CHAPITRE II : (Partie 1)

Introduction à la théorie de l'Information.

1. Théorie de la compression ; entropie.
2. Méthodes statistiques (Codage de Huffman).
3. Méthodes par dictionnaire (type Lempel - Ziv).

I)-Théorie de la compression ; entropie.

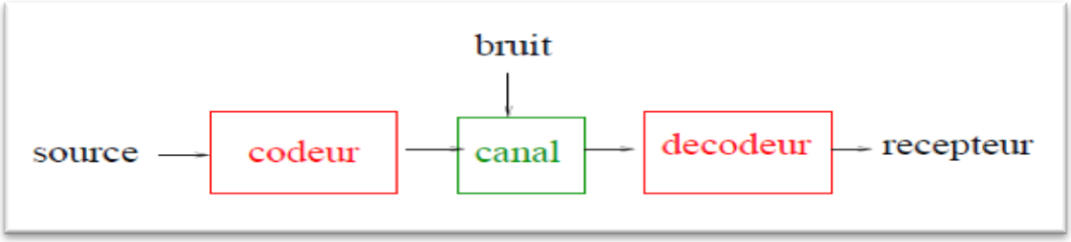
1. Introduction:

La théorie des communications s'intéresse aux moyens de transmettre une information depuis ***la source*** jusqu'à un destinataire à travers ***un canal***.

La nature de ***la source*** peut être très variée. Il peut s'agir par exemple ***d'une voix***, ***d'un signal électromagnétique*** ou d'une séquence de ***symboles binaires.***

Le canal peut être une ligne téléphonique, une liaison radio, un support optique.

La transmission peut se faire dans l'espace ou dans le temps.



- Le codeur représente l'ensemble des opérations effectuées sur la sortie de la source avant la transmission.
- Ces opérations peuvent être par exemple:
  - ***La modulation,***
  - ***La compression,***
  - ***Le brouillage,***
  - ***L'ajout de redondance pour combattre les effets du bruit, ou encore l'adaptation à des contraintes de spectre.***
- Elles ont pour but de rendre la sortie de la source compatible avec le canal. Enfin le décodeur doit être capable, à partir de la sortie du canal, de restituer ***de façon acceptable l'information fournie par la source.***

* Le but du ***codeur de source*** est de représenter la sortie de la source en ***une séquence binaire***, et cela de façon ***la plus économique possible.***

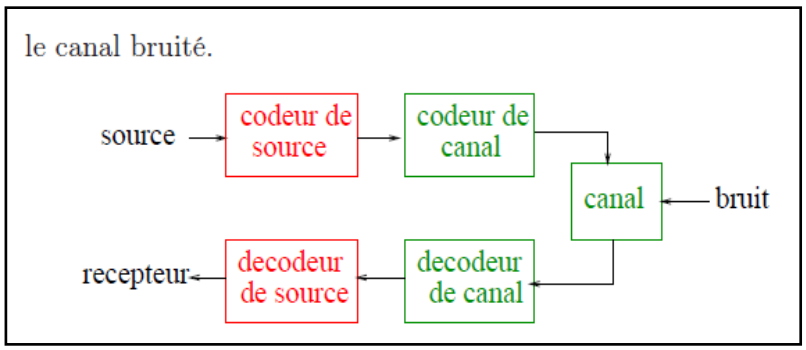
* Le but du ***codeur de canal*** et de ***son décodeur*** est de reproduire le plus fidèlement possible cette séquence binaire.

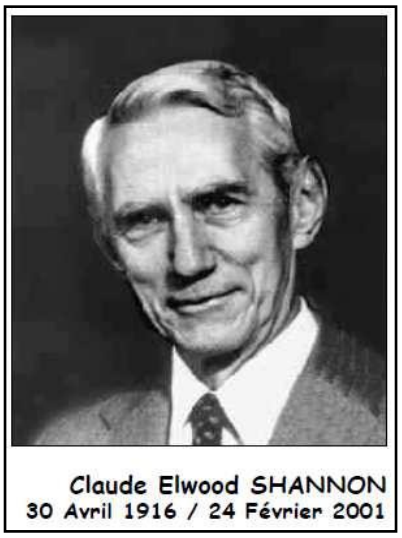
2- Définition:

La théorie de l'information a été créée par ***C. E. Shannon*** dans les ***années 1940***. Il s'agit ***d'une théorie mathématique*** qui décrit les aspects les plus fondamentaux des systèmes de communication.

