Université Libre de Bruxelles
Faculté des Sciences Appliquées

Professeur : Nicolas Cerf
Edition 2006-2007

Théorie de l'Information, du Codage et des Télécommunications

ELEC377

Notes de cours par Jonathan Verlant-Chenet

Table des matières

Chapitre 1

Introduction et définitions

1.1 Introduction et rappels

1.1.1 Introduction génerale

Le cours de théorie de l'information a pour but détudier la transmission de données (le plus souvent, sous forme de bits) d'une source à un utilisateur récepteur par le biais d'un canal bruité (voir figure 1.1). Par canal bruité, il faut comprendre qu'aucun dispositif de transmission (ligne de transmission, ondes, ...) n'est parfait et qu'il peut donc inverser certains bits transmis et fausser l'information interceptée par l'utilisateur.
Pour cette raison, il existe un codage de l'information de la source avant l'émission dans le canal, pour comprimer les données (voir chapitre 3 page 37) et ajouter des redondances permettant de lutter contre le bruit (voir chapitre 5 page 63). Une fois la transmission effectuée via le canal, un décodeur est chargé de retrouver quels étaient les messages envoyés par la source, pour finalement les transmettre à l'utilisateur.

La théorie de l'information est une théorie mathématique fortement statistique, qui a été créée par Shannon. L'ensemble de ce chapitre a donc pour but d'introduire les entitées mathématiques qui interviendront dans cette théorie.

1.1.2 Rappels de probabilités

Les sources transmettant de l'information constituent une suite d'évènements aléatoires, c'est-à-dire une suite d'expériences dont on ne connaît à priori pas le résultat. Ceci nous permet donc

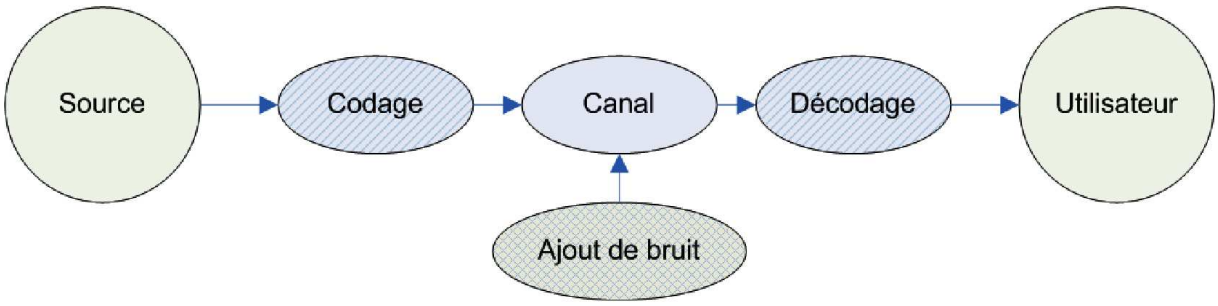

Fig. 1.1 – Cheminement de l'information

d'introduire la notion de probabilité, vue dans le cours de Probabilités et Statistique de seconde année de Bachelier. La probabilité de l'évènement x est notée $p(x)$. On utilisera également la notion de probabilité conditionnelle :

$$p(x|y) = \frac{p(x,y)}{p(y)} \tag{1.1}$$

La formule $p(x|y)$ se lisant "la probabilité de x, sachant y", avec $p(x,y)$ étant la probabilité de l'intersection des ensembles x et y. Si ces évenements sont indépendants, alors la probabilité $p(x,y)$ est simplement donnée par :

$$p(x,y) = p(x)p(y) \tag{1.2}$$

impliquant que :

$$p(x|y) = \frac{p(x,y)}{p(y)} = \frac{p(x)p(y)}{p(y)} = p(x) \tag{1.3}$$

Ce qui signifie bien que la probabilité de x n'est pas influencée par l'évènement y.
Gardez bien en mémoire ces trois formules 1.1, 1.2 et 1.3, elles seront utilisées dans l'ensemble du cours et souvent aux séances d'exercices.

