Statistical coding

Classification of the statistical codes

Information data coding

- ***Objectives***
  - Transcription of information to facilitate coding

    code signal *(Transcoding)*
  - Information compression

    reducing information size
  - Protection against transmission errors

    against loss and decision errors
  - Keeping transmitted information secret

    encryption
- ***Definition of a code***

  application of S in $\mathcal{A} = \{ a_1, a_2, \ldots\ldots, a_q \}$

  message m_i S code-word M_i $\mathcal{M}$ finite sequences of $\mathcal{A}$

Information coding consists of transcribing messages from an information source in the form of a sequence of characters taken from a predefined alphabet. The objectives of coding fall into four main categories:

> ***transcribing*** information in a form that makes it easy to create a signal that can handle the information, or easy to handle the information automatically. To do this, different codes for representing the information are used depending on the envisaged application, with transcoding operations frequently being used;
>
> ***reducing*** the number of information symbols needed to represent the information (in terms of the total number of symbols used): this is a space-saving role;
>
> ***preventing*** quality loss (distortion, noise) caused by the transmission channel and which lead to errors when you reconstruct the information when it leaves the transmission channel (upon reception);
>
> ***protecting*** confidential information by making it unintelligible except for its intended recipient.

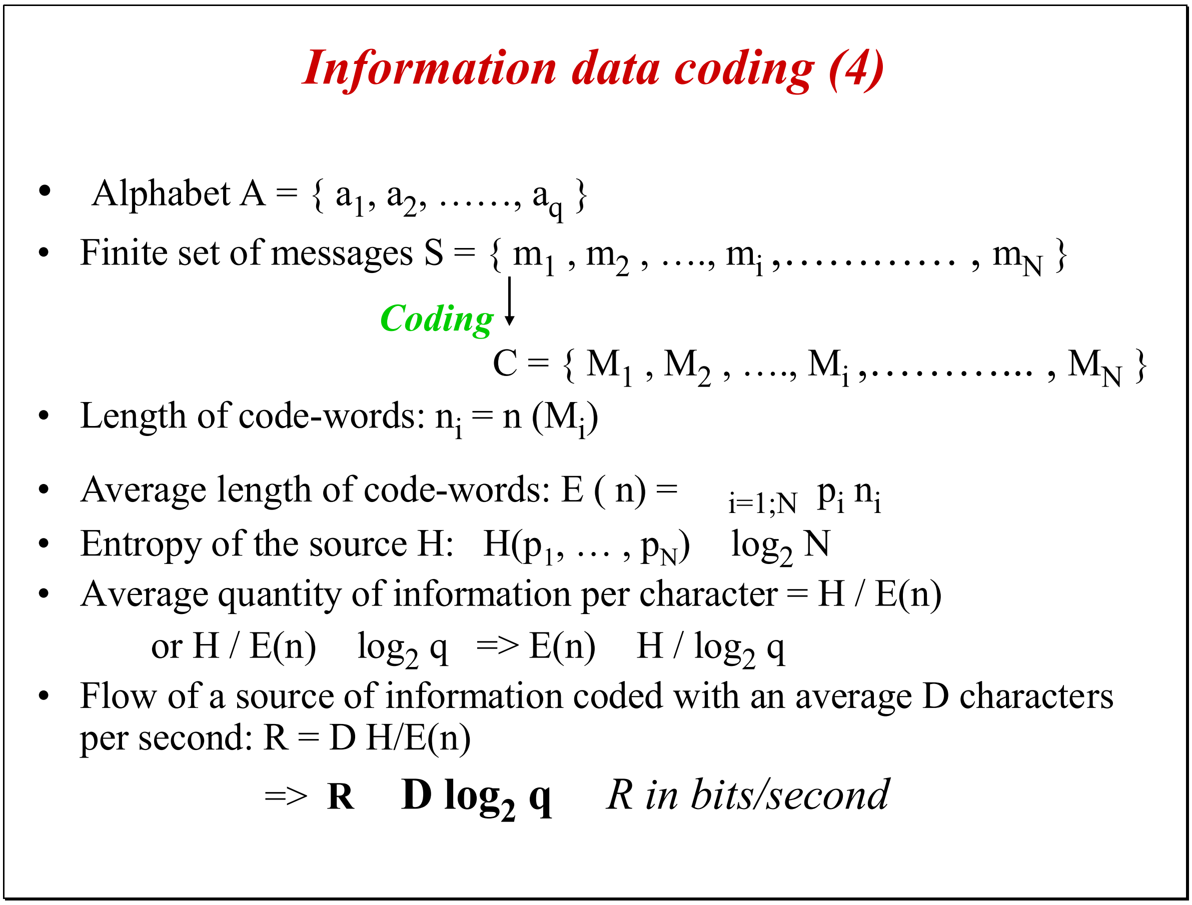
Definition of a code:

