Codage source sans perturbation

1. Introduction

- En général l'alphabet de la source diffère de l'alphabet du canal. Le but du codage de source est de permettre le passage de l'alphabet de la source à celui du canal.

- Afin que l'efficacité soit maximale, on réalise *une adaptation statistique de la source au canal,* soit :

$$C = \max(H(X)) = \log(D)$$

Ou D est le nombre de symboles de l'alphabet du canal. Le but du codage de source est donc de transformer la source primaire en une source a entropie maximale.

2. Codes à décodage unique

Soit une source $[S]=[s_1, s_2,\dots, s_N]$

dont les probabilités sont : $[P]=[p(s_1), p(s_2),\dots, p(s_N)]$

Soit $[X]=[x_1, x_2,\dots, x_D]$ l'alphabet du code (donc du canal)

Avec ces lettres on forme un nombre N de mots-code :
$[C]=[c_1, c_2,\dots, c_N]$

Les mots-code sont des successions finies de lettres de l'alphabet [X], le codage établit une relation bijective entre les symboles s_k de S et les mots C_k de C. On peut cependant à partir de X former des mots qui n'ont pas de correspondant dans S. Les mots auxquels correspondent des symboles de S, s'appellent *mots-code*.

Si les mots-code sont choisis convenablement on peut construire un code à décodage unique qui n'a pas besoin de signe séparateur entre les mots.

Exemple de code à décodage unique

S_k	Code A	Code B	Code C	Code D
S_1	00	0	0	0
S_2	01	10	01	10
S_3	10	110	011	110
S_4	11	1110	0111	111

Code A	S_1 S_2	S_2	S_1 S_2	S_2	S_3 S_4	S_3	S_3 S_4	S_4	S_3 S_4	S_3	S_1 S_2
	0	1	0	1	1	0	1	1	1	0	0
Code B	S_1	S_2 S_3 S_4	S_2	S_2 S_3 S_4	S_3 S_4	S_3	S_2 S_3 S_4	S_3 S_4	S_4	S_4	S_1
Code C	S_1 S_2 S_3 S_4	S_2 S_3 S_4 S_2	S_1 S_2 S_3 S_4	S_2 S_3 S_4	S_3 S_4 S_3	S_1 S_2 S_3 S_4	S_2 S_3 S_4	S_3 S_4	S_4	S_1 S_2 S_3 S_4 S_1	S_1 S_2 S_3 S_4

Code non instantané (retard de décodage)

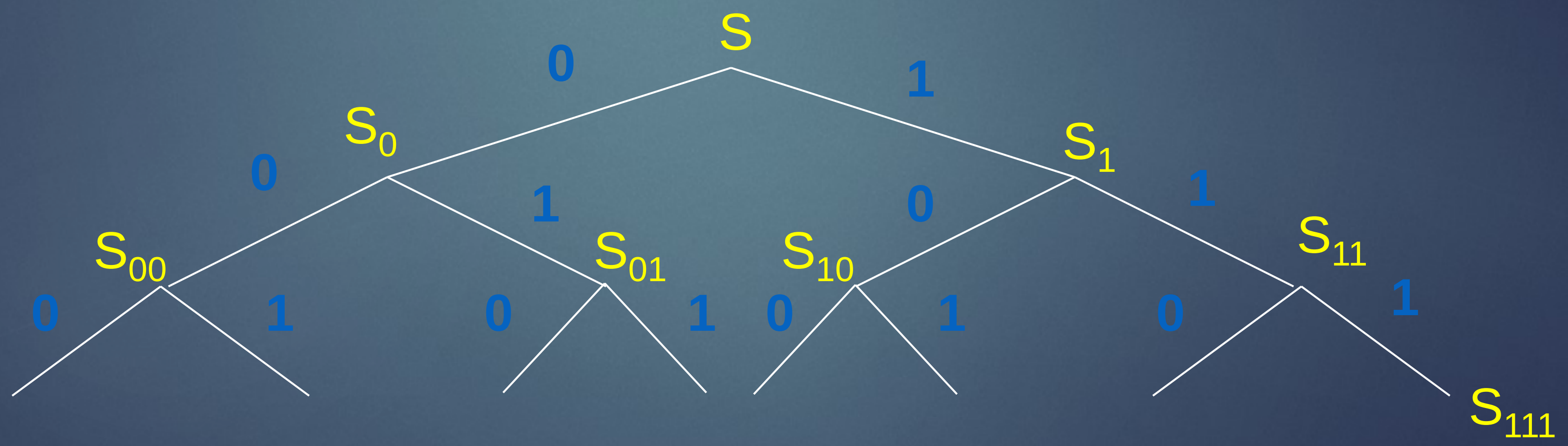
Conditions pour qu'un code soit instantané