On utilisera également la notion de variable aléatoire X (discrète dans ce cours), qui est une application qui envoit l'ensemble des résultats d'une expérience vers un réel. Par exemple, soit X la variable aléatoire liée à un jet de dés : les valeurs qu'elle peut prendre sont les éléments de l'ensemble $\{1,2,3,4,5,6\}$, et on définira par exemple la probabilité que X valle 3 comme $p(X=3) = p(3) = \frac{1}{6}$

1.2 L'entropie de Shannon

1.2.1 Définition

L'entropie d'une variable aléatoire X d'alphabet $\mathcal{X} = \{x\}$ (où les x représentent les valeurs possibles de X) est mathématiquement définie par :

$$\boxed{H(X) = -\sum_{x \in \mathcal{X}} p\left(x\right) \log_2 p(x)} \tag{1.4}$$

Cette entropie peut être interprêtée de différentes façons. Elle peut être vue comme :

- L'incertitude sur X

- L'information nécessaire pour décrire X, c'est-à-dire pour lever l'incertitude
- Le nombre de questions binaires (oui/non) nécessaires pour caractériser X
- La longueur moyenne de la description la plus courte de X

Note complémentaire

La définition mathématique de l'entropie n'a pas été prise au hasard. Elle découle d'une autre notion importante : l'information propre d'un symbole x, vue aussi comme l'incertitude sur le symbole x. Celle-ci est définie par $I(x) = -\log_2 p(x)$. En effet, on considère que l'apparition d'un évènement peu probable apporte beaucoup d'information, alors qu'un évènement certain ne fournit aucune information. Par exemple, si une source émet un symbole x avec une probabilité $p(x) = 1$, alors l'information propre de ce symbole vaudra $I(x) = -\log_2 p(x) = 0$, ce qui est logique : si on sait pertinemment que la source va nous fournir le symbole x, alors il n'existe aucune incertitude sur celui-ci, et son information propre est nulle.
De ce fait, la fonction logarithme se prête parfaitement à la définition : plus la probabilité de x s'approche de 1, plus l'information diminue. Le signe "-" est simplement dû au fait que $p(x)$ est par définition compris entre 0 et 1 et que le logarithme d'un tel nombre est négatif (il serait en effet illogique de parler d'une information négative).
La base du logarithme a aussi son utilité. Dans la définition de l'entropie, on pose la base 2 car on considèrea dans la suite du cours que nous travaillons avec des sources binaires (qui utilisent les bits). Mais nous pourions très bien définir une information propre pour d'autres bases de travail, en utilisant une autre base pour le logarithme.

La définition de l'entropie est donc une valeur moyenne de l'information propre de tous les symboles existants dans l'alphabet $\mathcal{X}$ de la source.

1.2.2 Propriétés

- Si la distribution $p\,(x) = \frac{1}{|\mathcal{X}|}$ est uniforme telle que $|\mathcal{X}| = n$ est la taille de l'alphabet $\mathcal{X} = \{\, x_1 ... x_n \,\}$, alors :

$$H\,(X) = -\sum_{x=1}^{n} \frac{1}{n} \log_2 \frac{1}{n} = \log_2 n \tag{1.5}$$

Il s'agit de la manière la plus désordonnée de répartir les probabilités.

Exemple

On jette une pièce équilibrée avec une probabilité uniforme $p(x) = P(X = x) = \frac{1}{2}\ \forall x$ avec

$$\mathcal{X} = \{pile, face\}$$

. L'entropie d'une telle source est donnée par :

$$H(X) = -p(face)\log_2 p(face) - p(pile)\log_2 p(pile) = 1\ bit$$

- Rajouter un élément de probabilité nulle ne modifie pas l'entropie.

