Y) = H(Y) - H(Y/X)$$

- If $X$ and $Y$ independent, then $I(X ; Y) = 0$;
- If $X = Y$, then $I(X ; Y) = H (X) = H (Y)$
- General case:

$$0 \leq I(X ; Y) \leq H(X) ; \quad and \quad 0 \leq I(X ; Y) \leq H(Y) ;$$

the Venn diagram summarizes the definition of the mutual information as well as relations between different entropies that have been defined in paragraph I:
The calculation of the information shared between X and Y may be done directly from the marginal and joint probabilities:

\[
I(X;Y) = \sum_{i=1}^{N} \sum_{j=1}^{M} p(x_i, y_j) \cdot \log \left( \frac{p(x_i, y_j)}{p(x_i) p(y_j)} \right)
\]

**Appendix:**

1) it is sometimes considered *in particular in communication* (X : input, Y: output of a channel) that the (average) mutual information \( I(X;Y) \) is more important than the entropy. That is why in some books, one begins to set \( I(X;Y) \) from (elementary) mutual information \( i(x, y) \) and the definition of entropy can be deducted:

- \( i(x_i, y_j) = \log \left( \frac{p(x_i, y_j)}{p(x_i)} \right) = i(y_j, x_i) \Rightarrow \) information provided about the event \( X = x_i \), about the occurrence of the event \( Y = y_j \),
- \( I(X ; Y) = E\{ i(x_i, y_j) \} \) and then \( h(x_i) = i(x_i, y_j) \) is the self information of the event, and \( H(X) = I(X ; X) \).

2) \( I(X, Y) \) can also be interpreted as the “relative entropy” (or Kullback Leibler distance) between the joint probability \( p(x, y) \) and the product distribution \( p(x) p(y) \)

3) Extension : if we consider 3 discrete R.V. \( X, Y, \) and \( Z \), the **Conditional mutual information of \( X \) and \( Y \) given \( Z \)** is defined by:

\[
I(X;Y|Z) = H(X|Z) - H(X|Y, Z) = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{L} p(x_i, y_j, z_k) \cdot \log \left( \frac{p(x_i, y_j|z_k)}{p(x_i|z_k)p(y_j|z_k)} \right)
\]

**Application Exercise**

Consider the case of two binary discrete memoryless sources \( X, Y \) such as:
- \( X \) emits symbols “0” and “1”, with equiprobability.
- the emission of \( Y \) depends on \( X \) in this way : "1" is emitted if \( X \) emits “0”, else “0” or “1” can be emitted with equiprobability

Compute various probabilities, entropies and mutual information for the system \( (X, Y) \)?
I.5 Markov Source: an introduction

Until now: source without memory (independence of successive symbols). In practice, there is often time dependence between the symbols \( x[n] \) issued by the source at time indices \( n \).

(1.5.a) Entropy of a Markov Source (of order \( M = 1 \)):

Let \( X \) be a source with \( N \) possible symbols \( \{ x_1, x_2, ..., x_N \} \), where the state \( X[n+1] \) at time “\( n+1 \)” depends on the state \( X[n] \) at the previous instant “\( n \)” according to the transition matrix \( M \):

\[
\Pr( X[n+1] / X[n] )
\]

<table>
<thead>
<tr>
<th>( X[n+1] )</th>
<th>( x_1 )</th>
<th>( ... )</th>
<th>( x_N )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_j )</td>
<td>( p_{1/j} )</td>
<td>( p_{N/j} )</td>
<td></td>
</tr>
<tr>
<td>( ... )</td>
<td>( )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( x_N )</td>
<td>( p_{1/N} )</td>
<td>( p_{N/N} )</td>
<td></td>
</tr>
</tbody>
</table>

With \( p_{i/j} = \Pr(X[n+1] = x_i | X[n] = x_j) \) and then: \( \sum_{i=1}^{N} p_{i/j} = 1 \), \( \forall j = 1, ..., N \)

The dependency on the past is summarized to that of the last state that is reached.

(for a more accurate formulation, \( X[1] \rightarrow X[2] \rightarrow ... \rightarrow X[n] \rightarrow X[n+1] \) forms a Markov chain, see Appendix 1.5.B)

- At time “\( n \)” : system is characterized by a probability set \( P[n] = \{ p_{1/n}, p_{2/n}, ..., p_{N/n} \} \) for different states \( \{ x_1, x_2, ..., x_N \} \), with \( p_{j/n} = \Pr \{ X[n] = x_j \} \)

- At time “\( n+1 \)” : the set of probabilities changes \( P[n+1] = \{ p_{1/n+1}, p_{2/n+1}, ..., p_{N/n+1} \} \), with

\[
p_{i/n+1} = \sum_{j=1}^{N} p_{i/j} \cdot p_{j/n}, \quad \forall i = 1, ..., N
\]

which gives in algebraic formulation:

\[
P_{[n+1]} = P[n] \cdot M
\]

- **Stationary state of the system**: It can be proved that if the matrix is such that “all states communicate effectively” (i.e. if it is possible to go with positive probability from any state of the Markov chain to any other state in a finite number of steps), the system reached (as \( n \rightarrow +\infty \)) a stationary distribution (the set of probability \( P^* = \{ p_1, p_2, ..., p_N \} \) is then unchanging from on time “\( n \)” to the next “\( n+1 \)”).

Notes:

- the stationary distribution \( P^* \) is unique and reached (more or less quickly) whatever the starting set of probabilities \( P_{[0]} \). Of course, if the initial state of a the Markov chain is drawn according to the stationary distribution \( P_{[0]} = P^* \) then the Markov chain forms a stationary process.

- \( P^* \) is therefore solution of the linear system: \( P = P \cdot M \), with additional condition \( \sum_{i=1}^{N} p_i = 1 \).

- **Entropy (or Entropy rate) of the Markov source**: the per symbol entropy of the source, or “entropy rate” (denoted \( H(X) \) or \( \overline{H}(X) \) and defined in appendix) plays the role of entropy for sources with memory. When the source is with memory, the knowledge of previous symbols makes less unexpected symbol which will be issued. There is thus an entropy reduction resulting from taking into account the past issued symbols. For a first order stationary Markov source, \( \overline{H}(X) \) is equal to the conditional entropy of the source at one time, given the previous state (denoted \( H_{M=1}(X) \)). Then, for an order 1 Markov source, the entropy can be computed when the stationary distribution is reached, as:

\[
H_{M=1}(X) = H( X[n+1] | X[n] ) \quad \text{with} \quad H(X[n+1] | X[n]) = - \sum_{j=1}^{N} \sum_{i=1}^{N} (p_{i/j} \cdot p_j) \times \log(p_{i/j})
\]
Which is also equivalent, according to the conditional entropy definition (Cf I.4), to the average value of the uncertainty in $X$, given the different possible previous states:

$$H_{M=1}(X) = \sum_{j=1}^{N} p_j H(X_{[n]}/X_{[n-1]} = x_j) \quad \text{with} \quad H(X_{[n]}/X_{[n-1]} = x_j) = -\sum_{i=1}^{N} p_{i,j} \cdot \log(p_{i,j})$$

Appendix note for the most general case of a (not necessary Markov nor stationary) source with memory:

Let the source defined by a sequence of R.V. $X_{[1]}, X_{[2]}, \ldots, X_{[n]}$ with possible dependency between R.V. but not necessary a Markov process, nor a stationary process.

- **Entropy Rate**: it is basically a measure of the uncertainty per output symbol of the source. The general definition of the (per symbol) entropy, or “entropy rate” is: $$\overline{H}(X) = \lim_{n \to \infty} \frac{1}{n} H(X_{[n]} | X_{[1]}, X_{[2]}, \ldots, X_{[n]})$$, in Sh/symbol.

- For **stationary processes**, the entropy rate can also be computed as the conditional entropy of the last random variable given the past. Then $$\overline{H}(X) = \lim_{n \to \infty} H(X_{[n]} | X_{[1]}, X_{[2]}, \ldots, X_{[n-1]})$$

- For a stationary Markov chain, the entropy rate is reduced to $$\overline{H}(X) = H(X_{[n]} | X_{[n-1]})$$, where the conditional entropy is calculated using the given stationary distribution.

(1.5.b) **Appendix 1: Markov Chain and "Data Processing Theorem"**:

The source with memory (order 1) introduced, (In 1.5.a) represents actually a Markov chain: $X_{[1]} \rightarrow X_{[2]} \rightarrow \cdots \rightarrow X_{[n]} \rightarrow \cdots$

**Markov chain**: Let $X, Y, Z$, be 3 discrete random variables with finite alphabets. They are said to form a Markov chain in the direction $X \rightarrow Y \rightarrow Z$ if and only if $Z$ is independent of $X$ conditionally to the knowledge of $Y$:

- $\Pr(Z = z | Y = y, X = x) = \Pr(Z = z | Y = y)$

Or in an equivalent way

- $\Pr(Z = z, Y = y, X = x) = \Pr(X = x)$. $\Pr(Y = y | X = x)$. $\Pr(Z = z | Y = y)$

**Data Processing Theorem**: If $X \rightarrow Y \rightarrow Z$ form a Markov chain, then: $I(X; Z) \leq I(X; Y)$

No processing on $Y$ can increase the information that $Y$ contains about $X$.

