Chapitre 5

CODAGE STATITIQUE

Présentation

Entropie

Codage de l 'information

- Codage de l 'information ≡ représentation codée de l 'information

- **Rôles multiples du codage**
  - préparation de la transformation message => signal transmis
  - adaptation débit source - capacité du canal (compression)
  - codage de protection contre les erreurs de transmission (détection / correction d'erreurs)
  - chiffrage (secret des communications)
  - tatouage (marquage du titre de propriété)
  - transcodage (changement d 'alphabet: contraintes de transmission)

Un système de communication a pour but d'acheminer des messages d'un expéditeur (source d'information) vers un destinataire (utilisateur de l'information). Le signal supportant l'information à transmettre doit être compatible avec les caractéristiques du canal de transmission. Le **codage de l'information** doit établir une correspondance injective entre le message transmis par la source et la suite des symboles d'informations $\{b_n\}$ transmis à l'émetteur.

Ce codage a des rôles multiples. Il prépare la transformation du message en un signal (effectué dans la partie de l'émetteur : codage information à signal). Il adapte la source d'information à la capacité de transmission du canal. Il permet de protéger l'information contre les erreurs de transmission (dans une certaine limite) en vue de leur détection ou/et de leur correction (dues aux perturbations dans le canal). Il permet également dans certains cas d'assurer le secret des communications (le chiffrage), et de tatouer pour marquer une propriété.

Un canal de transmission donné doit pouvoir transporter des messages de nature variée d'où la nécessité d'un transcodage permettant de transformer la représentation des messages à partir d'un dictionnaire M_k utilisant l'alphabet A_k en une représentation des mêmes messages à partir d'un dictionnaire M_0 utilisant l'alphabet A_0. Cet alphabet particulier est souvent l'ensemble binaire {0, 1}, mais ce n'est pas le seul.

Codage de l 'information

➢ *Définitions*

- Sources S de messages: production d'une suite de messages, chacun d'eux étant sélectionné dans un ensemble M des messages

  (M : dictionnaire de messages possibles M = { m_1 , m_2 , ….}, les m_i sont aussi appelés « mots »)

- Message : suite finie de symboles

  (caractères pris dans A : alphabet)

- Alphabet : ensemble fini de symboles A = { a_1, a_2, ……, a_k }

♦ *Définitions* :

On appelle **message** toute suite finie de caractères pris dans un **alphabet** A : ensemble fini de **symboles** (par exemple : lettres, chiffres, …).

On appelle **source de messages** S, le système qui produit une suite temporelle de messages m_i, chacun d'eux étant pris dans l'ensemble M des messages possibles. M est appelé **dictionnaire de messages** (ou mots). Le message transmis est en fait un texte formé par des règles syntaxiques de messages élémentaires appelés mots : M = {m_1, m_2, … }, chaque mot s'écrit par une suite finie fixée de symboles pris dans un alphabet A.

En fonction des applications envisagées, les sources de message S peuvent utiliser des dictionnaires de nature très différentes : depuis les messages écrits utilisant les caractères de l'alphabet, les chiffres, les caractères de ponctuation et de tabulation jusqu'aux messages visuels où les messages sont des images numérisées où, par exemple, chaque mot est un pixel s'écrivant comme la suite de 8 symboles (binaires) pris sur l'alphabet {0, 1} (octet).

*Entropie d 'une source (*SHANNON 1948*)*

- Définition de l '***incertitude*** et de l '***entropie***
  - **Incertitude** I d 'un événement E :

    $I(E) = -\log_2 Pr\{E\}$ *Unités*: bit (BInary uniT si $\log_2$)

    nat (NAtural uniT si Log_e): 1 nat=1,443 bit

    si source simple s_n => $I(s_n) = \sum_{i=1;n} I(m_{\alpha i})$
  - **Entropie** H d 'une variable aléatoire discrète X :

    $H(X) = E_X [I(X)] = \sum_{i=1;n} p_i I(X_i) = - \sum_{i=1;n} p_i \log_2(p_i)$
  - Propriétés de l 'entropie
    - $H \geq 0$; H est continue , symétrique ; $H(p_1, \ldots, p_N) \leq \log_2 n$
    - si $(p_1, \ldots, p_n)$ et $(q_1, \ldots q_n)$ sont 2 répartitions de probabilités

      ==> $\sum_{i=1;n} p_i \log_2(q_i / p_i) \leq 0$ car $Log\ x < x - 1$

L'***incertitude*** I d'un événement E de probabilité Pr(E) est définie par :

$$I(E) = \log_2 \frac{1}{Pr(E)} = - \log_2 Pr(E)$$

- Remarques :

- si Pr(E) = 1/2 alors I (E) = 1 (incertitude unité)
- si Pr(E) = 1 alors I (E) = 0 : l'incertitude est nulle pour un événement certain.
- L'unité est le ***bit*** (*Binary Unit*) à ne pas confondre avec le bit : *Binary digit*.
- On peut utiliser la fonction logarithme népérien à la place du logarithme en base 2, dans ce cas l'unité est le ***nat*** (Natural Unit = 1,443 bit).

On considère maintenant que les événements E sont en fait des réalisations d'une variable aléatoire discrète X. On définit alors l'***entropie*** H comme étant l'incertitude moyenne de la variable aléatoire X. Si on considère en fait chaque événement x_i, $i \in [1, n]$, comme une réalisation d'une variable aléatoire X (i.e. X est une variable aléatoire à valeurs dans $\{ x_1, x_2, \ldots, x_n \}$) :

$$H(X) = E_X \{ I(X) \} = \sum_{i=1..n} \Pr\{X = x_i\} \, . \, I(x_i) = \sum_{i=1..n} p_i . I(x_i), \text{ avec } p_i = \Pr\{X = x_i\}$$

L'entropie dépend de la loi de probabilité de X mais n'est pas fonction des valeurs prises par X. Elle s'exprime en bits (ou nats) et représente le nombre moyen de bits nécessaires à coder en binaire les différentes réalisations de X.