Exemple

Dans l'exemple du lancé de pièce ci-dessus, ajoutons l'évènement de probabilité nulle suivant : "la pièce tombe sur la tranche". Sa probabilité tendant vers 0, on a $p(tranche) = \epsilon \to 0$, qui aura, dans l'entropie, la contribution suivante : $\epsilon \log_2 \epsilon$ avec

$$\lim_{\epsilon \to 0} \epsilon \log_2 \epsilon = \lim_{\epsilon \to 0} \frac{\log_2 \epsilon}{1/\epsilon}$$

Vu l'indétermination, on applique le théorème de l'Hospital, et on dérive les termes du numérateur et du dénominateur, pour obtenir finalement :

$$\lim_{\epsilon \to 0} -\frac{1/\epsilon}{1/\epsilon^2\ \ln 2} = \lim_{\epsilon \to 0} -\epsilon = -0$$

On a donc finalement l'entropie suivante :

$$H(X) = -p(face)\log_2 p(face) - p(pile)\log_2 p(pile) - p(tranche)\log_2 p(tranche)$$
$$= 1 - \lim_{\epsilon \to 0} \epsilon \log_2 \epsilon = 1 + 0 = 1\ bit$$

- L'entropie est positive :

$$\boxed{H(X) \geq 0} \tag{1.6}$$

En effet, une probabilité étant comprise entre 0 et 1, le logarithme $\log_2 p(x)$ est négatif, si bien que $-\log_2 p(x)$ est positif. Ainsi, on obtient :

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) log_2 p(x) = \sum_{x \in \mathcal{X}} p(x)[-log_2 p(x)]$$

$p(x)$ et $-log_2 p(x)$ étant définis comme positifs, la somme sera bien positive.

- L'entropie peut être définie comme l'espérance mathématique de la fonction $-log_2 p(x)$, c'est-à-dire :

$$H(X) = E[-log_2 p(x)]_{X \tilde{} p(x)}$$

où $X \tilde{} p(x)$ signifie que X est distribué comme $p(x)$.

- L'entropie ne dépend, par définition, que de la probabilité $p(x)$ et non de x.

1.3 Axiomes de Shannon

La théorie de Shannon s'appuie sur trois axiomes fondamentaux :

- L'entropie $H_n(p_1, p_2, ..., p_n)$ définie pour les n probabilités $\{p_1, p_2, ..., p_n\}$ est continue en ces probabilités.
- Si toutes les probabilités $\{p_1, p_2, ..., p_n\}$ sont réparties uniformément, alors H_n est monotone croissante de n.
- Axiome de regroupement : si X résulte de choix successifs, alors l'entropie H est la moyenne des entropies individuelles.

Démonstration

Cette démonstration s'effectue sur un alphabet de 4 éléments mais peut se généraliser à tout alphabet. Soit les différents choix successifs représentés schématiquement ci-dessous.

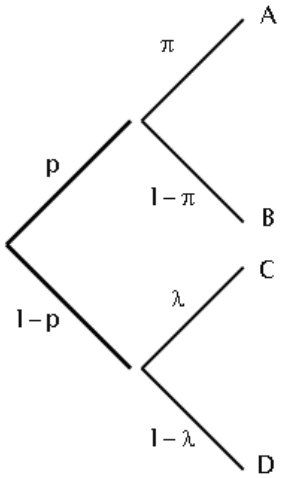

FIG. 1.2 – Succession de choix amenant aux éléments {A,B,C,D} de l'alphabet

Il y a quatre chemins possibles, dont les probabilités sont données par :
– Symbole A : probabilité p puis π, ce qui donne $p(A) = p\pi$
– Symbole B : probabilité p puis $1 - \pi$, ce qui donne $p(B) = p(1 - \pi)$
– Symbole C : probabilité $1 - p$ puis λ, ce qui donne $p(C) = (1 - p)\lambda$
– Symbole D : probabilité $1 - p$ puis $1 - \lambda$, ce qui donne $p(D) = (1 - p)(1 - \lambda)$
L'entropie de cette source est donc donnée par :

$$H_4[p\pi, p(1-\pi), (1-p)\lambda, (1-p)(1-\lambda)]$$
$$= -p\pi \log_2 p\pi - p(1-\pi) \log_2 p(1-\pi) - (1-p)\lambda \log_2 (1-p)\lambda - (1-p)(1-\lambda) \log_2 (1-p)(1-\lambda)$$