1.5.c) **Annexe 2**: extension to a Markov source with order $M$: symbole sent at time « $n$ » depend on the $M$ previous symbols, send at time « $n-I$ » to « $n-M$ ». Generalization of the case $M=1$.

The source $X$ est is known with the knowledge of the transit (or a priori conditional) probabilities: $\Pr(x_i / X_j) = \Pr(x_{[n]} = x_i / X_j)$, with $X_j$ the state for the $M$ previous symbols.

$$H(X / X_j) = -\sum_{i=1}^{N} p(x_i / X_j) \cdot \log(p(x_i / X_j)) = -\sum_{i=1}^{N} p(X_{i} / X_{j}) \cdot \log(p(X_{i} / X_{j}))$$

with $X_j = (x[n], \ldots, x[n-M+1])$; $X_{j} = (x[n-1], \ldots, x[n-M])$

The entropy (or entropic rate) of the (stationary) source can be computed by using the conditional probabilities.
\[ H_n(X) = \cdot \sum_{i=1}^{N^M} p(X_i) \cdot H(X / X_i) \]

Which depends on the transit probabilities \( p(X_i, X_j) \), contained in the transit matrix \( T(N^M \times N^M) \) where \( T_{ij} = p(X_i, X_j) \).

It can be proved that \( H_0(X) \) is a decreasing sequence so that:

\[ 0 \leq H_0(X) \leq H_{m-1}(X) \leq \ldots \leq H_1(X) \leq H_0(X) \leq \text{lb}(N) \]

With \( H_0(X) \) : entropy of the memoryless source.

*The main tools of Information Theory have been introduced, we now will see how they may apply to the problems of source and channel coding, in order to obtain theoretically achievable limits.*
Exercice 1 : Dice
Let the random experiment « rolling of two fair dice », and the following Random Variable (R.V.) :
1. \( P_1 \) assigned to 0 for even outcome of die 1, else to 1 for odd outcome of die 1.
2. \( X_1 \) represents the outcome (die 1).
3. \( X_2 \) represents the outcome (die 2).
4. \( X_{12} \) represents couple of outcomes (die 1, die 2).
5. \( \Sigma \) represents the sum of both outcomes (die 1 + die 2).

Compute and Comment :
A) the amount of information associated to the following events :
\[ \{ X_1 = 4 \}; \{ P_1 = 0 \}; \{ X_2 = 2 \}; \{ \Sigma = 6 \}; \{ X_{12} = (4,2) \}; \{ X_{12} = (4,2) \mid \Sigma = 6 \} \]

B) the Entropies of the R.V. \( X_1 \); \( P_1 \); \( X_{12} \); \( \Sigma \)

Answers : \( h_{\{X_1 = 4\}} = 2.585 \text{ Sh} \); \( h_{\{P_1 = 0\}} = 1 \text{ Sh} \); \( h_{\{\Sigma = 6\}} = 2.845 \text{ Sh} \); \( h_{\{X_{12} = (4,2)\}} = 5.17 \text{ Sh} \); \( h_{\{X_{12} = (4,2) \mid \Sigma = 6\}} = 2.325 \text{ Sh} \); \( H(X_1) = 2.585 \text{ Sh/die} \); \( H(P_1) = 1 \text{ Sh/state} \); \( H(X_{12}) = 2H(X_1) \); \( H(\Sigma) = 3.2744 \text{ Sh/number} \)

Exercice 2 : extension for 3 R.V., chain rule, and Data Processing Theorem
Let 3 discrete Random Variables \( X, Y, Z \) with finite alphabet. Prove and comment (by a diagram) the 3 following properties (for memory : \( Pr(X = x, Y = y \mid Z = z) = Pr(X = x \mid Y = y, Z = z)Pr(Y = y \mid Z = z) \)) :

(1) \( H(X, Y, Z) = H(X) + H(Y \mid X) + H(Z \mid Y, X) \)
N.B : via the conditional entropy \( H(Y, Z \mid X) = H(Y \mid X) + H(Z \mid Y, X) \)

(2) \( I((X,Y) ; Z) = I(X ; Z) + I(Y ; Z \mid X) \)
N.B. : via the definition of conditional mutual information : \( I(Y ; Z \mid X) \overset{\text{def}}{=} H(Y \mid X) - H(Y \mid Z, X) \)

(3) « Data Processing Theorem » : If \( X \rightarrow Y \rightarrow Z \) form a Markov Chain, then \( I(X ; Z) \leq I(X ; Y) \).
(i.e. \( P_{X,Y,Z}(x,y,z) = P_X(x)P_{Y\mid X}(y\mid x)P_{Z\mid Y}(z\mid y) \leftrightarrow X \text{ and } Z \text{ conditionally independant given } Y) \)
N.B. : - via the expression of \( I(Y, Z ; X) \) in function of \( I(X ; Z) \) and next \( I(X ; Y) \)
- via \( I(X ; Z \mid Y) = 0 \) (to be checked for a Markov chain \( X \rightarrow Y \rightarrow Z \) )
II. Source Coding (Data Compression)

II.1 Introduction

We have seen in the part I.1 than the maximum entropy (\( H(Q) \) if \( Q \) is the size of the alphabet) of a discrete source was achieved when different symbols were equally probable, with additional hypothesis of independency between successive symbols. When the source symbols are not equi-probable or and not independent, the source presents some redundancy, which means that the natural source alphabet is not used optimally (carries a quantity of information less than its potential). We are then interested in more efficient representation of data generated by the discrete source.

- **Major objective of source encoding**: transforming the "source + source encoder" into a "standardized" source:
  - (almost) without redundancy, so that the bit rate is reduced (\( D_b(U) \leq D_b(S) \) bit/sec) for a real time transmission (assumed by default).
  - without Loss of Information (only "Lossless compression method" are considered in the course): The amount of information (in Sh) of all messages to be transmitted is retained after coding.
  Thus, for a real time transmission of the source, the information rate (flow of information per second) is the same after coding: \( H_t(U) = H_t(S) \), in Sh/sec.

N.B: real-time transmission means here that the natural emission duration of the source is respected, equal to the total number of symbols of the source to be transmitted multiplied by the baud-rate of the source \( D(S) \). Example: If the total sequence of source messages results from 1 hour of video output of a digital camera, the emission will last one hour.

- **Other function of source encoding**: necessary conversion between the alphabet of the source \( A_s = \{s_1, s_2, \ldots, s_N \} \) and the alphabet in input of the channel \( A_u = \{u_1, u_2, u_Q \} \). The N-ary source alphabet may be very different depending on the situation, but the input alphabet of the channel is more often binary: \( Q = 2 \).
  Example: to map the alphabet of \( N = 26 \) letters of a text to binary symbols. The direct conversion of the letters into 5-bit binary sequence is possible but not efficient (\( 2^5 = 32 > 26 \)).

It should be noted at this stage (source encoding/decoding study) that ideal channel (noiseless channel) that routes symbols without error is assumed. The hypothesis of a channel without error is however plausible if a "standardized" channel, that means "channel + channel coding" is used (Cf figure 1).

**Vocabulary note**: to avoid confusion in the chapter II, we will reserve the word "symbol" (or elementary symbols, or encoded symbols) to designate the elements resulting from the source coding (elements of the Q-ary alphabet, with most often \( Q = 2 \) for a binary source code). On the other hand, the word "message" will designate the N-ary symbols (also called "letter" or "source letter") issued by the discrete original source, or the N-ary symbols obtained after k-th order extension of the source. The term "code-word" shall designate a sequence of symbols resulting from the encoding of 1 given message.

Figure 4: Adaptation of the source to the channel
II.2 characterization of a source encoding

II.2a) definitions:
- **Source code**: is a mapping from the messages of the source (N-ary letters \{s_i, i=1…N\} \in As first, or later generalization with extended words of k letters, As^k) to the set of finite length strings of symbols from a Q-ary alphabet (code-words or sequences C_i = \{u_l^{(i)}, …, u_{l_i}^{(i)}\} composed of \(l_i\) Q-ary elementary symbols \(u_j^{(i)} \in Au, with i = 1 … N \text{ first, and later i = 1 … N^k}\)). We get a fixed- to variable-length coding.

- **Length** \(l_i\) of the codeword \(C_i\) (in symbols): positive integer equal to the number of elementary symbols in the code-word \(C_i\), used to represent the source message \(s_i\).

The encoding process of the source \(S\) : consists to encode the sequence of source messages (letters or later words of \(k\) letters) by concatenating the code-words corresponding to each message.

⇒ use of variable- length code-words to achieve Data compression.