Given a set, called alphabet A made up of q characters a_i : $\mathcal{A} = \{ a_1, a_2, \ldots, a_q \}$and $\mathcal{M}$ the finite set of finite sequences M_i of characters (for example: $M_i = a_{10}\ a_4\ a_7$).
Given a finite set of messages emitted by a message source S: $S=\{ m_1, \ldots, m_N \}$.

A code refers to any application of S in $\mathcal{A}$: coding of S through the use of the alphabet A.

The element M_i of $\mathcal{M}$ which corresponds to the message m_i of S is called the ***codeword*** of m_i. Its ***length***, noted as n_i, is the number of characters belonging to $\mathcal{A}$ which compose M_i.

The ***decoding*** of a sequence of sent messages m_i involves being able to separate the codewords in a received sequence of codewords M_i. This is why we sometimes use a special spacing character in an alphabet.



Information data coding (4)

- Alphabet A = { a_1, a_2, ……, a_q }
- Finite set of messages S = { m_1 , m_2 , …., m_i ,………… , m_N }

Coding ↓

C = { M_1 , M_2 , …., M_i ,………… , M_N }

- Length of code-words: $n_i = n\,(M_i)$
- Average length of code-words: E (n) = $_{i=1;N}\ p_i\, n_i$
- Entropy of the source H: H(p_1, … , p_N) $\log_2 N$
- Average quantity of information per character = H / E(n)

or H / E(n) $\log_2 q$ => E(n) H / $\log_2 q$

- Flow of a source of information coded with an average D characters per second: R = D H/E(n)

=> **R** **D log$_2$ q** *R in bits/second*

From here on, we shall call the **messages** produced by the information source **m$_i$** and **M$_i$** the **codewords** associated with them.

We will call **n$_i$** = n(M$_i$) the number of characters belonging to an **alphabet** $\mathcal{A}$ (Card($\mathcal{A}$) = q) necessary for coding m$_i$, n$_i$ being the **length** of the codeword M$_i$. If the source uses N possible different messages, the average length of the codewords is given by:

$$E(n) = \sum_{i=1}^{8} p_i n_i \quad \text{, where } p_i = \Pr\{\, m_i \,\}.$$

H is the average uncertainty (i.e. the entropy) of the source S **per message** sent, so the average uncertainty (i.e. the entropy) **per character** (of the alphabet $\mathcal{A}$) equals $\dfrac{H}{E(n)}$ and we have: $\dfrac{H}{E(n)} \leq \log_2 q$ (because we have q characters in the alphabet $\mathcal{A}$), so: E(n) $\dfrac{H}{\log_2 q}$.

Finally, if the coded information source produces D characters per second taken from the alphabet $\mathcal{A}$, $\dfrac{H}{E(n)}$ being the average information transported per character in bit/character, the character rate R of information is: $R = D \cdot \dfrac{H}{E(n)}$.

This character rate is then limited by: R ≤ D.log$_2$q.

Coding and decoding information (5)

- ***Efficiency*** of a code: $= n_{min} / E(n)$ => $= H / (E(n) \log_2 q)$
- ***Redundancy*** of a code : = 1 -
- *Simple examples: codes C_1 and C_2*
- **Constraints**: separation of code-words & unambiguous reading of code-words => *regular and inverting codes*
- ***Regular code***: if m_i m_j ==> M_i M_j *(injective application)*
- ***Inverting codes*** : 2 sequences of distinct messages

  ==> 2 sequences of distinct codes

  if $(m_1, \ldots, m_i)$ $(m_1, \ldots, m_j)$ => $(M_1, \ldots, M_i)$ $(M_1, \ldots, M_j)$

  examples: fixed length codes; codes with separator

- ***Irreducible code***: inverting code that can be decoded without any device M_i is not a prefix of M_j i , j

Some definitions and properties linked to information encoding and decoding:

<u>Efficiency</u>:

For a given alphabet A, the efficiency of a code is η given by:

$$\eta = \frac{n_{min}}{E(n)} = \frac{min\ E(n)}{E(n)} = \frac{\frac{H}{\log_2 q}}{E(n)} = \frac{H}{E(n)\log_2 q} \quad , \quad \eta \in [0,1]$$

<u>Redundancy</u>:

The mathematical redundancy is defined by the factor ρ = 1 - η. Redundancy can be used to increase the robustness of the coding when faced with transmission errors for the coded information (error detection and correction).