On considère maintenant une source d'informations S définie par un ensemble de message m_i possibles (dictionnaire) : $S\{m_1, m_2, \ldots, m_N\}$, et par un mécanisme d'émission de messages tel que :

$s_n = \{m_{\alpha 1}, m_{\alpha 2}, \ldots, m_{\alpha n}\}$ avec $m_{\alpha 1}$: 1^{er} message émis, …, $m_{\alpha n}$: $n^{ième}$ message émis.

Attention : l'indice « i » dans αi définit l'indice temporel dans la suite de messages émis par la source. αi définit l'indice du $i^{ème}$ message émis dans le dictionnaire M des messages possibles, en général : $N \neq n$.

Le choix des $m_{\alpha i}$ s'effectue selon une loi de probabilité donnée. L'émission d'une source discrète d'informations correspond donc à une suite de variables aléatoires X_i, $i \in [1, n]$: une variable aléatoire associée à la réalisation de chaque message émis $m_{\alpha i}$ de la source.

La probabilité de s_n s'exprime alors comme un produit de probabilités conditionnelles :

$$\Pr(s_n) = \Pr\{X_1 = m_{\alpha 1}\} \Pr\{X_2 = m_{\alpha 2} / X_1 = m_{\alpha 1}\} \ldots \Pr\{ X_n = m_{\alpha n} / X_1 = m_{\alpha 1}, \ldots, X_{n-1} = m_{\alpha n-1}\}$$

Dans le cas de sources simples, les n variables aléatoires X_i sont indépendantes et de même loi donc :

$$\forall (i, j) \in [1, n] \times [1, N], \Pr\{ X_i = m_j \} = p_j, \text{ et } \Pr\{s_n\} = p_{\alpha 1} . p_{\alpha 2} \ldots p_{\alpha n}$$

$$\Rightarrow I (s_n) = - \log_2 \Pr\{s_n\} = - \log_2 (p_{\alpha 1} . p_{\alpha 2} \ldots p_{\alpha n}) = \sum_{i=1..n} - \log_2 p_{\alpha i} = \sum_{i=1..n} I (m_{\alpha 1})$$

$$\boxed{I (s_n) = \sum_{i=1..n} I (m_{\alpha 1})}$$

Dans le cas d'une source discrète de « n » messages m_i, ou chaque message m_i est associé à la probabilité p_i, l'entropie H de la source S est donnée par :

$$\boxed{H(S) = - \sum_{i=1}^{n} p_i \, . \log_2 p_i}$$

♦ *Propriétés de l'entropie :*

- Comme $0 \leq p_i \leq 1$ et que $\sum_{i=1}^{n} p_i = 1$, alors $H(X) > 0$: l'entropie est positive.
- Soient $(p_1, p_2, \ldots, p_n)$ et $(q_1, q_2, \ldots, q_n)$ deux lois de probabilité, alors $\sum_{i=1}^{n} p_i \log_2 \frac{q_i}{p_i} \leq 0$.

  En effet, $\forall x > 0$, on a $\ln x \leq x - 1$. D'où $\ln \frac{q_i}{p_i} \leq \frac{q_i}{p_i} - 1$, soit $\log_2 \frac{q_i}{p_i} \leq \frac{1}{\ln 2}(\frac{q_i}{p_i} - 1)$

  donc $\sum_{i=1}^{n} p_i \log_2 \frac{q_i}{p_i} \leq \frac{1}{\ln 2} \sum_{i=1}^{n} p_i \left(\frac{q_i}{p_i} - 1 \right) = \frac{1}{\ln 2} \left(\sum_{i=1}^{n} q_i - \sum_{i=1}^{n} p_i \right) = \frac{1}{\ln 2} (1 - 1) = 0$.
- L'entropie d'une variable aléatoire X à n valeurs possibles est maximale et vaut $\log_2 n$ lorsque la loi de X est uniforme. En prenant $q_1 = q_2 = \ldots = q_n = \frac{1}{n}$ (loi uniforme), dans la propriété précédente :

$$\sum_{i=1}^{n} p_i \log_2 \frac{q_i}{p_i} \leq 0 \quad \Leftrightarrow -\sum_{i=1}^{n} p_i \log_2 p_i \leq -\sum_{i=1}^{n} p_i \log_2 q_i$$

$$\Leftrightarrow H(X) \leq -\sum_{i=1}^{n} p_i \log_2 \frac{1}{n}$$

$$\Leftrightarrow H(X) \leq -\log_2 \frac{1}{n} \sum_{i=1}^{n} p_i = \log_2 n$$

- L'entropie est continue et symétrique.

Dans la suite de ce cours, nous utiliserons systématiquement la base 2 du logarithme.

Exemple simple :

On considère une source S, à loi uniforme, qui envoie des messages à partir de l'alphabet français à 26 caractères (a,b,c, …, z). On ajoute à cet alphabet le caractère « espace » comme séparateur des mots.

L'alphabet est donc constitué de 27 caractères : $H(S) = -\sum_{i=1}^{27} \frac{1}{27} \log_2 \frac{1}{27} = \log_2(27) = 4{,}75$ bits d'information par caractère. En réalité, l'entropie est voisine de 4 bits d'information par caractère sur un très grand nombre de textes français.