Elle consiste:

- ✓ L'élaboration et l'étude de modèles pour la source et le canal qui utilisent différents outils comme les probabilités.




Claude Elwood SHANNON
30 Avril 1916 / 24 Février 2001

NB: Cette séparation entre codage de source et codage de canal n'implique en pratique aucune limitation sur les performances du système complet.

Sources et codage de sources:

1)* Source =
- ✓ Siège d'évènement aléatoires qui constitue le message émis.
- ✓ Caractérisation = Entropie.

2)* Codage de source
- ✓ But: Supprimer la redondance pour réduire le coût de transmission.
- ✓ 2 Théorèmes fondamentaux de Shannon.

3)* Types de source:

- ✓ Source discrète sans mémoire = la probabilité d'apparition d'un symbole ne dépend pas des symboles précédents.
- ✓ Source discrètes avec mémoire = modélisation par processus stochastique: chaine de Markov.
- ✓ Source discrètes stationnaires = probabilité d'apparition des différents symboles indépendants du temps.

II- Description d'un système de codage à longueur variable.

⦿ *On se donne une source discrète (données, fichier, . . .) dont chaque symbole x prend une parmi **M** valeurs possibles* $\{x_1, x_2, \ldots, x_M\}$.

⦿ *Une distribution de probabilité **p(x)** caractérise les statistiques de cette source, on la suppose connue (ou estimée) sous la **forme*** $\{p_1, p_2, \ldots, p_M\}$, *où pi est la probabilité d'occurrence du symbole xi .*

⦿ *Le codeur code chaque symbole de source* x_i *par un **mot de code*** c_2 ***. Le code** est l'ensemble des mots de codes* $\{c_1, \ldots, c_M\}$.

*Chaque mot de code **Ci** est composé par des symboles de code **{0,1}***

⦿ *Un **code à longueur variable (VLC : Variable-Length Code) est tel que les** différents mots de code n'ont pas nécessairement la même longueur, en bits.*

⦿ *On note* l_i *la longueur en bits du mot de code ci . La distribution des longueurs du code est donc* $\{l_1, l_2, \ldots, l_M\}$.

⦿ *La **valence du codage (V) est le** nombre de symboles de code.*

⦿ *La **taille** d'un codage est le nombre de mots de code.*

Information propre

⦿ Soit Un événement **X** ayant une probabilité **P(X)**. L'information propose associée à **X** est donnée par

$$i(X) = log_2 \frac{1}{P(X)} = -log_2 P(X)$$

:

⦿ Si la probabilité de l'événement A est faible (i.e. P(X)=0.1):

$$i(X) = -log_2 (0.1) = 3.3219$$

⦿ Intuitivement, quant on apprend la réalisation d'un événement peu probable, on reçoit beaucoup d'information et vice versa.

$$log_2 (0.1) = \frac{\ln (0.1)}{\ln (2)} = \frac{\log_{10} (0.1)}{\log_{10} (2)}$$

Entropie

⦿ Soit $X_1, \ldots, X_n$ le résultat d'une expérimentation donnée de probabilité $P(X_1), \ldots, P(X_n)$. L'entropie associé a cette expérience est donnée par:

$$H(X) = \sum_{i=1}^{n} P(X_i) i(A_i) = -\sum_{i=1}^{n} P(X_i) log_2(X_i)$$

⦿ L’entropie c’est le nombre minimal moyen de symboles binaires (b=2) nécessaire pour représenter la source.

⦿ L’entropie c’est alors la limite théorique de compression sans perte qu’on peut atteindre: Shannon.