- **Average code-word length** \(L\) (or (Expected-) length of the code):

\[
L = \sum_{i} p(s_i) \cdot l_i, \text{ in elementary symbols (per code-word, or per message)}
\]

\(L\) indicates the compactness of the code: the more \(L\) is low and the more (equivalent) output bit rate \(D(U)\) will be reduced:

\[
D(U) = D(S).L \Rightarrow D_s(U) = D(S)L.\log Q, \text{ with most often binary symbols (}\log Q = 1\).
\]

\(D(S)\) in message /sec, \(D(U)\) in (elementary) symbols /sec, \(D_s(U)\) in bit /sec

When the source encoding is lossless, there must be the following relationship between the entropies before and after source encoding:

\[
H(U) = \frac{H(S)}{L}
\]

\((H(S) \text{ in Sh/message }, \ H(U) \text{ in Sh/symb. })\)

\(H(U)\) is in agreement with the conservation of information rate: \(D(U) = L.D(S)\) and \(H(U) = H(S)/L \Rightarrow H(U) = H(S)\) \(L\)

We can already infer a lower bound for \(L\) (lossless code):

\[
L \geq \frac{H(S)}{\log Q} = L_{\text{min}}
\]

since the output entropy \(H(U)\) cannot exceed the maximum possible value \(\log Q\) for the (secondary or standardized) source U using Q-ary symbols (Cf I.3).

- **Coding efficiency** of the source encoder (0 ≤ η ≤ 1) is defined as:

\[
\eta = \frac{L_{\text{min}}}{L} = \frac{H(S)}{L.\log Q}
\]

Or also \(\eta = H(U)/\log(Q)\).

We check immediately that efficiency is derived by comparing the equivalent bit rate \(D_s(U) = D(U).\log(Q)\), in bit/sec, to the information rate \(H(U)\), in Sh/sec:

\[
\eta = \frac{H(U)}{D_s(U)}.
\]

- **Code redundancy** is by definition 1 - \(\eta\).

It also corresponds to the redundancy of the secondary source U (source S + source encoding): \(R(U) = 1 - \eta\)

Indeed, by definition, the U redundancy is (Cf I.3): \(R(U) = 1 - H(U)/\log(Q)\)

Anticipating on later, we give an example to illustrate that source coding permits to reduce redundancy and so the bit rate. The main idea is that data compression can be achieved by assigning short code-words to commonly occurring (more probable) messages and longer code-words to less frequent messages.
Example:
construction of a binary (Q = 2) code source for a source delivering N = 4 different letters:

<table>
<thead>
<tr>
<th>Alphabet Source: A</th>
<th>Probabilities: Pr</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1; s2; s3; s4</td>
<td>(0.64; 0.16; 0.1; 0.1)</td>
</tr>
</tbody>
</table>

Example of code C:
<table>
<thead>
<tr>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>110</td>
<td>111</td>
</tr>
</tbody>
</table>

alphabet of output elementary symbols: A_u={0, 1}

code-words Lengths:
l1= 1, l2=2, l3=3, l4=3 symbols

Average code-word length: L = 0.64 x 1 + 0.16 x 2 + 0.1 x 3 + 0.1 x 3 => L = 1.56 symbols

Example:
Sequence of messages:
<table>
<thead>
<tr>
<th>s1</th>
<th>s2</th>
<th>s1</th>
<th>s3</th>
<th>s1</th>
<th>s4</th>
<th>s1</th>
<th>s2</th>
<th>s1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Sequence of coded symbols:
| 0  | 10 | 00 | 0  | 1  | 10 | 0  | 1  | 1  | 0  | 0  | 10 |

=> Cod. C: 17 bits

Characterization of the source S:
-entropy: \( H(S) = 1.5 \text{ sh/letter} \) (< 2 Sh, max. with N = 4)
-redundancy: \( R(S) = 25 \% \)

We assume a baud rate of source S, \( D(S) = 17 \text{ Mmessage/sec} \) or \( D_6(S) = D(S) \cdot \log_2(N) = 34 \text{ Mbps} \)

=> information rate: \( H_t(U) = H_t(S) = H(S) \cdot D(S) = 25.5 \text{ MSh/sec} \)

With the code C, the (average) bit rate in output of the source encoder is: \( D(U) = D(S) \cdot L = 26.52 \text{ Mbit/sec} \)

That is a bit rate reduced by 22% by the code source C.

In output of the encoder:
- entropy: \( H(U) = H(S) / L = 0.962 \text{ sh/ bits} \)
- redundancy: \( R(U) = 1 - H(U) / \log_2(Q) = 3.8 \% \): the redundancy has been well decreased
- coding efficiency: \( \eta = 1-R(U) = 96.2 \% \)

We check \( L = 1.56 \leq L_{\text{min}} = H(S) / L = 1.5 \text{ symbols} \)

II.2b) Functional Requirements for codes

Let us consider some restrictions on variable length code to allow decoding without ambiguity, so that any original messages sequence can be reconstructed perfectly from the encoded sequence. A code with this necessary feature is called “uniquely decodable”. Moreover, we are generally interested in a special class of uniquely decodable codes, called “instantaneous codes”.

• uniquely decodable codes: for each finite length source sequence of N-ary messages, the corresponding sequence of Q-ary coded symbols is different from the sequence of coded symbols corresponding to any other source sequence. Such a code is therefore:
  1. “non-singular”: each source letter is mapped to a different non-empty codeword, i.e. the mapping from source letters to codewords is one-to-one (otherwise lossy coding)

    example: code C’ = {0, 10, 110, 110} is not regular
  2. “distinguishable” (or “separable”): the beginning and the end of the successive code-words is found without ambiguity when observing a sequence of encoded symbols (no matter what the original sequence of letters were).

    example: code C’ = {1, 10, 110, 111} is non-singular but not “distinguishable”

    received sequence “110” => s1s2 or s3?

Sufficient conditions (but not optimum for Data Compression !) to have a “distinguishable” code: use codewords with constant lengths, or use a separator or...
Appendix: necessary and sufficient condition for a code to be “uniquely decodable” is that its extension is non singular. (The extension of a code is the mapping of finite length source sequences to finite length symbols strings, obtained by concatenating for each letter of the source sequence the corresponding codeword produced by the original code).

- **Instantaneous codes**: special class of “uniquely decodable codes” such as the decoding of each successive codeword can be accomplished as soon as the end of its receipt, without waiting for receipt of the beginning of the future codeword. The end of a codeword is immediately recognizable, and therefore there is no need of separator. Instantaneous codes satisfy a restriction (which will be stated) known as the “prefix condition”.

Examples: code $C' = \{0, 01, 011, 0111\}$ is “uniquely decodable” but not “instantaneous” (uses separator “0”). Code $C = \{0; 10; 110, 111\}$ is “instantaneous”

**Prefix Condition**

A code is instantaneous if and only if (iff) no code-word is the prefix of any other code-word.

For this reason, an “instantaneous code” is also called a “prefix code” (or “prefix-condition code”).

=> graphical representation of any code by a (Q-ary) tree, where:
- each node represents a sequence of symbols, depending on the path from the root to this node,
- each node has up to Q branches (children) representing possible values “0”, “1”, ..., “Q-1” of the symbols,
- (the root node has Q branches for the Q possible values of the first symbol of the codewords leading),
- one code-word is represented by a node, and then N nodes are code-words
  (but all nodes are not codewords! All the other nodes have “no sense” for this code).

*Example with $Q = 2$*: root (and possibly each node) is divided into 2 branches (“0”, “1”)

=> prefix condition: each codeword is represented by a leaf (terminal nodes) in the tree,
  (since no codeword can be the descendent of any other codeword).

**Kraft Inequality**:

**Necessary and Sufficiency Condition for the existence of an instantaneous Q-ary code with given code-word lengths**:

For any instantaneous code over an alphabet of size $Q$, the code-word lengths ($l_1, l_2, ..., l_N$) must satisfy the inequality:

$$\sum_{n=1}^{N} Q^{-l_n} \leq 1$$

Conversely, for any given set of codeword lengths that satisfy this inequality, there exists an instantaneous code with these code-word lengths.

Proof:

N.C.: one count the total number of excluded terminal nodes of the full Q-ary tree
- Construct a “full” Q-ary tree with $Q^{l_N}$ terminal nodes (nodes at level $l_N$ = length of the longest codeword), where we assume $l_1 \leq l_2 \leq ... \leq l_N$.
- Prefix condition $\Rightarrow$ choice of a code-word with length $l_i$ eliminates its descendant, and then excludes $Q^{l_N-l_i}$ terminal nodes of the full tree $\Rightarrow$ necessarily: $Q^{l_N-l_i} \leq Q^{l_N}$

S.C. (Converse): For a given set of lengths satisfying the Kraft’s inequality $\Rightarrow$ we can construct the Q-ary tree of an instantaneous code (label the first node of length $l_1$ for codeword 1, and remove its descendant from the tree. Then label the first remaining node of depth $l_2$ as codeword 2, etc.).