Chapitre 5

CODAGE STATISTIQUE

Codage de Huffman

Codage statistique optimal

- ***Définitions***:
  - S: source discrète et simple de messages m_i de loi de probabilité $p = (p_1, \ldots\ldots\ldots\ldots, p_N)$ (source homogène)
  - Codage sur un alphabet $A = \{ a_1, a_2, \ldots\ldots, a_q \}$
  - Entropie de la source H(S) et longueur moyenne des mots-codes E(n)
- ***Théorème de Mac Millan***:
  - il existe au moins un code irréductible déchiffrable vérifiant :

$$H / \log2\ q \leq E(n) < (H / \log2\ q) + 1$$

$\Rightarrow$ égalité si p_i de la forme : $p_i = q^{-ni}$ (si $q = 2$ => $n_i = - \log2\ p_i$)

- ***Théorème de SHANNON*** (1er théorème sur le codage sans bruit)

$$H / \log2\ q \leq E(n) < (H / \log2\ q) + \varepsilon$$

Nous avons vu dans la ressource précédente : « Définition du codage et propriétés» que les objectifs du codage sont principalement de transcrire des informations et de réduire la quantité de symboles nécessaires à la représentation de l'information. En « optimisant le codage », on souhaite donc réduire au maximum cette quantité de symboles :

On considère une source simple d'information homogène $S = \{m_1, m_2, \ldots, m_N\}$ munie d'une loi de probabilité $p = \{ p_1, p_2, \ldots, p_N \}$ où $p_i = Pr\{m = m_i\}$.
Si M_i est le mot-code correspondant au message m_i, on appelle $n_i = n(M_i)$ le nombre de caractères appartenant à l'alphabet A (Card(A) = q) nécessaires au codage de m_i, n_i est donc la longueur du mot-code M_i.

La longueur moyenne des mots-codes est donc : $E(n) = \sum_{i=1}^{N} p_i n_i$.

L'incertitude moyenne d'une source est l'entropie H. L'incertitude moyenne par caractère de l'alphabet A est donc égale à $\frac{H}{E(n)}$, elle vérifie : $\frac{H}{E(n)} \leq \log_2 q$ car l'alphabet A comporte « q » caractères. De cette inégalité, on déduit que $E(n) \geq \frac{H}{\log_2 q}$.

Pour optimiser le codage, on souhaite donc réduire E(n), la longueur moyenne des mots-codes. Cette longueur moyenne ne peut pas être rendue inférieure à $\frac{H}{\log_2 q}$. Néanmoins, deux théorèmes démontrent qu'il est possible d'obtenir l'égalité $E(n) = \frac{H}{\log_2 q}$ et d'obtenir un code optimal :

- ***Théorème de Mac Millan :***

Une source d'information S d'entropie H codée de façon déchiffrable par un alphabet à q caractères est telle que : $E(n) \geq \frac{H}{\log_2 q}$ et il existe au moins un code irréductible d'une loi donnée qui vérifie : $E(n) \leq \frac{H}{\log_2 q} + 1$. L'égalité est atteinte si $p_i = q^{-ni}$ (i.e. $n_i = - \log_q p_i$) et on a alors un codage optimal.

Remarque :

Dans le cas particulier d'un l'alphabet binaire {0, 1}, on a q = 2. Si la relation $n_i = - \log_2 p_i$ est vérifiée, on a alors E(n) = H : l'entropie est la borne inférieure de l'ensemble des longueurs moyennes des mots-codes et cette borne inférieure est atteinte.

- ***Théorème de SHANNON sur le codage sans bruit :***

Toute source d'information homogène est telle qu'il existe un codage irréductible pour lequel la longueur moyenne des mots-codes est aussi voisine que l'on veut de sa borne inférieure $\frac{H}{\log_2 q}$. La démonstration de ce théorème utilise le codage irréductible en blocs (le codage en bloc affecte à chaque bloc de « k » messages de S, consécutifs ou non, un mot-code).

Codage statistique optimal

- ***Codes de Fano - Shannon***

- ***Codes arithmétiques*** *(codage en bloc, de type codage d'intervalles)*
  possibilités d'adaptation en ligne

- ***Codes de Huffman***
  3 principes de base:
  - si $p_i < p_j \Rightarrow n_i \geq n_j$
  - les 2 codes les moins probables sont de même longueur
  - les 2 codes les moins probables (de longueur max) ont le même préfixe de longueur n_{max}-1

La présentation ci-dessus met en avant trois types de codes qui s'approchent du code optimal :

♦ ***Le codage de Fano- Shannon :***

Il essaie d'approcher au mieux le code irréductible le plus compact, mais la probabilité p_i ne s'écrit généralement pas 2^{-ni} et donc le codage ne peut qu'approcher le codage optimal.
Les probabilités p_i associées aux messages m_i sont ordonnées par ordre décroissant puis on fixe n_i tel que : $n_i \geq \log_2 \frac{1}{p_i} \geq n_i - 1$. Enfin, on choisit chaque mot code M_i de longueur n_i de sorte qu'aucun des mots-codes choisis avant n'en soit un préfixe (évite les ambiguïtés au décodage cf. : « *Classification des codes* »).

♦ ***Le codage arithmétique :***

Le code est associé à une séquence de messages émis par la source et non pas à chaque message. Contrairement au code de Huffman qui doit avoir une longueur entière de bits par message, et qui ne permet donc pas toujours de réaliser une compression optimale, le codage arithmétique permet de coder un message sur un nombre non entier de bits : c'est une méthode plus performante, mais aussi plus lente. Les différents aspects de ce codage sont amplement développés dans la ressource : « *Codage statistique : codage arithmétique* ».

- ***Le codage de Huffman :***

Le code de Huffman est le code irréductible optimal. Il s'appuie sur trois principes :

- si $p_j > p_i$ alors $n_i \leq n_j$,
- les deux mots les moins probables ont des longueurs égales,
- ces derniers sont écrits avec les mêmes n_{max}-1 premiers caractères.

En utilisant itérativement cette procédure, on construit les mots-codes M_i des messages m_i.

- Exemple :

On considère la source $S = \{m_1, m_2, \ldots, m_8\}$ munie de la loi de probabilité : $p_1 = 0,4$; $p_2 = 0,18$; $p_3 = p_4 = 0,1$; $p_5 = 0,07$; $p_6 = 0,06$; $p_7 = 0,05$; $p_8 = 0,04$.