Utilisons ici la propriété des logarithmes suivante :

$$\log a.b = \log a + \log b \tag{1.7}$$

On obtient :

$$
\begin{aligned}
H_4 &= -p\pi[\log_2 p + \log_2 \pi] - p(1-\pi)[\log_2 p + \log_2 (1-\pi)] - (1-p)\lambda[\log_2 (1-p) + \log_2 \lambda] \\
&\quad -(1-p)(1-\lambda)[\log_2 (1-p)) + \log_2 (1-\lambda)] \\
&= -p\pi \log_2 p - p\pi \log_2 \pi - p(1-\pi) \log_2 p - p(1-\pi) \log_2 (1-\pi) - (1-p)\lambda \log_2 (1-p) - (1-p)\lambda \log_2 \\
&\quad -(1-p)(1-\lambda) \log_2 (1-p)) - (1-p)(1-\lambda) \log_2 (1-\lambda) \\
&= -p\pi \log_2 p - p\pi \log_2 \pi - p \log_2 p + p\pi \log_2 p - p \log_2 (1-\pi) + p\pi) \log_2 (1-\pi) - \lambda \log_2 (1-p) \\
&\quad +p\lambda \log_2 (1-p) - \lambda \log_2 \lambda + p\lambda \log_2 \lambda - \log_2 (1-p)) + \lambda \log_2 (1-p)) + p \log_2 (1-p)) - p\lambda \log_2 (1 \\
&\quad - \log_2 (1-\lambda) + \lambda \log_2 (1-\lambda) + p \log_2 (1-\lambda) - p\lambda \log_2 (1-\lambda) \\
&= -p[\pi \log_2 \pi + \log_2 (1-\pi) - \pi \log_2 (1-\pi)] - p \log_2 p - \lambda \log_2 \lambda + p\lambda \log_2 \lambda - \log_2 (1-p) + p \log_2 (1 \\
&\quad - \log_2 (1-\lambda) + p \log_2 (1-\lambda) + \lambda \log_2 (1-\lambda) - p\lambda \log_2 (1-\lambda) \\
&= -p \log_2 p - (1-p) \log_2 (1-p) + p[-\pi \log_2 \pi - (1-\pi) \log_2 (1-\pi)] + (1-p)[-\lambda \log_2 \lambda - (1-\lambda) \log_
\end{aligned}
$$

Par définition de l'entropie, on obtient donc finalement :

$$H_4[p\pi, p(1-\pi), (1-p)\lambda, (1-p)(1-\lambda)] = H_2(p, 1-p) + pH_2(\pi, 1-\pi) + (1-p)H_2(\lambda, 1-\lambda) \quad (1.8)$$

1.4 Concavité de l'entropie

L'entropie a pour propriété que

$$H_2[\lambda p_1 + (1-\lambda)p_2] \geq \lambda H_2(p_1) + (1-\lambda)H_2(p_2) \quad (1.9)$$

Exemple

Montrons un exemple pour $\lambda = \frac{1}{2}$. Soit deux canaux dans lesquels on intercepte les sorties x_1 et x_2. Les probabilités de chaque sortie sont données par

$$x_1 = \begin{cases} 0 \text{ avec probabilité } p_1 \\ 1 \text{ avec probabilité } 1-p_1 \end{cases}$$

et

$$x_2 = \begin{cases} 0 \text{ avec probabilité } p_2 \\ 1 \text{ avec probabilité } 1-p_2 \end{cases}$$

On fait un mélange 50/50 des résultats des deux canaux et on obtient le résultat x :

$$x = \begin{cases} 0 \text{ avec probabilité } \overline{p} = \frac{1}{2}p_1 + \frac{1}{2}p_2 \\ 1 \text{ avec probabilité } 1-\overline{p} = \frac{1}{2}(1-p_1) + \frac{1}{2}(1-p_2) \end{cases}$$

où $\overline{p} = \frac{p_1+p_2}{2}$ est la probabilité de mélange
Ainsi, on a bien $H_2(\overline{p}) \geq \frac{1}{2}H(p_1) + \frac{1}{2}H(p_2)$, ce qui se voit sur le schéma ci-dessous.