Préfixe : Soit $c_i=x_{i1}, x_{I2}, \ldots, x_{im}$ un mot du vocabulaire d'un code. La suite de lettres $x_{i1}, x_{I2}, \ldots, x_{ik}$ avec $k<m$, s'appelle le préfixe du mot $c_{i.}$

La condition nécessaire et suffisante pour qu'un code soit instantané est qu'aucun mot du code ne soit le préfixe d'un autre mot du code.

Algorithme de construction d'un code binaire instantané

$$S=\left[\underbrace{s_1\, s_2,\ldots,s_k,}_{S_0}\underbrace{s_{k+1},\ldots\ldots,s_N}_{S_1}\right] \quad S_1=\left[\underbrace{s_{k+1},s_{k+2},\ldots\ldots s_l,}_{S_{10}}\underbrace{s_{l+1},\ldots\ldots,s_N}_{S_{11}}\right]\ldots\ldots$$



3. Longueur moyenne d’un mot code

Soit t_i le temps de transmission du mot-code c_i. Si le coût de transmission est fonction linéaire du temps de transmission, alors le coût moyen par message est :

$$\overline{C}=\sum_{i=1}^{N} t_i\, p(c_i)=\sum_{i=1}^{N} t_i\, p(s_i)$$

Si toutes les lettres x_i de l’alphabet [X] ont la même durée τ de transmission alors :

$$t_i = l_i \,.\, \tau$$ l_i : longueur du mot-code c_i

Si on considère pour simplifier que τ=1

$$t_i = l_i$$

$$\overline{C}=\sum_{i=1}^{N} p(s_i)\, l_i = \overline{l}$$

Le coût moyen de transmission est égal à la longueur moyenne d’un mot

4. Limite inférieure de la longueur moyenne d'un mot code

Soit une source $[S]=[s_1, s_2,\dots, s_N]$

dont les probabilités sont : $[P]=[p(s_1), p(s_2),\dots, p(s_N)]$

Soient les mots-code : $[C]=[c_1, c_2,\dots, c_N]$

dont les probabilités sont les mêmes que les messages de la source : $[P_c]= [P]=[p_1=p(s_1), p_2= p(s_2),\dots, p_N= p(s_N)]$

Les longueurs des mots-code sont : $[L]=[l_1, l_2,\dots, l_N]$

L'alphabet du code est : $[X]=[x_1, x_2,\dots, x_D]$

Entropie de la source est :

$$H(S)=H(C)=-\sum_{i=1}^{N} p(s_i)\log\left(p(s_i)\right)$$

Entropie de l'alphabet du code [X] est :

$$H(X)=-\sum_{i=1}^{N} p(x_i)\log\left(p(x_i)\right)$$

$$H(S)=H(C)=\bar{l}\,.H(X)$$

$$Max[H(S)]=Max[H(C)]\Rightarrow p(x_1)=p(x_2)=.....=p(x_D)=\frac{1}{D}$$

$$H(S)\leq\log(D)$$

donc :