On place ces probabilités par ordre décroissant dans la colonne $p_i^{(0)}$ du tableau ci-dessous. On remarque que dans la colonne $p_i^{(0)}$ les probabilités des messages m_7 et m_8 sont les plus faibles, on les somme alors puis on réordonne, toujours par ordre décroissant, les probabilités afin de créer la colonne $p_i^{(1)}$:

Messages	$p_i^{(0)}$	$p_i^{(1)}$
m_1	0,4	0,4
m_2	0,18	0,18
m_3	0,1	0,1
m_4	0,1	0,1
m_5	0,07	0,09
m_6	0,06	0,07
m_7	0,05	0,06
m_8	0,04	

De manière générale, on somme les deux probabilités les plus faibles de la colonne $p_i^{(k)}$, puis on réordonne les probabilités par ordre décroissant afin d'obtenir la colonne $p_i^{(k+1)}$. Finalement on a donc le tableau suivant :

Messages	$p_i^{(0)}$	$p_i^{(1)}$	$p_i^{(2)}$	$p_i^{(3)}$	$p_i^{(4)}$	$p_i^{(5)}$	$p_i^{(6)}$
m_1	0,4	0,4	0,4	0,4	0,4	0,4	0,6
m_2	0,18	0,18	0,18	0,19	0,23	0,37	0,4
m_3	0,1	0,1	0,13	0,18	0,19	0,23	
m_4	0,1	0,1	0,1	0,13	0,18		
m_5	0,07	0,09	0,1	0,1			
m_6	0,06	0,07	0,09				
m_7	0,05	0,06					
m_8	0,04						

On attribue les bits ‘0’ et ‘1’ aux deux derniers éléments de chaque colonne :

Messages	$p_i^{(0)}$	$p_i^{(1)}$	$p_i^{(2)}$	$p_i^{(3)}$	$p_i^{(4)}$	$p_i^{(5)}$	$p_i^{(6)}$
m_1	0,4	0,4	0,4	0,4	0,4	0,4	0,6 '0'
m_2	0,18	0,18	0,18	0,19	0,23	0,37 '0'	0,4 '1'
m_3	0,1	0,1	0,13	0,18	0,19 '0'	0,23 '1'	
m_4	0,1	0,1	0,1	0,13 '0'	0,18 '1'		
m_5	0,07	0,09	0,1 '0'	0,1 '1'			
m_6	0,06	0,07 '0'	0,09 '1'				
m_7	0,05 '0'	0,06 '1'					
m_8	0,04 '1'						

Pour chaque message m_i, on parcourt le tableau de gauche à droite et on détecte dans chaque colonne la probabilité associée $p_i^{(k)}$ (chemin en bleu sur la figure ci-dessous).
Le mot-code M_i est alors obtenu en partant de la dernière colonne (à droite) et en remontant vers la première colonne (à gauche), tout en sélectionnant les bits associés aux probabilités $p_i^{(k)}$ du message m_i (rectangles verts sur la figure ci-dessous).

On propose par exemple de déterminer le mot-code M_6 du message m_6. On détecte donc toutes les probabilités $p_6^{(k)}$:

Messages	$p_i^{(0)}$	$p_i^{(1)}$	$p_i^{(2)}$	$p_i^{(3)}$	$p_i^{(4)}$	$p_i^{(5)}$	$p_i^{(6)}$
m_1	0,4	0,4	0,4	0,4	0,4	0,4	0,6 '0'
m_2	0,18	0,18	0,18	0,19	0,23	0,37 '0'	0,4 '1'
m_3	0,1	0,1	0,13	0,18	0,19 '0'	0,23 '1'	
m_4	0,1	0,1	0,1	0,13 '0'	0,18 '1'		
m_5	0,07	0,09	0,1 '0'	0,1 '1'			
m_6	0,06	0,07 '0'	0,09 '1'				
m_7	0,05 '0'	0,06 '1'					
m_8	0,04 '1'						

Le mot-code M_6 est alors obtenu par simple lecture de droite à gauche des bits contenus dans les rectangles verts : '0' – '1' – '0' – '1'. En procédant de même pour chacun des messages, on obtient :

Messages	code de Huffman
m_1	1
m_2	0 0 1
m_3	0 1 1
m_4	0 0 0 0
m_5	0 1 0 0
m_6	0 1 0 1
m_7	0 0 0 1 0
m_8	0 0 0 1 1

La longueur moyenne des mots-codes vaut donc :

$$E(n) = \sum_{i=1}^{8} p_i \, n_i = 0{,}4\times1 + 0{,}18\times3 + 0{,}1\times3 + 0{,}1\times4 + 0{,}07\times4 + 0{,}06\times4 + 0{,}05\times5 + 0{,}04\times5$$

$$E(n) = 2{,}61$$

On peut comparer cette grandeur avec l'entropie H de la source :

$$H = -\sum_{i=1}^{8} p_i \, . \log_2 p_i = 2{,}552$$

L'efficacité η du code de Huffman pour cet exemple est donc de $\frac{2{,}552}{2{,}61} = 97{,}8\ \%$. À titre de comparaison, il faut 3 bits pour coder en binaire naturel 8 messages différents ($2^3 = 8$). Pour cet exemple, l'efficacité du code binaire naturel est donc de seulement $\frac{2{,}552}{3} = 85\ \%$.

Chapitre 5

CODAGE STATISTIQUE

Codage Arithmétique

Codage arithmétique

➢ ***Principes de base :***

- le code est associé à la séquence des symboles m_i (messages), et non pas à chaque symbole de la séquence.

- on code des intervalles $[c_i, d_i[$ pour chaque symbole.

- itération jusqu'à avoir codé la séquence entière

- on code une séquence par un unique nombre réel appartenant à l'intervalle [0, 1 [.

Le codage arithmétique permet, à partir de la probabilité d'apparition des symboles d'une source de créer un seul mot-code qui soit associé à une séquence de longueur arbitraire de symboles. Ceci diffère du code de Huffman qui attribue des mots-codes de longueurs variables à chaque symbole de la source.
Le code associé à une séquence est un nombre réel de l'intervalle [0, 1[. Ce code est construit par subdivisions récursives d'intervalles. Un intervalle est subdivisé pour chaque nouveau symbole qui appartient à la séquence. On obtient, en définitive, un sous-intervalle de l'intervalle [0, 1[tel que tout nombre réel appartenant à cet intervalle représente la séquence à coder.