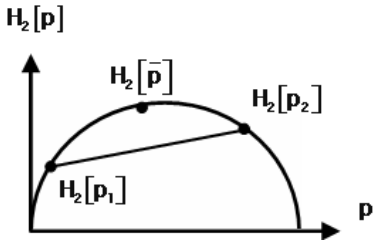

FIG. 1.3 – Illustration de la concavité de l'entropie

1.5 Entropies de mélange (ou des systèmes bipartites)

Différentes entropies peuvent être définies quand on mélange deux informations. On les met en évidence par le diagramme de Venn entropique ci-dessous.

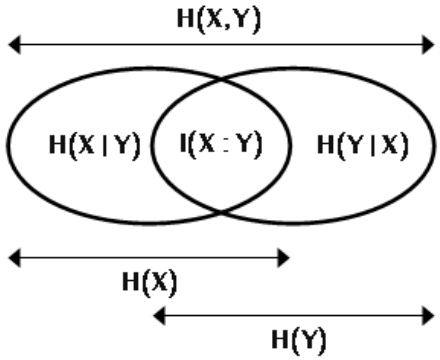

FIG. 1.4 – Diagramme de Venn pour un système bipartite

1.5.1 Entropie jointe

Pour une paire (X,Y) distribuées comme $p(x, y)$, on définit l'entropie jointe comme étant :

$$\boxed{H(X,Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log_2 p(x,y) = H(Y,X)} \tag{1.10}$$

Elle est considérée, dans le diagramme de Venn ci-dessus, comme l'union des deux ensembles entropiques, l'intersection des deux ensemble n'étant comptée qu'une seule fois.

1.5.2 Entropie conditionnelle

Pour une paire (X,Y) distribuée comme $p(x, y)$, l'entropie conditionnelle de X sachant Y est définie comme la somme des entropies conditionnelles à chaque x pris séparément, et pondérées par les probabilités de chaque x :

$$H(Y|X) = \sum_{x\in\mathcal{X}} p(x)H(Y|X = x)$$

Afin de développer cette formule, utilisons ici la définition de l'entropie (1.4) en y insérant des les probabilités conditionelles p(y|x) :

$$H(Y|X=x) = -\sum_{y\in\mathcal{Y}} p(y|x)\log_2 p(y|x)$$

En insérant cette définition dans la formule de H(Y|X) ci-dessus, on obtient :

$$H(Y|X) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x)p(y|x)\log_2 p(y|x)$$

Par la propriété (1.1) des probabilités, on a que $p(x)p(y|x) = p(x,y)$, ce qui donne au final :

$$\boxed{H(Y|X) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log_2 p(y|x)} \tag{1.11}$$

Il s'agit donc de l'incertitude sur Y sachant X. Par la définition de l'espérance mathématique, on peut aussi noter que $H(Y|X) = E[-\log_2 p(x|y)]_{(X,Y)(x,y)}$

1.5.3 Règle de chaîne

La règle de chaîne pour l'entropie des systèmes bipartites définit un lien entre l'entropie jointe et l'entropie conditionnelle. Elle se retrouve très facilement à partir du diagramme de Venn. Celle-ci stipule en effet que l'entropie jointe de X et Y est donnée par la somme de l'entropie de X et de l'entropie de Y sachant X :

$$\boxed{H(X,Y) = H(X) + H(Y|X)} \tag{1.12}$$

Par la symétrie du diagramme de Venn, on peut aussi dire que cette entropie jointe est la somme de l'entropie de Y et de l'entropie de X sachant Y :

$$H(X,Y) = H(Y) + H(X|Y)$$

Voyons comment démontrer cette propriété plus rigoureusement.