**MacMillan theorem**: generalizes the Kraft theorem as a NCS for existence of uniquely decodable code, with the same inequality.

$\Rightarrow$ **Important consequence**: it is always possible to replace an “uniquely decodable code” by an “instantaneous code” with the same set of code-word lengths (and then the same efficiency).
Hence, the class of uniquely decodable codes, although larger, does not offer any further choices for the set of codeword lengths than the class of instantaneous codes.

- Codes “absolutely optimal”: instantaneous codes such that \( L = L_{\text{min}} \).

Equality can only occur when the coding output alphabet \( A_u \) is used with equal probability \( p(u_1) = p(u_2) = \ldots = 1/Q \), and with independence between successive symbols (we get a DMS of symbols after absolutely optimal code, if exists).

This implies necessarily than the probability distribution of the source messages must be:

\[
p(s_i) = p(C_i) = Q^{-l_i}, \quad \text{for } i = 1, \ldots, N.
\]

Such a pmf (where \( l_i \) are integers) is called \( Q \)-adic with respect to \( Q \).

It should be noted that for absolutely optimal codes, Kraft-MacMillan inequality is an equality.

Remarks:
- this result can also be proved directly by Gibbs inequality under constraint of Kraft-MacMillan condition for the codeword lengths (see direct proof of Source coding theorem).
- a \( Q \)-adic pmf, allowing to perfectly match the codeword lengths is a very special (rather meretricious) case.

But how do we match the instantaneous code to an arbitrary source? The answer to this problem lies in applying the source encoder on blocks (or words) of \( k \) letters rather than on individual letters.

II.3 Source Coding theorem (first Shannon theorem)

A source being given, is it possible to get an achievable limit to the lossless data compression (or to the resulting baud-rate) thanks to an appropriate encoding? First Shannon theorem answers in terms of asymptotic property source encoding:

**Shannon's source coding theorem** (or “noiseless coding theorem”)

Let \( S \) be a stationary discrete source \( S \) of alphabet size \( N \), with \( H(S) \) the entropy per letter (or entropy rate in case of source with memory). It is possible to use an instantaneous encoding process (possibly operating on blocks of \( k \) source letters) with a code alphabet of size \( Q \), in such a way that the average codeword lengths (expressed in elementary \( Q \)-ary symbols per letter) is as close as desired to the lower bound \( L_{\text{min}} = H(S)/\log_2(Q) \):

\[
L = L_{\text{min}} + \varepsilon, \quad \forall \varepsilon > 0,
\]

In other words, it is always possible to find a prefix source coding bringing to an (equivalent) binary rate after coding, \( D_b(U) = D(U)/\log_2(Q) \) in bits/sec, as close as desired to the information rate of the source \( H_0(S) = H(S)D(S) \) in Sh/sec:

\[
D_b(U) = H_0(S) + \varepsilon', \quad \forall \varepsilon' > 0.
\]

Thus, there exists an encoding process such as:
- the efficiency of the code is as close as desired to 1,
- entropy in output encoding, \( H(U) \), is as close as desired to \( \log_2(Q) \),
- elementary symbols after coding \( u_i \) are as close as desired to the uniform distribution, with independence between successive symbols.

The key of the theorem lies in the use of large fixed-length blocks of the source output (large value for \( k \)) to perform the coding (i.e. the mapping to variable-length codewords), as revealed by the two fundamental results (steps A and B) given in the proof of the theorem.

**Elements of Proof of the source coding theorem** (in case of Discrete Memoryless Source):

A. first lemma:

For a source without memory \( S \), it is possible to assign codewords to the source letters in such a way that the prefix condition is satisfied and the average length of the codewords, \( L \), satisfies:

\[
\frac{H(S)}{\log_2(Q)} \leq L \leq \frac{H(S)}{\log_2(Q)} + 1
\]

Such construction allows therefore already an approach of optimal encoding messages (with an overhead of 1 symbol).
Proof (exercise): prove this first lemma in using “absolutely optimum” codes properties (easy!).

Other (direct) proof: Gibbs inequality applied with \( p_i = p(s_i) \), and \( q_i = Q^{-1} / \sigma \), with the necessary normalization coefficient \( \sigma = \sum Q^{-l_i} \), leads to:

\[
H(S) = -\sum_{i=1}^{N} p_i \cdot \log_2 (p_i) \leq \sum_{i=1}^{N} p_i \cdot l_i \cdot \log_2 (Q) + \log_2 (\sigma) = L \cdot \log_2 (Q) + \log_2 (\sigma)
\]

Under constraint of Kraft-Mac Millan condition, \( \sigma \leq 1 \), and then:

\[
H(S) - L \cdot \log_2 (Q) \leq \log_2 (\sigma) \leq 0
\]

Firstly, we confirm a proof of the necessary lower bound \( L \geq H(S) / \log_2 (Q) \).
Secondly, we have equality \( H(S) = L \cdot \log_2 (Q) \) if and only if (iff):

- \( \sigma = 1 \) (Kraft-Mac-Millan equality), and
- \( p(s_i) = q_i = Q^{-1} \) (Gibbs equality condition), (1)

that is codewords lengths must be such as:

\[
l_i = \frac{- \log_2 (p_i)}{\log_2 (Q)}\]

Note that this set of lengths satisfies the Kraft-Mac Millan inequality.

The lemma is proved by multiplying (3) by \( p_i \) and by summing over \( i = 1 \) to \( N \).

Remark: the code obtained thanks to the procedure (3) is sometimes called “Shannon code”. It is generally not an optimal code (others codes can often achieve better efficiency), but is enough to prove the lemma, and next the source coding theorem when applied to \( S^k \), the k-th order extension of \( S \) (see B.).

B. Application of the first lemma to \( S^k \) (kth-order extension of the discrete memoryless source \( S \)):

For a source without memory \( S \), it is possible to assign \( N^k \) code-words to the alphabet \( A^k \) containing the \( N^k \) possible blocks of \( k \) source letters in such a way that the prefix condition is satisfied and the average length of the codewords \( L \), satisfies:

\[
\frac{H(S)}{\log_2 (Q)} \leq L \leq \frac{H(S)}{\log_2 (Q)} + \frac{1}{k}
\]

Finally, it is possible to choose the size of blocks, \( k \), large enough so that \( L = L_{\text{min}} + \epsilon, \forall \epsilon > 0 \), which proves the (asymptotic) Shannon’s source coding theorem, for any discrete memoryless source.

Proof:
First lemma applied to \( S^k \) : \( k \)-th order extension of \( S \), with a Q-ary prefix condition code used to encode blocks of \( k \) letters. Note that the entropy of the dms extended source is \( H(S^k) = k \cdot H(S) \), and that average length of the codewords \( L_k \) is expressed in Q-ary symbols by words of \( k \) letters, and then in average \( L = L_k / k \).

Comments:
- the proof of the Shannon’s source coding theorem has been given for discrete memoryless source for mathematical convenience, but can be generalized for any stationary source with memory (and then for Markov sources). In this general case, the lower-bound \( L_{\text{min}} = H(S) / \log_2 (Q) \) is computed from the entropy per letter (or entropy rate), \( H(S) \).
- source coding theorem confirms the major role of the concept of entropy for DATA COMPRESSION field. The entropy \( H \) indicates the (achievable) minimum average number of bits required to (binary) encode a discrete source.
- distinction between the binary information unit (” Sh”), and the symbols of the binary alphabet (often referred to as "bits" or "digits"): this is only after an ideal source encoding (reaching the lower bound in the theorem) that each binary symbol can carry a quantity of information equal to 1 Shannon. In all other cases, it carries a lower amount of information.
Appendix: other interpretation of entropy and source coding theorem, from the notion of “typical sequences set” [Mac03] [Cov03].

II.4 source coding techniques
There exist practical methods to design efficient codes. The most used algorithms are:
Shannon-Fano coding,
Huffman coding,
Lempel-Ziv algorithm,
...
Applied in: fax systems, commands "pack", "compress", "gzip" in UNIX....

We present in the following two instantaneous source-coding algorithms, assuming that the letters of the source are coded individually (k=1). These algorithms can of course be used to code the k-th order extension of the source (then 1 source message = 1 word of k letters), what is moreover necessary to improve efficiency, but at the price of an increased complexity.

II.4.a) Shannon-Fano codes
We know to get the length L minimum, the lengths of the codewords should be \( l_i = -\frac{\log p_i}{\log Q} \) if the results were integers, which is usually not the case. The present algorithm will attempt to approach this condition (giving the same length for nearly the same probability of occurrence), in an easy (sub-optimal) procedure to implement.

Algorithm is presented in the particular case of a binary code \((Q = 2)\) but is easily generalized for \(Q > 2\).