Codage arithmétique

➢ *Définitions:*

- Soit une source $S = \{s_1, s_2, \ldots, s_N\}$ avec $p_k = P(s_k)$

- $[Ls_k, Hs_k [$ est l'intervalle « ***assigné*** » au symbole s_k

  avec: $Hs_k - Ls_k = p_k$

➢ *Algorithme de codage:*

1) Initialisation: $L_c = 0$; $H_c = 1$
2) Déterminer les longueurs des sous intervalles liés aux symboles
3) Passer au codage sur le prochain symbole s_k
4) Déterminer, pour le sous-intervalle considéré, le raffinement en sous intervalles
5) Répéter les étapes 2, 3, et 4 jusqu'à obtenir le code de la séquence entière

Soit $S = \{s_1, \ldots, s_N\}$ une source pouvant délivrer N symboles et munie des probabilités suivantes : pour tout entier k de [1, N], $P\{s_k\} = p_k$.

Pour coder une séquence $s_M = \{s_{\alpha 1}, s_{\alpha 2}, \ldots, s_{\alpha M}\}$ de M symboles, on utilise l'algorithme suivant ($s_{\alpha k}$ est le $k^{ème}$ symbole se la séquence émise par la source) :

- Étape 1 :

On initialise un premier intervalle avec deux bornes : la borne inférieure $L_c = 0$ et la borne supérieure $H_c = 1$ (correspondant à la probabilité de choisir un premier symbole $s_{\alpha 1}$ parmi tous les symboles s_k de la source).
La taille ce cet intervalle est donc définie par : taille $= H_c - L_c = 1$.

- Étape 2 :

Cet intervalle est partitionné en N sous-intervalles $[Ls_k, Hs_k [$ en fonction des probabilités de chaque symbole s_k de la source. Ce sont les partitions initiales. La longueur ($Hs_k - Ls_k$) de ces sous intervalles est donnée par : $Ls_k - Hs_k = p_k$, on a donc :

$$Ls_k = L_c + \text{taille} \times \sum_{i=1}^{k-1} p_i \quad \text{et} \quad Hs_k = L_c + \text{taille} \times \sum_{i=1}^{k} p_i \ .$$

- Étape 3 :

On choisit le sous-intervalle correspondant au prochain symbole $s_{\alpha k}$ qui apparaît dans la séquence. On redéfinit alors l'intervalle initial $[L_c, H_c[$: $\begin{cases} L_c = L_c + \text{taille} \times Ls_k \\ H_c = L_c + \text{taille} \times Hs_k \end{cases}$

- Étape 4 :

Cet intervalle est subdivisé à nouveau, selon le même procédé que celui utilisé dans l'étape 2.

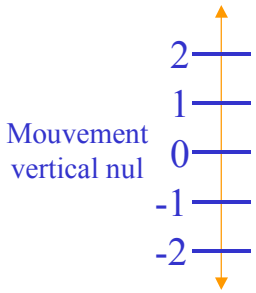
- Étape 5 :

Les étapes 2, 3, et 4 sont répétées jusqu'à obtenir le mot-code représentant la séquence complète des symboles source.

Codage arithmétique

Exemple

On considère la source S= {-2, -1, 0, 1, 2} qui définit une série de 5 vecteurs de mouvements verticaux possibles (vecteurs utilisés en codage vidéo)



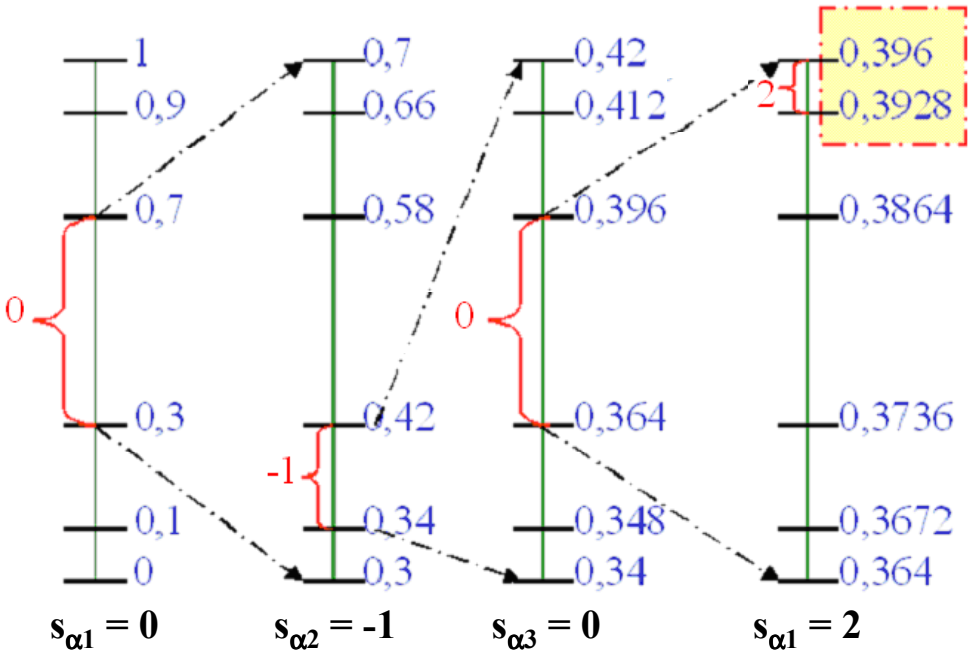
- 'Y' est la variable aléatoire qui désigne la valeur d'un vecteur de mouvement.
- On a les probabilités suivantes:
  - $P\{Y = -2\} = p_1 = 0,1$
  - $P\{Y = -1\} = p_2 = 0,2$
  - $P\{Y = 0\} = p_3 = 0,4$
  - $P\{Y = 1\} = p_4 = 0,2$
  - $P\{Y = 2\} = p_5 = 0,1$

⇒ L'objectif est de coder la séquence de vecteurs de mouvement (0, -1, 0, 2)

Voici un exemple d'application du codage arithmétique dans le cadre de codage des vecteurs de mouvement d'un signal vidéo.