Here is a simple example: we consider a source of 4 possible messages $\{m_1, m_2, m_3, m_4\}$ of probabilities: $p_1 = 0.5$; $p_2 = 0.25$; $p_3 = p_4 = 0.125$, respectively.
Given the following two codes C_1 (simple binary codage) and C_2 (variable length code):

Messages Codes	m_1	m_2	m_3	m_4
C_1	0 0	0 1	1 0	1 1
C_2	0	1 0	1 1 0	1 1 1

For C_1 : $\eta = 1.75/2 = 0.875$ and $\rho = 1 - \eta = 0.125$.
For C_2 : $\eta = 1.75/1.75 = 1$ and $\rho = 1 - \eta = 0$.
The code C_2 is of maximum efficiency (unitary) while code C_1 is not.

Regular code:

Any given code-word is associated with only one possible message (application S→A is bijective): if $m_i \neq m_j$ then $M_i \neq M_j$.

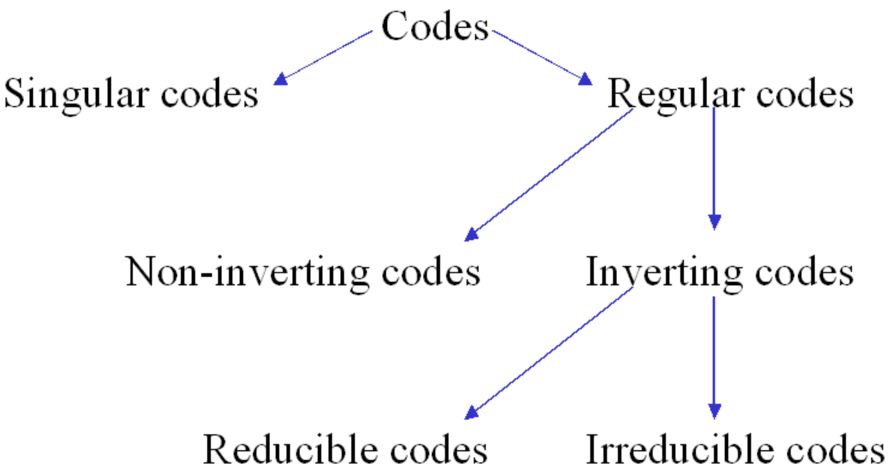
Inverting code:

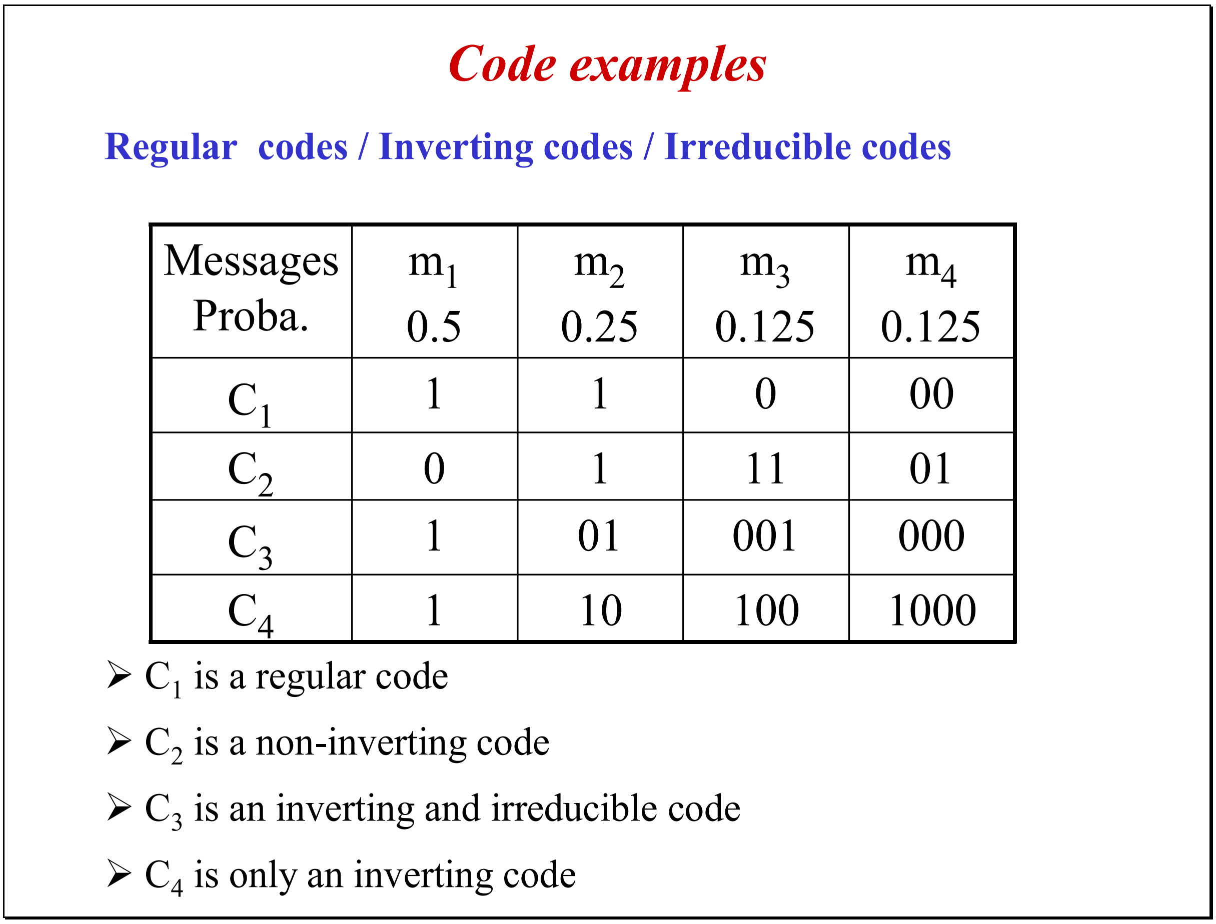
The code is inverting if two distinct sets of messages $(m_1, \ldots, m_i)$ and $(m_1, \ldots, m_j)$ necessary lead to distinct codings (for example code of fixed length such as C_1 and codes with separator). An inverting code is then a special case of a regular code.

Irreducible code:

This is a decryptable code that can be read directly without any special device (fixed length code, separator). To do that, any code-word M_i of a message m_i must have no prefix that is another code-word M_j.

In this way, we can create a hierarchical classification to characterize a code's type:





Code examples

Regular codes / Inverting codes / Irreducible codes

Messages Proba.	m_1 0.5	m_2 0.25	m_3 0.125	m_4 0.125
C_1	1	1	0	00
C_2	0	1	11	01
C_3	1	01	001	000
C_4	1	10	100	1000

- ➢ C_1 is a regular code
- ➢ C_2 is a non-inverting code
- ➢ C_3 is an inverting and irreducible code
- ➢ C_4 is only an inverting code

Here are four codes C_1, C_2, C_3 and C_4 given as examples of the previous definitions and properties. We suppose that the four messages m_1, m_2, m_3, and m_4 are distinct.

The code C_1 is not regular: $m_1 \neq m_2$ but $C_1(m_1) = C_1(m_2)$, and also $C_1(m_3) = C_1(m_4)$.

The code C_2 is a non-inverting code: the two texts $\{m_1, m_2\}$ and $\{m_4\}$ are different, but they lead to the same code « 01 ».

The code C_3 is an inverting and irreducible code: two distinct texts made up of sequences of messages, for example $\{m_1, m_3, m_4\}$ and $\{m_1, m_2\}$ always lead to different codes and no code-word $M_i = C_3(m_i)$ is prefixed by another code-word $M_j = C_3(m_j)$

The code C_4 is an inverting code but not irreducible: two distinct texts always lead to different codes but the code-words $M_i = C_4(m_i)$ are the prefixes of all the code-words $M_j = C_4(m_j)$ once $i < j$.