It is based on the following procedure, which can be represented using a tree:

1) Arrange the source messages such as the probabilities are in the descending order
2) Divide the list of messages into two \((Q)\) subsets as balanced as possible, in the sense of the sum of elementary probabilities messages.
3) Assign respectively the symbol “0” and “1”, (up to \(Q-1\)) to the first and second (up to \(Q-1\)) subsets (root divided into \(Q = 2\) branches)
4) Repeat the steps 2) 3) with each subset (nodes divided into 2 (or \(Q\)) new branches) until that the operation becomes impossible (then each message has become a corresponding code-word leaf on the tree).

We note that since there may be an ambiguity in the choice of the subsets in each dichotomy, the Shannon-Fano code is not unique.

\[ \begin{align*}
\text{Example 1} & : \text{Source S with 5 messages with respective probabilities: } 0.4 ; 0.19 ; 0.16 ; 0.15 ; 0.1. \\
& \text{Case of a binary Shannon-Fano code delivers respective codewords: } 00, 01, 10, 110, 11 \\
& \text{with: } L=2.25 \text{ symb., } L_{\text{min}} = H(S)/1 = 2.15 \text{ symb. } \Rightarrow \text{efficiency } \eta = L_{\text{min}}/L = 95.6\%, \text{ redundancy } = 4.4\% \\
\end{align*} \]

\[ \begin{align*}
\text{Example 2} & : \text{Source S with 2 letters \{A,B\} with respective probabilities: } pA = 0.8 \text{ and } pB = 0.2. \\
1) & \text{Case of a direct (fixed length) coding (codewords 0 and 1): } L=1 \text{ symb. (letter), } \eta = H(S)/(L.1) = 72.2\%, \\
2) & \text{Case of a binary Shannon-Fano coding of } S^2, \text{ second-order extension of S: messages } \{AA, AB, BA, BB\} \text{ with probabilities } \{0.64 ; 0.16 ; 0.16 ; 0.04\} \text{ have respectively the corresponding codewords: } 0, 10, 110, 111. \\
& \text{with: } L_2=1.56 \text{ symb. (per word of 2 letters) } \Rightarrow \text{efficiency } \eta = H(S^2)/(L_2.1) = 2H(S)L_2 = 92.56\%, \\
3) & \text{Case of a binary Shannon-Fano coding of } S^3, \text{ order 3- extension of S: messages } \{AAA, AAB, ABA, ABB, BAB, BBA, BBB\} \text{ with probabilities } \{0.512 ; 0.128 ; 0.128 ; 0.128 ; 0.032 ; 0.032 ; 0.032 ; 0.008\} \text{ have respectively the corresponding codewords: } 0, 100, 101, 110, 11100, 11101, 11110, 11111. \\
& \text{with: } L_3=2.184 \text{ symb. (per word of 3 letters) } \Rightarrow \text{efficiency } \eta = H(S^3)/(L_3.1) = 3H(S)L_3 = 99.17\%, \\
\end{align*} \]

II.4.b) Huffman codes
The Huffman code is optimum in the sense that no other instantaneous code for the same alphabet (and same probabilities distribution) can have a better efficiency (i.e. a lower expected length). In particular, the code efficiency obtained by Huffman algorithm is greater than or equal to the code efficiency obtained by Shannon-Fano algorithm. But this efficiency is \(\leq 1\) (optimum code is not necessarily absolutely optimal code !)

It can be proved that an instantaneous optimum code must satisfies the following properties:
- the shorter codewords are assigned to source messages with higher probabilities,
- lengths of the two (Q) longest (i.e. less likely) codewords are equal
- the two (Q) longest codewords differ only in the last symbol (and correspond to the two least likely source messages).

Algorithm is presented in the particular case of a binary code (Q = 2) but is easily generalized for Q > 2.
It is based on the following procedure, which can be represented by using a tree:

1) Arrange the “outputs” (source messages in the initial iteration) in decreasing order of their probabilities.
2) Combine the two (Q) least probable messages together into a single new “output” (node of the tree) that replaces the two (Q) previous ones, and whose probability is the sum of the corresponding probabilities.
3) If the number of remaining “outputs” is 1 (the remaining node is the root of the tree), then go to the next step; otherwise go to step 1 (and increment the number of iterations), with a new list to be arranged, with a number of “outputs” reduced.

For encoding (assignment of the Q-ary symbols to the different nodes, we have to proceed backward i.e. from the root of the tree (last iteration node) to the different terminal nodes (including first iteration nodes):

4) Assign arbitrary “0” and “1” (up to “Q-1”) as first symbol of the 2 (Q) words (nodes) corresponding to the 2 (Q) remaining outputs (last iteration node with sum = 1).
5) If an output is the result of the merger of 2 (Q) outputs in a preceding iteration, append the current word (node) with a “0” and “1” (up to “Q-1”) to obtain the word (node) for the preceding outputs and repeat 5). If no output is preceded by another output in a preceding iteration, then stop (first iteration).

Example 1: Source S with 5 messages with respective probabilities: 0.4; 0.19; 0.16; 0.15; 0.1.
Case of a binary Huffman source encoding: a set of codewords: 1, 000, 001, 010, 011
with: L = 2, 2 symb., Lmin = H(S) / 1 = 2, 15 symb. => efficiency η = Lmin/L = 97.7%, redundancy = 2.3 %

Example 2: Source S with 2 letters {A, B} with respective probabilities: pA = 0.8 and pB = 0.2.
- case of a binary Huffman encoding of S², order 2– extension of S, delivering messages {AA, AB, BA, BB} with probabilities {0.64; 0.16; 0.16; 0.04} for codewords: 0, 11, 100, 101.
  We get L = 1, 56 symb. and an efficiency η = 92.56 %: idem code Shannon-Fano
- case of a binary Huffman encoding of S³, order 3- extension of S, delivering messages {AAA, AAB, ABA, BAA, ABB, BAB, BBA, BBB} with probabilities {0.512; 0.128; 0.128; 0.128; 0.032; 0.032; 0.032; 0.008} for a possible set of respective codewords: 0, 100, 101, 110., 11100, 11101, 11110, 11111.
  We gets L = 2, 184 symbols and an efficiency η = 99.17 %, same as for Shannon-Fano code in this example.

Comments on the (lossless) coding source:
- CAUTION, no reduction of information: data compression but not information compression!
- The more redundant is the source, the more useful could be the source encoding.
- Need to have knowledge of a statistical description of the source. In practice, estimation of the messages frequency (which may be adaptive) from the observation
- Source encoding makes the messages more vulnerable to errors in transmission, which cause the multiplication of errors after source decoding. “Channel encoder” will reintroduce redundancy, which obviously will reduce the bit-rate reduction gained by the source encoder.
**File of Information Theory n°1 : Source coding, Information rate**

**Exercise 1: application.** A "Quality system" in output of a production line provides for each product (with a regular rate) a test result among 3 states: {B (OK), D (defective), V (to be checked / tuned)}, with the respective probabilities: 60 %, 10 %, 30 %.

180000 products are tested during one hour. The result of this control has to be sent in real-time via a binary modem with a maximum bit-rate 96 bit/sec, assumed without error.

1°) What is the Baud Rate of messages of the source S? Infer whether it is possible to make a (lossless) transmission using a direct (i.e. fixed length) binary encoding: {B, H, V} -> {00, 11, 01}?

2°) Compute the entropy (Sh/letter), redundancy and information rate (Sh/sec) of the discrete source S assumed without memory?

3°) With an ideal binary source encoding process, what is the minimum “average length of the code” theoretically possible to achieve, and the minimum possible bit-rate after coding $D_{b_{min}}$? Is then the (lossless) transmission possible through the modem?

4°) Huffman coding: to possibly reduce throughput, we use a source Huffman encoder. What is the expected length of the code, the code efficiency, and the resulting bit-rate?

5°) Order 2 extension: to further reduce the bit-rate, individual letters are not encoded separately, but Huffman coding is applied to groups of 2 letters. Check that 1 possible result of encoding is as follows: BB => 1, BV => 000, VB => 001, VV => 0100, DB => 0110, DB => 0111, DV => 0011, VD => 010100, DD => 010101 What is then the bit rate after coding? Conclude. Appendix question A1) Answer to 4) and 5) with a Shannon-Fano coding?

6°) Source with memory S*: it is now assumed that there is a dependency between the State of the product tested at instant $n$ and the State of the product tested at previous instant $n-1$, according to:

<table>
<thead>
<tr>
<th>$PR(S_{n+1} / S_n)$</th>
<th>(B)</th>
<th>(V)</th>
<th>(D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(B)</td>
<td>0.7</td>
<td>0.3</td>
<td>0</td>
</tr>
<tr>
<td>(V)</td>
<td>0.6</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>(D)</td>
<td>0</td>
<td>0.3</td>
<td>0.7</td>
</tr>
</tbody>
</table>

- compute the (stationary) probabilities of the 3 states and check their agreement with the general statement,
- compute the new entropy, as well as redundancy, and information rate of S. Conclusion.

**Exercise 2: Source coding with imposed code-words length.**

Let a memoryless source with size alphabet $N = 6$. We want to apply an instantaneous code to this source.