Codage arithmétique

Subdivisions de la partition initiale



Pour coder la séquence $\{s_{\alpha 1}, s_{\alpha 2}, s_{\alpha 3}, s_{\alpha 4}\} = \{s_3, s_2, s_3, s_5\} = \{0, -1, 0, 2\}$, on divise l'intervalle [0, 1[en 5 partitions initiales, puis on se place sur le sous-intervalle correspondant au premier vecteur de mouvement de la séquence (de valeur 0). Pour le prochain symbole de la séquence, le vecteur de mouvement –1, on subdivise la partition initiale du premier vecteur '0' en autant d'intervalles qu'il y'a de vecteurs possibles.
On procède récursivement pour tous les symboles (ici les vecteurs de mouvement) de la séquence. Dès lors, on peut coder la séquence (0, -1, 0, 2) des vecteurs de mouvements verticaux par tout nombre réel appartenant à l'intervalle [0,3928 ; 0,936[. Le nombre 0,3945 code cette séquence. Il requiert 8 bits :

$$0{,}3945 = 0\times2^{-1} + 2^{-2} + 2^{-3} + 0\times2^{-4} + 0\times2^{-5} + 2^{-6} + 0\times2^{-7} + 2^{-8}$$

Nous utilisons donc 8/5 = 1,6 bits/symbole.

Codage arithmétique

Algorithme de décodage de la séquence

1) Initialisation: $L_c = 0$; $H_c = 1$

2) Calculer la longueur du sous-intervalle du code : $H_c - L_c$

3) Trouver le sous-intervalle $[Ls_k, Hs_k[$ du symbole avec $1 \leq k \leq N$

tel que: $Ls_k \leq (\text{mot-code} - L_c) / \text{longueur} < Hs_k$

4) Obtention du symbole: s_k

5) Mettre a jour le sous-intervalle:
$$\begin{cases} L_c = L_c + \text{longueur} \times Ls_k \\ H_c = L_c + \text{longueur} \times Hs_k \end{cases}$$

6) Répéter les étapes 2, 3, 4 et 5 jusqu'à obtenir le décodage de tous les symboles de la séquence

La figure ci-dessus présente l'algorithme de décodage pour un mot-code obtenu après un codage arithmétique. On se propose d'appliquer cet algorithme à l'exemple précédent. On considère le mot-code '$M_c = 0{,}3945$' et on souhaite savoir à quelle séquence il correspond :

[Début Algorithme]

Étape 1 :

On initialise les bornes des sous-intervalles $[L_c, H_c]$: $L_c = 0$ et $H_c = 1$.

Étape 2 :

On calcule la longueur L du sous-intervalle : $L = H_c - L_c = 1$.

Étape 3 :

On calcule le nombre $(M_c - L_c) / L = 0{,}3945$, et on cherche k tel que ce nombre soit compris dans la partition initiale $[Ls_k, Hs_k[$.
Dans le cas présent, il s'agit de l'intervalle [0,3 ; 0,7[qui est aussi la partition initiale $[Ls_3, Hs_3[$.

Étape 4 :

Le ***premier symbole*** $s_{\alpha 1}$ de la séquence est le ***symbole*** s_3 (***vecteur de mouvement 0***).

Étape 5 :

On met à jour le sous-intervalle $[L_c, H_c[$ comme au moment du codage, de manière à décoder le second symbole :

$$L_c = L_c + L \times Ls_3 = 0 + 1 \times 0,3 = 0,3$$
$$H_c = L_c + L \times Hs_3 = 0 + 1 \times 0,7 = 0,7$$

Étape 2 :

On calcule la longueur L du sous-intervalle : $L = H_c - L_c = 0,7 – 0,3 = 0,4$.

Étape 3 :

$(M_c - L_c) / L = (0,3945 – 0,3) / 0,4 = 0,2363$.
Ce nombre est compris dans la partition initiale $[Ls_2, Hs_2[= [0,1 ; 0,3[$.

Étape 4 :

Le ***second symbole*** $s_{\alpha 2}$ de la séquence est le ***symbole*** s_2 (***valeur -1***).

Étape 5 :

$L_c = L_c + L \times Ls_2 = 0,3 + 0,4 \times 0,1 = 0,34$
$H_c = L_c + L \times Hs_2 = 0,3 + 0,4 \times 0,3 = 0,42$
Étape 2 :

On calcule la longueur L du sous-intervalle : $L = H_c - L_c = 0,42 – 0,34 = 0,08$.

Étape 3 :

$(M_c - L_c) / L = (0,3945 – 0,34) / 0,08 = 0,6812$.
Ce nombre est compris dans la partition initiale $[Ls_3, Hs_3[= [0,3 ; 0,7[$.

Étape 4 :

Le ***troisième symbole*** $s_{\alpha 3}$ de la séquence est le ***symbole*** s_3 (***valeur 0***).

Étape 5 :

$L_c = L_c + L \times Ls_3 = 0,34 + 0,08 \times 0,3 = 0,364$
$H_c = L_c + L \times Hs_3 = 0,34 + 0,08 \times 0,7 = 0,396$

Étape 2 :

On calcule la longueur L du sous-intervalle : $L = H_c - L_c = 0,396 – 0,364 = 0,032$.

Étape 3 :

(M_c - L_c) / L = (0,3945 – 0,364) / 0,032 = 0,9531.
Ce nombre est compris dans la partition initiale [Ls_5, Hs_5[= [0,9 ; 1[.

Étape 4 :

Le ***quatrième symbole*** $s_{\alpha 4}$ de la séquence est le ***symbole*** s_5 (***valeur 2***).