$$H(S)=H(C)=\bar{l}\,.H(X)\leq\bar{l}\,.\log(D)$$

$$\boxed{\bar{l}\geq\frac{H(S)}{\log(D)}=\bar{l}_{\min}}$$

5. Capacité, efficacité et redondance du code

Capacité : $C = \max(H(X)) = \log(D)$

Efficacité du code : $\eta = \frac{\bar{l}_{\min}}{\bar{l}}$

$$\begin{cases} \bar{l}_{\min} = \frac{H(S)}{\log(D)} \\ \bar{l} = \frac{H(S)}{H(X)} \end{cases} \Rightarrow \boxed{\eta = \frac{H(S)}{\bar{l}\log(D)} = \frac{H(X)}{\log(D)}}$$

Redondance :

$$\boxed{\rho = 1 - \eta = \frac{\bar{l}\log(D) - H(S)}{\bar{l}\log(D)} = \frac{\log(D) - H(X)}{\log(D)}}$$

Exemple : [S]=[s_1, s_2, s_3 , s_4] et : [P]=[1/2, 1/4;, 1/8, 1/8]

$$H(S)=-\sum_{i=1}^{4} p(s_i)\log\left(p(s_i)\right)=\frac{7}{4}$$ bit/symbole

Soit [X]=[0,1] et le code suivant :

$$\boxed{\begin{array}{l} s_1 \longrightarrow 00 \\ s_2 \longrightarrow 01 \\ s_3 \longrightarrow 10 \\ s_4 \longrightarrow 11 \end{array}}$$

$$\bar{l}=\sum_{i=1}^{4} p(s_i)\,l_i=2.(\sum_{i=1}^{4} p(s_i))=2$$

$$\eta=\frac{H(X)}{\bar{l}\log(D)}=\frac{7/4}{2\log(2)}=\frac{7}{8}=0,875 \quad (D=2)$$

$$\rho=1-\eta=1-\frac{7}{8}=\frac{1}{8}=0.125$$

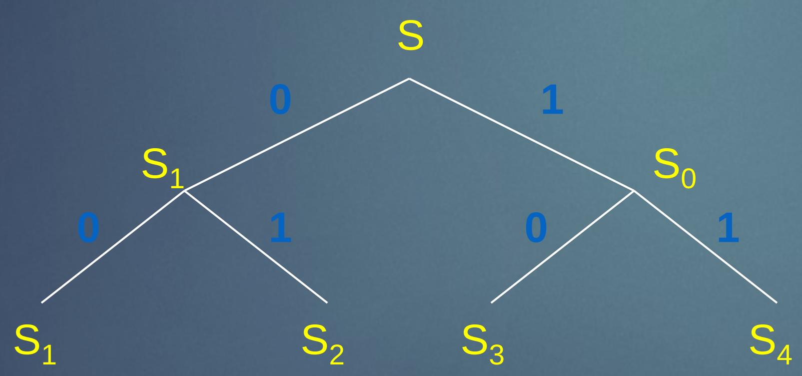
Soit [X]=[0,1] avec un autre le code :

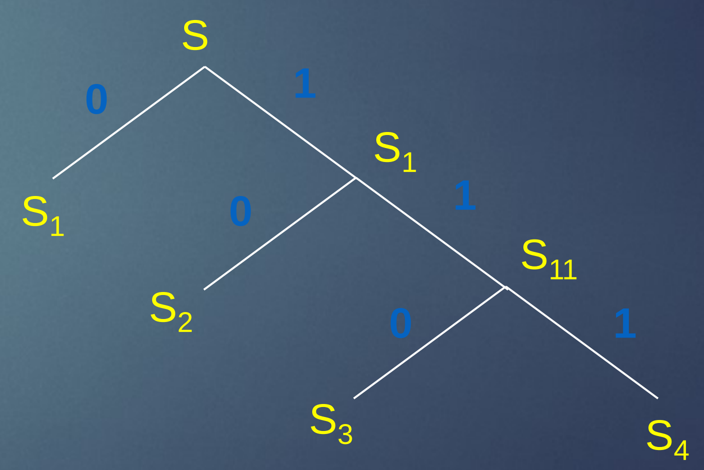
$$
\begin{aligned}
s_1 &\rightarrow 0\\
s_2 &\rightarrow 10\\
s_3 &\rightarrow 110\\
s_4 &\rightarrow 111
\end{aligned}
$$