2.1) Give the minimum size of the code alphabet $Q$ allowing to have:
   a) 1 codeword of length 1, and 5 codewords of length 2.
   b) 2 codewords of length 2, and 4 codewords of length 3.

   Give example of such a code with its tree for each case.

2.2) For the two previous cases, specify the efficiency of the code if it is assumed that the probabilities of the $N$ messages are {1/12; 1/8; 1/8; 1/8; 1/6; 3/8}, in choosing the best mapping for the codeword lengths.
III. Channel capacity and coding

III.1 "Channels" of the Information Theory

III.1.a) Introduction:
As seen in introduction chapter (figure 1), the Information Theory channel may include different blocks (or parts of block) of the communication chain. If we restrain our study to digital (binary or Q-ary) messages, the channel entry is discrete, but its output may be discrete or continuous.

Discrete output: a "hard" decision is taken into the demodulator output. For example, for binary output: bits 0 or 1 are decided by the use of a threshold applied to an analog signal in the demodulator output, that provides to the decoder a sequence of binary elements. The information loss is irreversible, but allow simple decoding algorithms, working with binary elements.

Continuous output: one "soft" decision is issued out of demodulator that provides the decoder a sequence of "analog" output samples. The performance of the decoder to "soft" entries is best, but with an increase in the complexity of algorithms for decoding.

The two major channels are discrete channel model, and Additive White Gaussian Noise (AWGN) continuous channel model. If the channel is memoryless, the output "symbol" at a given time depends statistically only on the corresponding input symbol (no inter-symbol-interference).

In the present chapter, we consider only the Discrete Memoryless (stationary) Channel (DMS). The AWGN continuous channel will be only evocated in chapter IV.

III.1.b) characterization of a Discrete Memoryless Channel: probability transition matrix

![Figure 5: Discrete Channel](image)

We note:  
- **X**: channel input, from an input finite alphabet \( A_x = \{x_1, \ldots, x_N\} \), with \( N \)-ary symbols
- **Y**: channel output, from an output finite alphabet \( A_y = \{y_1, \ldots, y_M\} \), with \( M \)-ary symbols

- if \( M = N \): in most often cases (= 2 for binary symbols),
- if \( M > N \): creation of intermediate levels,
- if \( M < N \): merger of levels (indistinguishable levels).

The channel is characterized by the link between the entry and the output, with a probabilistic model. Only matrix containing the conditional \( \text{à priori} \) probabilities \( p(y_j / x_i) \) for \( i = 1 \ldots N, j = 1 \ldots M \) (or \( \text{à posteriori} \) probabilities \( p(x_i / y_j) \)) characterize the channel, independently of the source (probabilities \( p(x_i) \)), unlike entropies \( H(X,Y), H(Y/X), H(X/Y), \ldots \).
Transition Matrix \([P (Y/X)]\), size \(N \times M\):

\[
[P(Y/X)] = \begin{bmatrix}
    p(y_1/x_1) & \cdots & p(y_M/x_1) \\
    \vdots & \ddots & \vdots \\
    p(y_1/x_N) & \cdots & p(y_M/x_N)
\end{bmatrix}
\]

The transmission channel is therefore a random type operator from the \(X\) space into the \(Y\) space.

Remarks:
- the sum of elements in a row is equal to 1:
  \[
  \sum_{j=1}^{M} p(y_j/x_i) = 1, \quad \forall \ i = 1...N, \text{ but it is not the case for the columns.}
  \]
- \(p(y_j) = \sum_{i=1}^{N} p(y_j/x_i) \cdot p(x_i)\), and in a (row) vector form \([P(Y)] = [P(X)]^T \cdot [P(Y/X)]\).
- special case: the matrix of a noiseless channel is square and equal to the identity matrix.

### III.1.c) Specific types of discrete memoryless channels: "uniform" channels

- **Uniform Channel for the Inputs**: the symbol issued at the input can be transformed into \(M\) symbols in output, with the same set of probabilities, whatever the issued symbol \(x_i\): channel disrupts the same way the different symbols input. (\(\Leftrightarrow\) each row of \([P(Y/X)]\) is a permutation of each other row, whatever the index of the row, \(i\)).

Thus, \(H(Y/X)\) is independent of \(p(x_i)\) and can be summarized in this special case to:

\[
H(Y/X) = H(Y/X = x_i) = -\sum_{j=1}^{M} p(y_j/x_i) \cdot lb(p(y_j/x_i)) \quad \forall i = 1...N
\]

**example**: \(N = 2, M = 3\):

\[
[P(Y/X)] = \begin{bmatrix}
    1-p-q & p \\
    q & 1-p-q
\end{bmatrix}
\]

- **Uniform Channel for the Outputs**: the channel transition matrix \([P(Y/X)]\) has the same set of probabilities on the different columns, whatever the index of the column \(j\). (\(\Leftrightarrow\) each column of \([P(Y/X)]\) is a permutation of each other column, whatever the index of the column, \(j\)).

**example**: \(N = 3, M = 2\):

\[
[P(Y/X)] = \begin{bmatrix}
    1 & 0 \\
    0 & 1 \\
    0.5 & 0.5
\end{bmatrix}
\]

For such a channel, an uniform input distribution \((p(x_i) = 1/N)\) leads to an uniform output distribution \((p(y_j) = 1/M)\):

if \(p(x_i) = 1/N, \forall i\), then \(p(y_j) = \sum_{i=1}^{N} p(x_i) \cdot p(y_j/x_i) = 1/N \cdot \sum_{i=1}^{N} p(y_j/x_i)\) is well independent of \(j\).

- **Symmetric channel**: doubly Uniform Channel (uniform for the inputs and for the outputs), with \(N = M\).

Example: binary symmetric channel (BSC), \(N = M = 2\), \(p = p(y_2/x_1) = p(y_1/x_2)\);

**Note**: \(p = Pe\): (binary) error probability \((Pe = p(x_2, y_2) + p(x_1, y_2))\):

\[
[P(Y/X)] = \begin{bmatrix}
    1-p & p \\
    p & 1-p
\end{bmatrix}
\]

This is the most widely used model for digital data transmissions.

"symmetric": errors are evenly spread over the 2 symbols. The conditional entropy \(H(Y/X)\) does not depend on symbols probabilities. \(H(Y/X)\) is equal to the entropy of an unbalanced binary source (with probability \(p\) for a given symbol, \(1-p\) for the other).

### III.2 a Capacity of discrete memoryless channel

#### III.2.a) Entropies, Mutual information, and definition of the channel capacity:

Let \(X\) and \(Y\) be 2 sources respectively in input and output of the channel.

Interpretations given the specific function of \(X\) and \(Y\):

- **H(X)**: amount of average information applied at the input of the channel. Also amount of average information of the source (after occurrence) = average uncertainty (before occurrence)
H(Y): average uncertainty at receiver (one part due to the information from X, one other part due to the noise/errors):

\[ H(Y) = I(X ; Y) + H(Y/X) \]

H(Y/X): "average channel error" = average uncertainty on Y knowing X, due to transmission errors.

I(X;Y): average information shared by X and Y, and then well conveyed.

We have:

\[ I(X ; Y) = H(X) - H(X/Y) \]

H(X/Y) : ambiguity (or equivocation) = average amount of information non conveyed, lost because of noise.

I(X ; Y) : measures the average amount of information really transmitted by the channel.

The capacity (per symbol) C of a discrete memoryless channel is defined as the largest average mutual information I(X;Y) that can be transmitted over the channel in one channel use (one symbol), maximized over all possible input probability assignments (assuming that X is a DMS source connected in input):

**The channel capacity per symbol (or channel use):**

\[ C = \max_{P(X)} \{ I(X ; Y) \} , \text{ in Sh/symb}. \]

The channel capacity is intrinsic to the channel, whereas entropies and mutual information depend on the symbols distributions of the source. C is the largest average amount of information that can be transmitted over the communication channel. The Shannon’s channel coding theorem will give the major significance of the concept of capacity (as the upper bound on the amount of information that actually can be reliably transmitted).

The capacity per unit of time (second) is obtained by multiplying per the channel baudrate \( Dc = D(X) \):

**The channel capacity per second:**

\[ C_t = C \cdot D(X) , \text{ in Sh/sec} \]

Remarks:
- the maximum value of I(X,Y) for a certain distribution of \( p(x_i) \) does exist (since the function is continuous, and the maximization is over a closed bounded region)
- the calculation of the capacity may be complex, and does not always take an analytical form,
- we check obviously that \( C \leq \log(N) \) Sh / symb and \( C_t \leq D_b(X) \) bit/sec.

**III.2.b) Capacity of special channels**

- **Noiseless channel:** deterministic lossless channel, with \( N = M \), and \( [P(Y/X)] = \text{id} \) (=> \( H(Y/X) = 0 \) and \( H(X/Y) = 0 \)). The set of probabilities at the input channel that maximizes I(X,Y) also maximizes H(X), with thus \( p(x_i) = 1/N, \forall i \). We get : \( C = \max \{ I(X ; Y) \} = \max \{ H(X) \} = \log(N) \).