[Fin Algorithme]

Le décodage du nombre 0,3945 aboutit à la séquence originale $\{s_{\alpha 1}, s_{\alpha 2}, s_{\alpha 3}, s_{\alpha 4}\} = \{s_3, s_2, s_3, s_5\} = \{ 0, -1, 0, 2 \}$.

Contrairement au codage de Huffman, le codage arithmétique permet de coder sur un nombre de bits non nécessairement entier. Cet avantage permet de réaliser une compression plus performante. Cependant le codage arithmétique est plus long que le codage de Huffman car il n'est pas possible de commencer le décodage avant d'avoir obtenu la séquence entière des symboles envoyés par la source.
Les performances du codage arithmétique peuvent être améliorées dès lors que les tables de probabilités ne sont pas fixées (comme c'est le cas dans l'exemple présenté), mais adaptées à la séquence courante et fonction des séquences précédentes. On parle alors de codage arithmétique adaptatif.

Chapitre 5

CODAGE STATITIQUE

Présentation

Classification des codes statistiques

Codage de l 'information

- ***Objectifs***
  - transcription des informations pour faciliter le codage
    code $\Rightarrow$ signal *(Transcodage)*
  - Compression d 'information
    réduction de la quantité d 'information
  - Protection contre les erreurs de transmission
    contre les dégradations et les erreurs de décision
  - Assurer le secret des informations transmises
    Chiffrement
- **Définition d 'un code**
  application de S dans $\mathcal{A} = \{ a_1, a_2, \ldots\ldots, a_q \}$
  message $m_i \in S \quad \Rightarrow \quad$ mot-code $M_i \in \mathcal{M}$ suites finies de $\mathcal{A}$

Le codage des informations consiste à transcrire les messages d'une source d'information sous la forme d'une suite de caractères pris dans un alphabet prédéfini. Les objectifs du codage sont principalement de quatre ordres :

- de ***transcrire*** les informations sous une forme permettant d'élaborer assez facilement le signal qui supportera les informations, ou de manipuler aisément ces informations automatiquement. Pour cela, différents codes de représentation de l'information sont utilisés en fonction des différentes applications envisagées et on utilise assez souvent des opérations de transcodage ;

- de ***réduire*** la quantité des symboles d'information (en terme de nombre total de symboles utilisés) nécessaires à la représentation des informations : c'est un rôle économique ;

- de ***lutter*** contre les dégradations (distorsions, bruit) amenées par le canal de transmission et conduisant à des erreurs de reconstruction de l'information en sortie du canal de transmission (en réception) ;

- d'assurer un secret dans la transmission des informations de façon à rendre l'information inintelligible sauf au destinataire de celle-ci.

- *Définition du code :*

Soit un ensemble, nommé alphabet A formé de q caractères a_i : $\mathcal{A}$ = { a_1, a_2, …, a_q }et $\mathcal{M}$ l'ensemble fini des suites finies M_i de caractères (par exemple : $M_i = a_{10}\ a_4\ a_7$).
Soit un ensemble fini de messages provenant d'une source S de messages : S={ m_1, …, m_N }.

On appelle ***code*** toute application de S dans $\mathcal{A}$: codage de S par l'alphabet A.

L'élément M_i de $\mathcal{M}$ qui correspond au message m_i de S est appelé le ***mot-code*** de m_i. Sa ***longueur***, notée n_i, est le nombre de caractères appartenant à $\mathcal{A}$ qui le forment.

Le ***décodage*** d'une suite de messages m_i envoyés implique qu'on soit en mesure de séparer les mots-codes dans la suite reçue de mots-codes M_i d'où, parfois, l'utilisation dans l'alphabet d'un caractère particulier d'espacement.

Codage de l 'information

- Alphabet A = { a_1, a_2, ……, a_q }
- Ensemble fini de messages S = { m_1 , m_2 , …., m_i ,…… , m_N }

Codage ↓

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

- Longueur des mots-codes : $n_i = n(M_i)$
- Longueur moyenne des mots-codes : $E(n) = \sum_{i=1;N} p_i n_i$
- Entropie de la source H : $H(p_1, \dots, p_N) \leq \log_2 N$
- Quantité moyenne d 'information par caractère = H / E(n)

  or $H / E(n) \leq \log_2 q \Rightarrow E(n) \geq H / \log_2 q$
- Débit d 'une source d'informations codée en moyenne avec D caractères par seconde : R = D H/E(n)

  => $\mathbf{R \leq D \log_2 q}$ *R en bit/seconde*

On appelle dorénavant $\mathbf{m_i}$ les **messages** produits par la source d'information et $\mathbf{M_i}$ les **mots-codes** associés à ceux-ci.
On appelle $\mathbf{n_i} = n(M_i)$ le nombre de caractères appartenant à **l'alphabet** $\mathcal{A}$ (Card($\mathcal{A}$) = q) nécessaires au codage de m_i, n_i est donc la **longueur** du mot-code M_i. Si la source utilise N messages différents possibles, la longueur moyenne des mots-codes est donnée par :

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

H est l'incertitude moyenne (i.e. l'entropie) de la source S **par message** envoyé, donc l'incertitude moyenne (i.e. l'entropie) **par caractère** (de l'alphabet $\mathcal{A}$) est égale à $\frac{H}{E(n)}$ et telle que : $\frac{H}{E(n)} \leq \log_2 q$ (car on a q caractères dans l'alphabet $\mathcal{A}$), donc : $E(n) \geq \frac{H}{\log_2 q}$.
Enfin, si la source codée d'information produit D caractères par seconde pris dans l'alphabet $\mathcal{A}$, $\frac{H}{E(n)}$ étant l'information moyenne transportée par caractère en bit/caractère, le débit R d'information est : $R = D.\frac{H}{E(n)}$. Ce débit est donc limité par : $R \leq D.\log_2 q$.