$$\bar{l} = \sum_{i=1}^{4} p(s_i)\, l_i = 1.75$$

$$\eta = \frac{7/4}{1.75\log(2)} = 1$$

$$\rho = 1 - \eta = 0$$





6. Codes optimaux absolus

$$\frac{H(S)}{\log(D)} = \bar{l}_{\min} \Rightarrow H(S) = H(C) = \bar{l}_{\min} \log(D)$$

Ceci est vrai si $p(x_1) = p(x_2) = \ldots.\ p(x_D) = 1/D$

Dans ce cas : $\eta = \dfrac{H(X)}{\log(D)} = 1$: **code optimal absolu**

Les lettres de l'alphabet étant considérées comme indépendantes

$$p(s_i) = p(c_i) = \left(\frac{1}{D}\right)^{l_i} = D^{-l_i}$$

Comme : $\displaystyle\sum_{i=1}^{N} p(s_i) = 1$ alors : $\displaystyle\sum_{i=1}^{N} D^{-l_i} = 1$

C 'est une condition nécessaire et suffisante pour qu'il existe un code absolu

$$\boxed{\sum_{i=1}^{N} D^{-l_i} \leq 1}$$: Inégalité de Mc Millan (code irréductible)

7. Premier théorème de Shannon

On a vu que : $p(s_i) = \left(\frac{1}{D}\right)^{l_i} \Rightarrow l_i = \frac{-\log(p(s_i))}{\log(D)}$

Etudions ce qui arrive lorsque les probabilités d'apparition des messages à coder sont arbitraires.

Dans ce cas $r_i = \frac{-\log(p(s_i))}{\log(D)}$ n'est généralement pas un nombre entier

La longueur du mot c_i du code [C] est alors choisie comme suit :

$$\frac{-\log(p(s_i))}{\log(D)} \leq l_i \leq \frac{-\log(p(s_i))}{\log(D)} + 1$$

l_i est le nombre entier le plus proche de r_i :

Il faut vérifier si les longueurs l_i satisfont l'inégalité de Mc Millan autrement dit si on peut former un code absolu (irréductible).

$$\frac{-\log(p(s_i))}{\log(D)} \le l_i \Leftrightarrow \log(p(s_i)) \ge l_i \log(D) = \log(D^{l_i})$$

ou encore : $D^{-l_i} \le p(s_i)$

d'où : $\sum_{i=1}^{N} D^{-l_i} \le \sum_{i=1}^{N} p(s_i) = 1$

Il existe donc un code absolu (irréductible) ayant des mots de longueurs l_i

à partir de :

$$\frac{-\log(p(s_i))}{\log(D)} \le l_i \le \frac{-\log(p(s_i))}{\log(D)} + 1$$

On obtient :

$$\frac{-\sum_{i=1}^{N} p(s_i)\log(p(s_i))}{\log(D)} \le \bar{l} \le \frac{-\sum_{i=1}^{N} p(s_i)\log(p(s_i))}{\log(D)} + 1$$

soit : $$\frac{H(S)}{\log(D)} \leq \bar{l} \leq \frac{H(S)}{\log(D)} + 1$$

Ceci est vrai pour toute source sans mémoire. En particulier pour une source fictive [S^n] où chaque symbole est constitué à partir d’une succession de n symboles de la source [S]. Dans ce cas on montre que l’entropie de la source [S^n] est :

$$H(S^n) = nH(S)$$

De cette façon au lieu de coder symbole par symbole, on fait un codage par groupe de n symboles, on note $\bar{l}_n$ la longueur moyenne d’un mot-code on a alors :

$$\frac{H(S^n)}{\log(D)} \leq \bar{l}_n \leq \frac{H(S^n)}{\log(D)} + 1$$

ou encore

$$\frac{H(S)}{\log(D)} \leq \frac{\bar{l}_n}{n} \leq \frac{H(S)}{\log(D)} + \frac{1}{n}$$