Without noise, an equiprobable input distribution (obtained through ideal source encoding) ensures the more efficient transmission. But in presence of noise, it is generally not true.

- **Uniform Channel for the Inputs:** we have proved in this case that \( H(Y/X) = H_{\text{const}} \) was independent of \( p(x_i) \). Then:

\[ C = \max \{ H(Y) \} - H_{\text{const}} \text{ obtained in maximizing } H(Y): \quad C \leq \log(M) + \sum_{j=1}^{M} p(y_j / x_i) \log(p(y_j / x_i)) \]

Equality holds if it exists (it is not always the case) a set of input probabilities \( \{ p(x_i) \} \) such that the output probabilities, \( \{ p(y_j) \} \) be equi-probable.

- **Uniform Channel for the Outputs:** in the general case, no further conclusion à priori for the capacity computation.

- **Symmetric channel:** the channel being uniform for the inputs, the capacity is obtained by maximizing H(Y). The channel being uniform for the outputs, we have shown (Cf III.1.c) that the absolute maximum for H(Y) (i.e. \( \log(M) \)) exists since the set of input probabilities \( p(x_i) \) that makes the outputs Y uniformly distributed exists: it corresponds to the choice of an uniform input distribution \( p(x_i) = 1/N \). We then get:

\[ C = \log(M) + \sum_{j=1}^{M} p(y_j / x_i) \log(p(y_j / x_i)) \]

Therefore, for a symmetric channel, the maximization of the mutual information I(X,Y) is achieved by using the inputs with equal probability.
**example of the Symmetric Binary Channel (SBC)**: we denote \( P_e = p(0/1) = p(1/0) \Rightarrow 1 - P_e = p(0/0) = p(1/1); \)
\((p(0/1)\) is here directly the binary probability of error\)

\[
C = \text{Max} \{ H(Y) - H(Y/X) \} = \text{Max} \{ H(Y) - H(Y/X) \} \text{ since channel is Uniform with respect to the inputs.}
\]

With: \( \text{max} \{ H(Y) \} = \text{lb}(2) = 1 \); corresponding to uniform distribution for \( \{p(y_j)\} \), obtained from uniform distribution for \( \{p(x_j)\} \):

\[
H(Y/X) = H(Y/X = x_i) = P_e \cdot \text{lb}(P_e) + (1 - P_e) \cdot \text{lb}(1 - P_e) \Rightarrow \text{corresponds to } H_2(P_e).
\]

Then:

\[
C = 1 + P_e \cdot \text{lb}(P_e) + (1 - P_e) \cdot \text{lb}(1 - P_e) = 1 - H_2(P_e)
\]

example: \( C = 0.92 \text{ sh/digit for } P_e = 10^{-2}. \)

![Figure 6: Capacity of the Symmetric Binary Channel versus the binary probability of error](image)

**Remarks:**
- for a SCB "without disturbance" \((P_e = 0): C \) is absolutely maximum, and \( H(X/Y) = 0, H(Y) = H(X) \)
  \( \Rightarrow C = \text{max} \{ H(X) \} = \text{lb}(2) \) Sh/digit, obtained for a set of equiprobable \( x_i \).
- for a SBC "completely disturbed" \((P_e = 0.5): C = 0 \text{ sh/digit} \Rightarrow \text{no information is transmitted!} \)

### III.3. Channel coding and fundamental theorem

**III.3.a) introduction**

- **A noiseless channel**, is characterized only by its baudrate, \( Dc = D(B) \) Symb/sec, and by the size of its alphabet, \( Q. \)
  
  The capacity of such a channel is given by \( \text{lb}(Q) \) Sh/Symb and corresponds to the maximum average amount of information that can be error-free transmitted, obtained thanks to (ideal) source coding that makes equiprobable input symbols.

Without loss of generality, we assume a **standardized source** \( U \) (real source + source coding) **without redundancy**, where \( H(U) = \text{lb}(Q) \) is entropy, with equi-probable and independent symbols, and \( Q \) is the size of the alphabet.

(And channel “input": \( B; \) output: \( B’; \) size of channel alphabet (both input/output): \( Q \))

**When the channel is noisy:**

- The maximum amount of information shared between channel output and input decreases \( (C < \text{lb}(Q)) \), and we also have to consider the quality of the transmission. However, to be usable, communication must be reliable, with a sufficiently low probability of error \( (< \varepsilon) \), or with a sufficiently low **ambiguity**: \( H(B/B’) < \varepsilon’ \), knowing that the amount of well transmitted average information \( (\text{CF III.2.a) is given by:} \)

\[
I(B ; B’) = H(B) - H(B/B’)
\]

- If the standardized source \( U \) was directly connected to the channel input \( (B = U) \), \( H(B) \) would be absolutely maximum \( (= \text{lb}(Q)) \) but \( H(B/B’) \) would then depend on the noisy channel \( (\geq \text{lb}(Q) - C) \), and may be too strong according to the imposed quality criteria \( (\varepsilon’) \) (or would correspond to a probability of error \( P_e > \varepsilon) \).

- To improve the reliability, the sequence of messages of the standardized source must be transformed by a so-called channel **coding process**, with the complementary channel decoding process in channel output.

This channel coding is necessary redundant to allow the decrease (in average) of \( H(B/B’) \). We have necessary to satisfy \( H(B) \leq C \) to get \( H(B/B’) = 0+ \) and \( I(B ; B’) \approx C- \).

Note that the channel coding will put back redundancy of a special sort, by introducing statistical dependence between the successive input symbols of the channel in \( B \) (but possibly with still equi-probable symbols).
Figure 7: Inserting a channel encoding to protect the binding

In summary:

Objective of the channel coding: protect the messages against disturbance of the channel, to make a noisy channel behave like a noiseless channel. It introduces redundancy by a coding process designed to make the noisy received signal decodable.

- When the channel encoder is applied, we have between the input and output of the encoder:
  - reduction of the standardized entropy: \( \frac{H(B)}{\log_2(Q)} < \frac{H(U)}{\log_2(Q)} = 1 \text{ sh/digit} \)
    Since redundancy \( R(B) > 0 \) \( \Rightarrow \) 1 digit is less informative after the channel encoding
  - conservation of the information rate (for real time transmission): \( H_t(B) = H_t(U) \text{ sh/sec} \)
    since the encoding is lossless \( \Rightarrow \) channel encoding does not add nor remove information.

  \( \Rightarrow \) increase in the (equivalent) Binary rate (for real time transmission): \( D_B(B) > D_B(U) \text{ bit/s} \)
  Indeed, the redundancy is positive: \( D_B(B) > H_t(B) = H_t(U) = D_B(U) \)

III.3.b) characterization of a channel coding

Channel encoding inserts redundant bits (or symbols) sequentially (convolutive codes) or per blocks (block codes).

We focus only on block-code in the following.

- Block - coding case, code \((n, k)\): We get a fixed- to fixed-length coding \((k > n)\)
  The symbols of the standardized source \(U\) are grouped into \(k\) symbols \((Q^k\) possible messages of the \(k\)-th order extension of the source, \(U^k\) \(): \( m_i = u_{i1} \ldots u_{ik} \), with \(1 < i_1, \ldots, i_k < Q\).
  Each message is mapped by the encoder into an unique code-word of \(n\) \(Q\)-ary symbols, with \(n > k\).
  Thus, there is an application from the \(Q^k\) messages (with length \(k\) symbols) to the \(Q^n\) word-codes (with fixed-length \(n\) symbols): 1 code = 1 set of \(Q^k\) code-words among \(Q^n\) possible.

Figure 8: A systematic \((n, m)\) block coding format, (with binary symbols: \(Q = 2\))

Note: The source \(B\) is with memory. It may issue equi-probable symbols with \(H(B) < H_{max} = \log_2(Q)\) because of dependency.

In order to achieve a real time data transmission, the baud-rate after channel coding must be increased with a \(n/k\) ratio:
\( D(B) = (n/k) D(U) \text{ symb/sec} \)

channel encoding rate:
\( \rho_{cc} = 1 - k/n = (n-k)/n \)

corresponds to the redundancy of the encoded source, \(R(B)\) (or redundancy of the code), if the standardized source \(U\) was well without redundancy.

Indeed: (average) entropy after channel encoding is \(H(B) = H(U).k/n\),
then the redundancy after coding \(R(B) = 1 - H(B)/\log_2(Q)\) with \(\log_2(Q) = H(U)\) here.
• Efficiencies (information theory): with respect to the channel capacity, we can define:
  - channel coding efficiency: \( \eta_K = \frac{H(B)}{C} \)
  - channel use efficiency: \( \eta_{CA} = \frac{I(B, B')}{C} \)

Note: these efficiencies do not reflect the reliability of the transmission.