⦿ L'entropie indique alors la quantité d'information nécessaire pour que le récepteur puisse déterminer sans ambiguïté ce que la source a transmis.

⦿ Plus le récepteur reçoit d'information sur le message transmis, plus l'entropie (incertitude) vis-à-vis de ce message décroît.

⦿ En particulier, plus la source est redondante, moins elle contient d'information.

⦿ En l'absence de contraintes particulières, l'entropie est maximale pour une source dont tous les symboles sont équiprobables.

⦿ **• Mesure la quantité d’information :**

⦿ Un signal peu informatif est ***redondant.***

⦿ Un signal informatif est ***très diversifié et peu prédictible.***

Exemple de calcul d’entropie

⦿ Considérons la séquence suivante:

1 2 3 2 3 4 5 4 5 6 7 8 9 8 9 10

⦿ Si l’on considère chaque symbole à part on:

$$P(1) = P(6) = P(7) = P(10) = \frac{1}{16}$$

$$P(2) = P(3) = P(4) = P(5) = P(8) = P(9) = \frac{2}{16}$$

- L’entropie est de 3.25 bits/symbole.

$$H = -\sum_{i=1}^{n} P\ i\ \log_2 P\ i\ = -4\frac{1}{16}\log_2\frac{1}{16} - 6\frac{2}{16}\log_2\frac{2}{16} = 3.25 \text{ bits}$$

- Nombre total des bits: 3.25*16=52 bits

⦿ La base du logarithme fixe l’unité de l’entropie : dans le cas de la **base 2 (V=2)**, l’entropie s’exprime en *shannon (sh). On peut démontrer que l’entropie est maximum dans le cas de l’équiprobabilité des* symboles.

⦿ L’efficacité d’un code est estimée par ses caractéristiques :

- **- Longueur moyenne :** $l_{moy} = \sum_i l_i \; p_i$
- **- Longueur minimum:** $l_{min} = \frac{H(X)}{log_2\,(V)}$
- **- Rendement:** $R = \frac{l_{min}}{l_{moy}} = \frac{H(S)}{l_{moy} log_2(V)}$
- **- Redondance :** $\rho = 1 - R$

Exemple: Supposons avoir *8 messages* $m_1, ...,m_8$ dont la probabilité d'occurrence est donnée par $p1 = p2 = ... = p8 = 1/8$.

- Il vient $\boxed{H(X) = -\sum_1^8 p_i \log p_i = 8.\frac{1}{8}.3 = 3}$
- Si l'alphabet est équivalent à {0,1}, alors $d=2$ et $\boxed{\bar{L} \geqslant \frac{3}{1} = 3}$
- *Donc, le codage à 3 bits [000,001,...,111] est efficace, ou non-redondant.*

Fondamental : Inégalité de Kraft

⦿ Fondamental : Inégalité de Kraft Soient l_1 , l_2 , ... , l_N *des longueurs de mots candidates pour coder une source N-aire* dans un alphabet binaire. Alors l'inégalité de Kraft :

$$\boxed{\sum_{i=1}^{M} 2^{-l_i} \leqslant 1}$$

⦿ Est une condition nécessaire et suffisante d'existence de codes déchiffrables et instantanés respectant ces longueurs de mots.

Test de décodabilité unique

⦿ Quelques définitions: Soit a= 010

b= 01011 => a est un préfixe de b et le *dangling suffix est 11*

- Algorithme de test de décodabilité unique

○ 1.Construction d'une liste de tout les mot-codes

○ 2.Détecter un mot code est un préfixe d'un autre mot code

○ 3.Rajouter le *dangling suffix à la liste [à moins qu'il ne soit précédemment ajouté]*

- 4.Itérer les étapes 2 et 3 en utilisant la nouvelle liste contenant le nouveau *dangling suffix Conditions d'arrêt:*

- 1.On tombe sur un *dangling suffix qui est un mot-code. Non*

- 2.Il n y a plus de *dangling suffixes*

Code préfixe

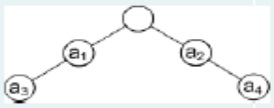
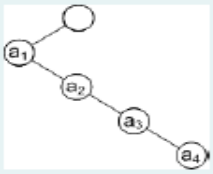
⦿ Pour s'assurer que le test de décodabilité unique soit positif, on peut considérer un code où aucun mot-code n'est préfixe d'un autre. Un tel code est un code préfixe.