à la limite si n est très grand on a :

$$\lim_{n\to\infty}\left(\frac{\bar{l}}{n}\right)=\frac{H(S)}{\log(D)}=\bar{l}$$

ou $\bar{l}$ est la longueur moyenne d'un mot code du code [C]

$$\Rightarrow \frac{H(S)}{\bar{l}}=\log(D)$$

$$\frac{H(S)}{\bar{l}}$$

peut être amenée aussi proche que l'on veut de la capacité du code log(D)

Ceci constitue le premier théorème de Shannon

Bien sur en pratique n aura toujours une valeur finie et on essaiera de construire des codes dont l'efficacité est proche de 1.

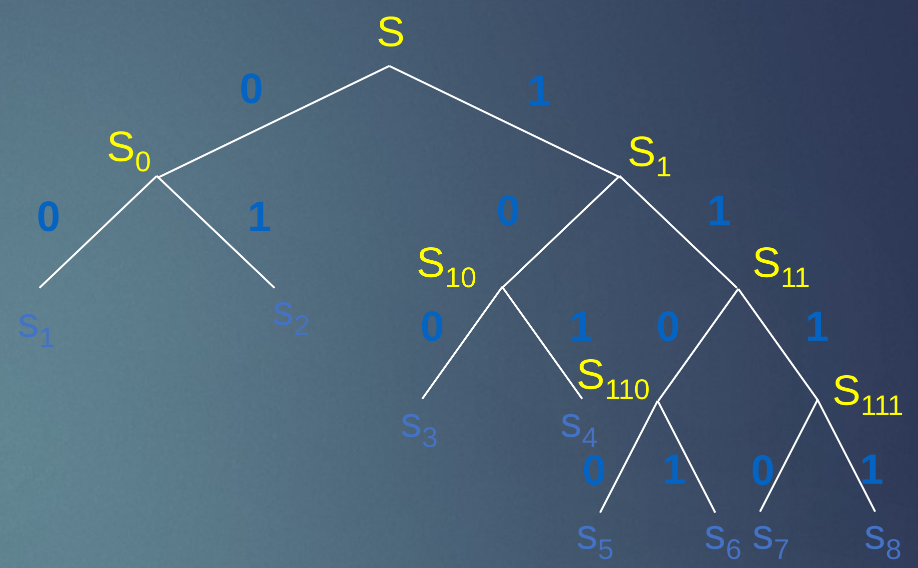
8. Codage de Shannon-Fano

Soit une source $[S]=[s_1, s_2,\ldots, s_N]$ qui peut être divisée en deux ensembles S_0 et S_1 dont les probabilités sont $p(S_0)=p(S_1)=$ ½, en supposant à nouveau que S_0 et S_1 puissent être divisées en deux ensembles S_{00}, S_{01} et S_{10}, S_{11} avec les probabilités sont $p(S_{00})=p(S_{01})=p(S_{10})=p(S_{11})=$ 1/4, et ainsi de suite jusqu'à ce que les ensembles en question ne contiennent plus qu'un élément **alors le codage sera absolu**