• Decoding: after transmission in a noisy channel, the decoding requires a rule of decision.
  Each received noisy codewords \( r_j \) (\( j = 1 \ldots Q \) possible), is translated into a message \( m_i \):
  - Maximum A Postériori:
    \( \{ m_i \text{ such as: } \text{Prob}(m_i/r_j) \text{ maximum} \} \)
    The Maximum à Postériori criteria leads to the minimization of the probability of error \( P_{ed} \).
    It is equivalent to use a Maximum Likelihood criteria in case of uniform probabilities for the messages:
  - Maximum Likelihood:
    \( \{ m_i \text{ such as: } \text{Prob}(r_j/m_i) \text{ maximum} \} \)
  We will then measure a binary error probability \( P_{ed} \), by comparing bits before encoding and after decoding.

**Basic example of the “repetition code”:** to improve the reliability but at the expense of information rate (in reasoning at constant baudrate \( D_c \) in the channel):
repeat \( R = 2r + 1 \) time every bit of source \( U \): can be regarded as a block code \((R, 1)\) encoding.
At the receipt of a block of \( R \) bits, if more than \( r \) bits are equal to “1”, we decide “1”, otherwise we decide “0”.
• for SBC, \( R = 3 \) repetition, and \( P_e = 10^{-2} \):
  - enhancement of the reliability: \( P_{ed} = 3 \times 10^{-4} \) versus \( P_e = 10^{-2} \) without channel coding:
  - error probability \( P_{ed} = \text{pr. 2 or 3 errors / message} = 3P_e^2(1 - P_e) + P_e^3 = 3 \times 10^{-4} \),
  But Non-efficient code:
  - redundancy: \( R(B) = \rho_{cc} = 66.7 \% \), \( H(B) = 1/3 \) Sh/digit for \( H(U) = 1 \) Sh/digit
  - channel coding efficiency: \( \eta_K = (1/3)/0.92 = 36.2\% \).
    because (with a SBC): \( C = 1 + P_e \text{lb}(P_e) + (1 - P_e) \text{lb}(1 - P_e) = 0.92 \text{Sh/digit for } P_e = 10^{-2} \).
  - information rate \( H(t)(U) = H(t)(B) = H(B).D(B) \), i.e. a rate divided by 3 compared to the situation without coding in reasoning at constant baudrate \( D_c \) in the channel (else \( D_c \) had to be multiplied by 3, in reasoning at constant Information rate instead of constant \( D_c \)).
• conclusion: with \( R \) repetitions, the probability of error \( P_{ed} \) decreases, but the information rate \( H(t)(U) \) must be divided by \( R \). When \( R \to \infty \), \( P_{ed} \to 0 \) but \( H(t)(U) \to 0 \), therefore very limited interest!

\( \Rightarrow \) C. Shannon proves a quite remarkable result (unexpected in 1948) concerning the existence of efficient codes:

### III.3.c) channel coding theorem

**Shannon’s channel coding theorem** (or: « Noisy channel theorem », « fundamental theorem of information theory »)

If (and only if) the information rate of the source \( U \) is less or equal to the channel capacity (both expressed per unit of time), i.e.
\( H_t(U) \leq C_t \text{ Sh/sec, there exists a channel coding scheme allowing transmission over a memoryless channel as reliable as desired, that is with an average probability of symbol error after decoding arbitrary small:} \)

\[ P_{ed} < \varepsilon, \forall \varepsilon \text{ real } > 0, \]

In other words, a redundancy rate \( R(B) \) close \( 1 - C / \text{lb}(Q) \), is theoretically enough possible to have reliable error-free transmission.

Note: \( H_t(U) \leq C_t \text{ sh/sec and } H(t)(U) = H(t)(B) \Leftrightarrow H(B).D(B) \leq C.D(B) \text{ sh/symb \Leftrightarrow } H(B) \leq C \text{ Sh/symb} \).

**NSC of the second Shannon’s theorem:** the entropy of the encoded source (i.e. after channel coding) in input of the channel should be less or equal to the capacity expressed per channel use (Sh per symbol).

**example:** SBC with \( P_e = 10^{-2} \), \( Q = 2 \) and a given Binary rate in the channel \( D_b(B) = D(B) = D_c = 34 \text{ Mbps} \).
And we desire a minimum information rate \( H_t(U) \geq H_0 = 30 \text{ MSh/s} * ; \)
  (Either a minimum bit rate \( D_b(U) \)) of 30 Mbps, assuming the source \( U \) without redundancy)

We have:
C = 0.92 Sh/digit, \( C_t = 0.92 \times D(B) = 31.3 \text{ MSh/sec} \) < \( D_b(B) = 34 \text{ Mbps} \).

source \( U \) without redundancy => \( H(U) = 1 \) sh/digit, and \( H_t(U) = D_b(U) = D_b(B) \).

(If we haven’t added redundancy in the channel coding, it would be: \( H_t(U) = H(U).D(B) = 1 \times D_b(B) = Ct = 31.3 \text{ MSh/sec} \) => impossible to get transmission as reliable as desired (2\(^{nd}\) Shannon’s theorem not satisfied).

• respect of 2\(^{nd}\) Shannon’s theorem: => maximum limit for the coding rate \( k/n \) such that: \( H_t(U) = (k/n). D_b(B) < Ct \)
  then \( (k/n) < 92 \% \), corresponding to a code rate \( \rho_c > 8 \% \) => 3 codes / Ped < \( \varepsilon \).

• constraint “information rate \( H_t(U) \geq H_{\theta} = 30 \text{ MSh/s} \)” => lower limit for the \( k/n \) ratio such that: \( (k/n). D_b(B) > H_{\theta} \)
  then \( (k/n) \geq 88.2\% \), corresponding to a rate of encoding \( \rho_c \leq 10.8 \% \).

Final conclusion: to pass at least an information of \( H_{\theta} \) and satisfy the 2nd Shannon’s theorem, the \( (k/n) \) ratio must satisfy:

\[
H_{\theta} < (k/n). D_b(B) < Ct
\]

A.N.: \( 88.23\% \leq k/n \leq 92 \% \).

but it is only a necessary (but not sufficient) condition to get a reliable communication => find a good code!

Key of the proof of the 2nd Shannon’s theorem: (the proof is in the book of G. Battiail for example):

Bound: as soon as information rate \( H_t = H_t(U) \) is less than the channel capacity \( C_t \), it can be proved that there exist codes allowing to achieve a decoding error probability Ped such as:

\[
\text{Ped} \leq 2^{-n \cdot F(H_t)}
\]

where
- \( F(H_t) \) is a function called "reliability" function: non-negative, zero for \( H_t \geq C_t \), decreasing with \( H_t \) if \( H_t < C_t \).
- \( n \) is the size of blocks

![Figure 9: Allure of the “reliability” function versus the information rate (left), and of th Error Probability after decoding versus the block size n, for a constant k/n ratio (right)](image)

In other words,
• for a given code rate or a given redundancy (\( \rho_c = 1 - k/n \)), increasing the size of blocks \( n \) can reduce the Error Probability Ped, with Ped → 0 for \( n \rightarrow \infty \)!

• price to be paid:
  - delay in the transmission due to the coding / decoding process
  - complexity increase due to the coding-decoding process

Comments:
• it is a theorem of existence, which does not say how to build efficient codes, which fed research during 50 years!
  (The Shannon’s proof of the theorem is based on the use of a random encoding).
• channel encoding theorem gives meaningless to the concept of channel capacity: \( C_t \).
Exercise 1: Capacity of specific channels
Calculate the capacity of different (discrete memoryless) channels below, and give a (or the) set of input probability to maximize the Mutual Information:

1.1) Z-channel, with transition matrix

\[
P(Y/X) = \begin{bmatrix}
0.9 & 0.1 \\
0 & 1
\end{bmatrix}
\]

1.2) Channel such that:

\[
\begin{array}{ccc}
x_1 & 1 & y_1 \\
x_2 & 1 & y_1 \\
x_3 & 1 & y_2
\end{array}
\]

1.3) Channel such that

\[
P(Y/X) = \begin{bmatrix}
0.5 & 0.5 & 0 & 0 \\
0 & 0 & 0.5 & 0.5
\end{bmatrix}
\]

Exercise 2: theoretical limit of reliable transmission over a « Symmetric Binary Channel »
Binary data have to be transmitted over a noisy modem, with maximum tuning bit rate \(Dc_{max} = 96\) kbit/s, modeled by a SBC with error probability \(Pe = 3 \times 10^{-2}\) (\(\forall Dc < Dc_{max}\)). We want to improve the reliability by inserting a channel coding (a systematic block code) between the (assumed perfect) source coding (bit rate = 64.5 kbit/s in output, without redundancy) and the channel.

1°) Does theoretically exist a channel coding allowing real time transmission with arbitrary low binary error probability after decoding (< \(10^{-12}\), for example).

2°) What is then the possible range of redundancy (or code-rate) for this channel coding?

Appendix Question: A1) If bit rate was fixed to \(Dc_{max}\), what would be the maximum information rate of the source to guarantee the existence of channel encoding bringing the desired reliability?