⦿ Ainsi on ne peut trouver aucun *dangling suffix qui est un mot code et l'algorithme du test de décodabilité ne fait aucune itération et se heurte directement à la condition 2éme d'arrêt.*

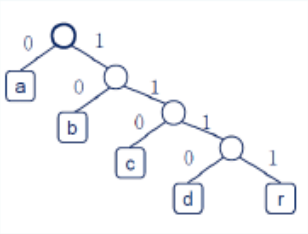
⦿ Pour vérifier si un code est un code préfixe il suffit simplement de tracer l'arbre binaire comme suit:

- On commence par un noeud racine.
- Chaque noeud a au maximum 2 branches.
- Convention: la branche à gauche=0 et la branche droite=1.
- Code préfixe: tout les mot-codes sont associés à des noeuds externes ou feuilles.

Vérification d'un code préfixe

Lettres	Code 2	Code 3	Code 4
a_1	0	0	0
a_2	1	10	01
a_3	00	110	011
a_4	11	111	0111
Arbre binaire			
Remarque	Code 2 n'est pas un code préfixe car il y a des mot-codes (a_1 et a_2) associés à des nœuds internes	Code 3 est un code préfixe tout les mot-codes sont associés à des nœuds internes	Code 4 n'est pas un code préfixe car il y a des mot-codes (a_1, a_2 et a_3) associés à des nœuds internes

Décodage d'un code préfixe

Lettres	Code
a	0
b	10
c	110
d	1110
r	1111
Arbre binaire	

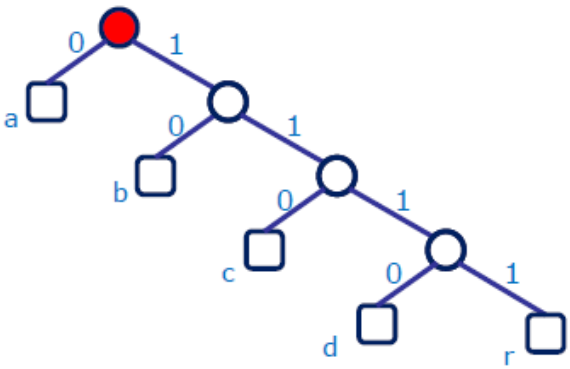
Séquence envoyée:

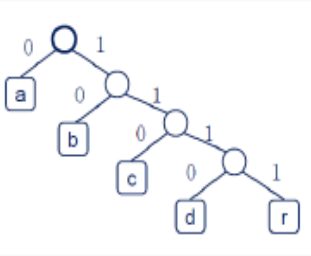
abracadabra = 0101111011001110010111110

Processus de décodage (1):

Reçu : 0101111011001110010111 1

Décodé : ----------



Lettres	Code
a	0
b	10
c	110
d	1110
r	1111
Arbre binaire	

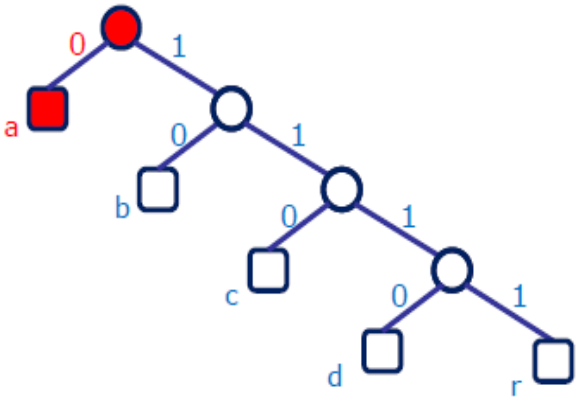
Séquence envoyée:

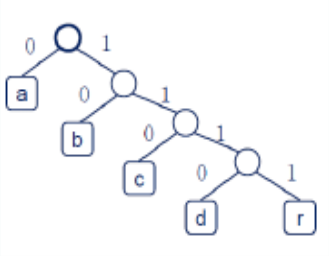
abracadabra = 0101111011001110010111110

Processus de décodage (2):

Reçu : 0101111011001110010111 10

Décodé : a----------



Lettres	Code
a	0
b	10
c	110
d	1110
r	1111
Arbre binaire	

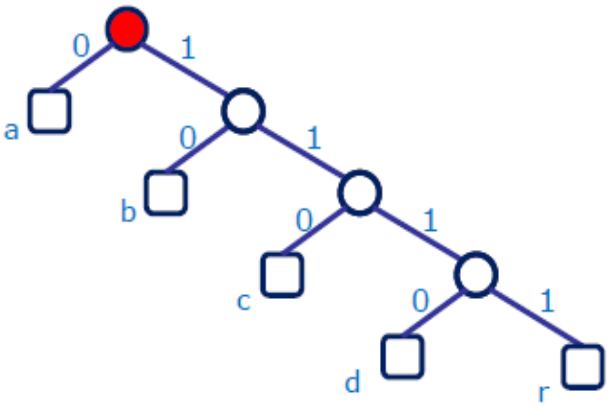
Séquence envoyée:

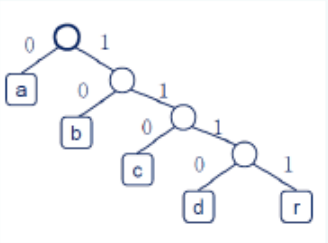
abracadabra = 0101111011001110010111110

Processus de décodage (3):

Reçu : 0101111011001110010111110

Décodé : a----------



Lettres	Code
a	0
b	10
c	110
d	1110
r	1111
Arbre binaire	

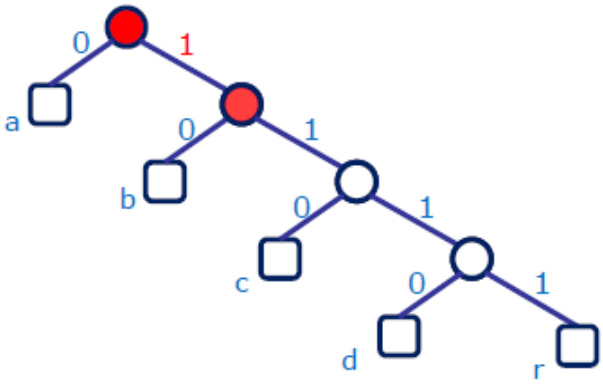
Séquence envoyée:

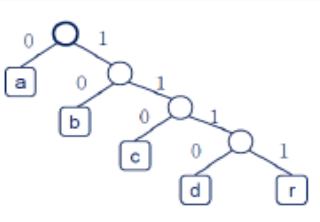
abracadabra = 0101111011001110010111110

Processus de décodage (4):

Reçu : 0101111011001110010111110

Décodé : a----------

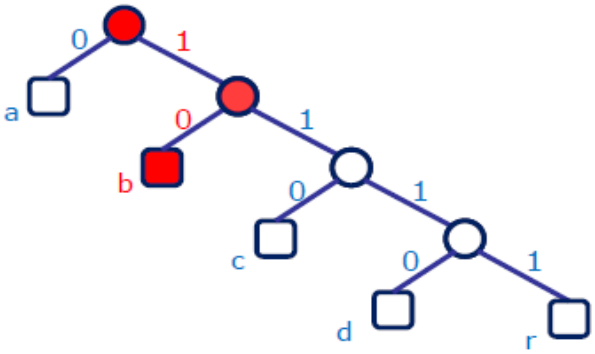


Lettres	Code
a	0
b	10
c	110
d	1110
r	1111
Arbre binaire	

Séquence envoyée:

abracadabra = 01011110110011100101111 0

Processus de décodage (5):

Reçu : 01011110110011100101111 0

Décodé : ab---------