Exemple : [S]=[s_1, s_2, s_3 , s_4, s_5, s_6, s_7 , s_8] et

[P]=[1/4, 1/4, 1/8, 1/8 1/16, 1/16, 1/16, 1/16]

s_k	$P(s_k)$					c_k	l_k
s_1	0,25	0	0			00	2
s_2	0,25		1			01	2
s_3	0,125		0	0		100	3
s_4	0,125			1		101	3
s_5	0,0625	1		0	0	1100	4
s_6	0,0625		1		1	1101	4
s_7	0,0625			1	0	1110	4
s_8	0,0625				1	1111	4



$$H(S) = -\sum_{i=1}^{8} p(s_i)\log\big(p(s_i)\big) = 2,75$$

La longueur moyenne d'un mot-code est :

$$\bar{l} = \sum_{i=1}^{8} l_i\, p(s_i) = 2,75 \quad \bar{l}_{\max} = \frac{H(S)}{\log(D)} = 2,75\ (D=2)$$

$$\boxed{\eta = 1}$$

9. Codage binaire de Huffman

Pour toute source discrète sans mémoire X(n), il existe un code instantané optimal représentant exactement cette source et uniquement décodable

Algorithme de Huffman

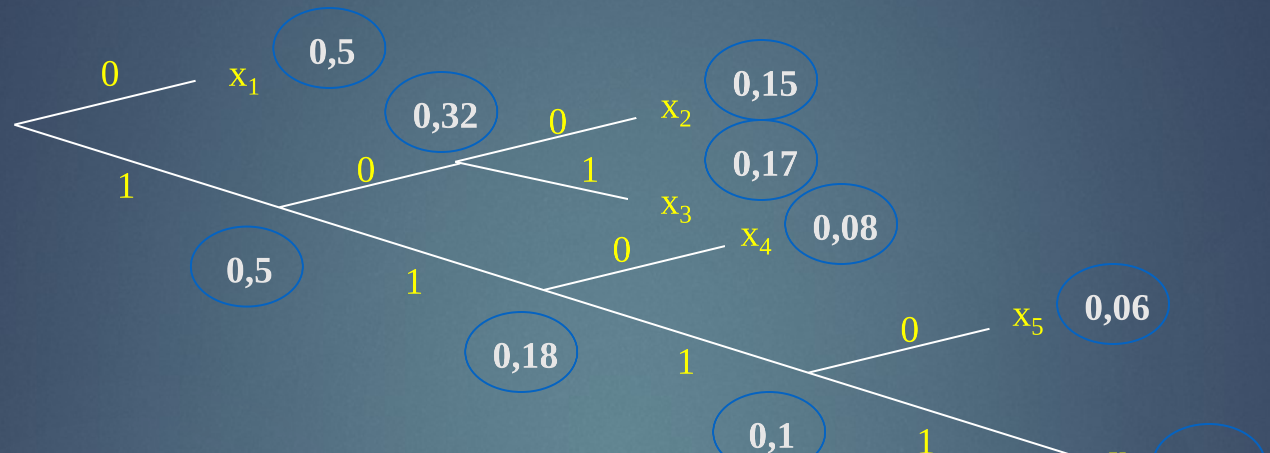
- On construit un arbre en partant des noeuds terminaux.
- On part de deux listes $\{x^1, \ldots, x^{Lx}\}$ et $\{px(1), \ldots, px(Lx)\}$,
- On sélectionne les deux symboles les moins probables, on crée deux branches dans l'arbre et on les étiquettes par les deux symboles binaires 0 et 1.
- On actualise les deux listes en rassemblant les deux symboles utilisés en un nouveau symbole et en lui associant comme probabilité la somme des deux probabilités sélectionnées.
- On recommence les deux étapes précédentes tant qu'il reste plus d'un symbole dans la liste.

cet algorithme est l'algorithme optimal

(Longueur moyenne des mots la plus faible)

Exemple:

Symboles	x_1	x_2	x_3	x_4	x_5	x_6
Probabilités	0,5	0,15	0,17	0,08	0,06	0,04



La longueur moyenne du code est :

$$\bar{l}=\sum_{i=1}^{Lx} px(x)l(c^i)=2,1 bits$$

L'entropie est de 2,06 valeur très proche de la longueur moyenne obtenue par Huffman.

$$\eta=\frac{H(S)}{\bar{l}\log(2)}=\frac{2,06}{2,1}=0,981$$

Exercices : 6, 7