Codage et décodage de l 'information (5)

- ***Efficacité*** η d 'un code : $\eta = n_{min} / E(n)$ => $\eta = H / (E(n) \log_2 q)$
- ***Redondance*** ρ d 'un code : $\rho = 1 - \eta$
- *Exemples simples: codes C_1 et C_2*
- **Contraintes** : séparation des mots-codes & lecture non-ambiguë des mots-codes => *codes réguliers et déchiffrables*
- ***Code régulier*** : si $m_i \neq m_j \Longrightarrow M_i \neq M_j$ *(application injective)*
- ***Code déchiffrable*** : 2 suites de messages distincts
  ==> 2 suites de codes distincts
  si $(m_{\alpha 1},\ldots, m_{\alpha i}) \neq (m_{\beta 1},\ldots, m_{\beta j})$ => $(M_{\alpha 1},\ldots, M_{\alpha i}) \neq (M_{\beta 1},\ldots, M_{\beta j})$
  exemples: codes à longueur fixe; codes avec séparateur

- ***Code irréductible***: code déchiffrable pouvant être décodé sans artifice
  M_i n 'est pas préfixe de M_j $\forall$ i , j

Quelques définitions et propriétés liées au codage et au décodage de l'information :

Efficacité :

Pour un alphabet A donné, on appelle efficacité d'un code le rapport η donné par :

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

Redondance :

On définit la redondance mathématique par le facteur $\rho = 1 - \eta$. La redondance peut être utilisée pour accroître la robustesse du codage face aux erreurs de transmission de l'information codée (détection et correction des erreurs).

Voici un exemple simple : on considère une source à 4 messages possibles $\{m_1, m_2, m_3, m_4\}$ de probabilités respectives : $p_1 = 0.5$; $p_2 = 0.25$; $p_3 = p_4 = 0.125$.
Soient les deux codes C_1 (codage binaire simple) et C_2 suivants (code à longueur variable) :

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

Pour C_1 : $\eta = \frac{1.75}{2} = 0.875$ et $\rho = 1 - \eta = 0.125$.

Pour C_2 : $\eta = \frac{1.75}{1.75} = 1$ et $\rho = 1 - \eta = 0$.

Le code C_2 est d'efficacité maximale (unité) alors que le code C_1 ne l'est pas.

Code régulier :

Tout mot-code donné n'est associé qu'à un seul message possible (l'application $S \rightarrow A$ est bijective) : si $m_i \neq m_j$ alors $M_i \neq M_j$.

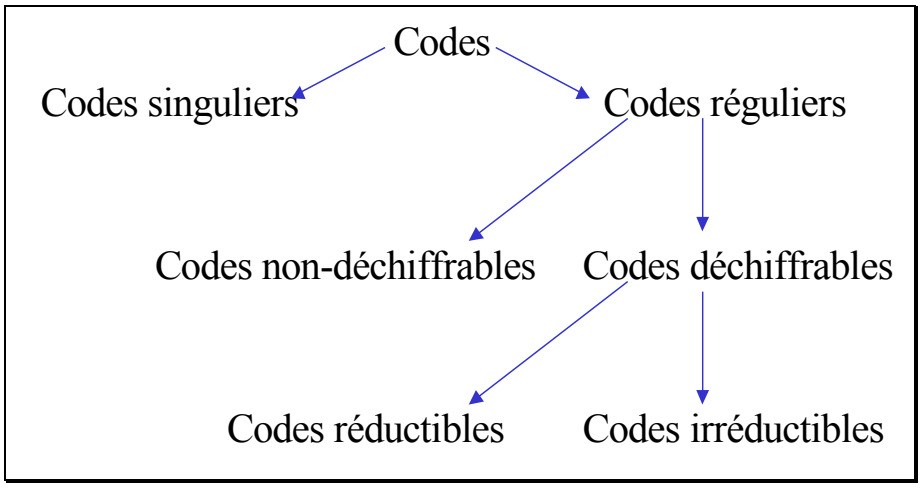
Code déchiffrable :

Un code est déchiffrable si deux textes distincts $(m_{\alpha 1}, \ldots, m_{\alpha i})$ et $(m_{\beta 1}, \ldots, m_{\beta j})$ conduisent nécessairement à des codages distincts (par exemple les codes à longueur fixe tels que C_1 et les codes avec séparateur). Un code déchiffrable est donc un cas particulier de code régulier.

Code irréductible :

C'est un code déchiffrable pouvant être lu directement sans artifices particuliers (code de longueur fixe, séparateur). Pour cela, tout mot-code M_i d'un message m_i ne doit avoir aucun préfixe qui soit un autre mot-code M_j.

On peut ainsi créer une classification hiérarchique pour caractériser le type d'un code :



Exemples de codes

Codes réguliers / Codes déchiffrables / Codes irréductibles

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 est non régulier
- C_2 est non-déchiffrable
- C_3 est déchiffrable et irréductible
- C_4 est seulement déchiffrable

Voici quatre codes C_1, C_2, C_3 et C_4 donnés en exemples des définitions et des propriétés précédentes. On suppose que les quatre messages m_1, m_2, m_3, et m_4 sont distincts.

Le code C_1 est non régulier : $m_1 \neq m_2$ mais $C_1(m_1) = C_1(m_2)$, et de même $C_1(m_3) = C_1(m_4)$.

Le code C_2 est non-déchiffrable : les deux textes $\{m_1, m_2\}$ et $\{m_4\}$ sont différents, mais ils conduisent au même code « 01 ».

Le code C_3 est déchiffrable et irréductible : deux textes distincts composés de suites de messages, par exemple $\{m_1, m_3, m_4\}$ et $\{m_1, m_2\}$, conduisent toujours à des codes différents, et aucun mot-code $M_i = C_3(m_i)$ n'est préfixe d'un autre mot-code $M_j = C_3(m_j)$

Le code C_4 est déchiffrable mais non-irréductible : deux textes distincts conduisent toujours à des codes différents, mais les mots-codes $M_i = C_4(m_i)$ sont les préfixes de tous les mots-codes $M_j = C_4(m_j)$ dès que $i < j$.