Démonstration

Reprenons la définition de l'entropie jointe (1.10) :

$$H(X,Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log_2 p(x,y)$$

Par la propriété (1.1) des probabilités, on sait que $p(x,y) = p(x)p(y|x)$. Ainsi :

$$H(X,Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x)p(y|x)\log_2 p(x)p(y|x)$$

Par la propriété des logarithmes (1.7), on décompose l'équation ci-dessus en deux :

$$H(X,Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}[p(x)p(y|x)\log_2 p(x) + p(x)p(y|x)\log_2 p(y|x)]$$
$$= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x)p(y|x)\log_2 p(x) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x)p(y|x)\log_2 p(y|x)$$

Pour le premier terme, on va réorganiser la double somme étant donné que la somme sur les y ne dépend que de p(y|x). Pour le deuxième terme, on réutilise la propriété (1.1) des probabilités pour retrouver la définition de l'entropie conditionnelle H(Y|X). On obtient ainsi :

$$H(X,Y) = -\sum_{x\in\mathcal{X}} p(x)\log_2 p(x)\sum_{y\in\mathcal{Y}} p(y|x) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log_2 p(y|x)$$

Par définition d'une probabilité, on a que

$$\sum_{y\in\mathcal{Y}} p(y|x) = 1$$

ce qui donne finalement :

$$H(X,Y) = H(X) + H(Y|X)$$

en utilisant les définitions (1.4) et (1.11).

D'autres propriétés découlent de cette règle de chaîne :

- Si X et Y sont indépendantes,

$$H(Y|X) = H(Y) \tag{1.13}$$

ce qui implique, par la règle de chaîne, que :

$$H(X,Y) = H(X) + H(Y|X) = H(X) + H(Y)$$

En effet, si X et Y sont indépendantes, on sait d'après les formules (1.2) et (1.3)

$$p(y|x) = p(y) \tag{1.14}$$

et

$$p(x,y) = p(x)p(y) \tag{1.15}$$

Reprenons la définition (1.11) de l'entropie conditionnelle :

$$H(Y|X) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log_2 p(y|x)$$

Par les propriétés (1.14) et (1.15), la définition devient :

$$H(Y|X) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x)p(y)\log_2 p(y)$$

$$= -\sum_{y\in\mathcal{Y}} p(y)\log_2 p(y)\sum_{x\in\mathcal{X}} p(x)$$

En utilisant le fait que

$$\sum_{x\in\mathcal{X}} p(x) = 1$$

et la définition de l'entropie (1.4), on obtient finalement $H(Y|X) = H(Y)$. Ce résultat est logique : si on connait X, l'incertitude sur Y en connaissant X est la même que l'incertitude sur Y.

- Si X et Y sont totalement corrélées, Y est fonction de X. En effet, si X et Y sont corrélées, on a par définition que

$$p(y|x) = \begin{cases} 1 \text{ si } y = y* = f(x) \\ 0 \text{ si } y \neq y* \end{cases}$$

On a alors :

$$H(Y|X) = \sum_{x\in\mathcal{X}} p(x)H(Y|X=x) = -\sum_{x\in\mathcal{X}} p(x) \sum_{y\in\mathcal{Y}} p(y|x)\log_2 p(y|x)$$

Par la définition de p(y|x) ci-dessus, il ne reste plus que le terme où $y = y*$, c'est-à-dire :

$$H(Y|X) = -\sum_{x\in\mathcal{X}} p(x)[p(y*|x)\log_2 p(y*|x) + 0]$$

Etant donné que $p(y*|x) = 1$, par définition, le logarithme s'annulle et la somme fait donc 0. On a donc $H(Y|X) = 0$, ce qui est logique. En effet, si X et Y sont corrélés, le fait de connaitre X implique qu'on connait Y et donc l'incertitude sur Y est nulle. Ainsi, $H(Y|X) = 0$ si $Y = f(X)$ et inversemment.

1.5.4 Entropie mutuelle

- Définition intuitive
  L'entropie mutuelle, ou information mutuelle, est définie par :

$$\boxed{H(X:Y) = I(X:Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = I(Y:X)} \tag{1.16}$$

On l'interprète comme étant l'information partagée par X et Y, ou bien la réduction de l'incertitude sur X (ou Y) amenée par la connaissance de Y (ou X).

Démonstration

La définition intuitive de l'entropie mutuelle suggère qu'elle est symétrique : $I(X,Y) = I(Y,X)$ Voyons comment montrer cela. Par définition, on a :

$$I(X:Y) = H(X) - H(X|Y)$$

Par la règle de chaîne (1.12), on transforme H(X|Y) :

$$I(X:Y) = H(X) - [H(X,Y) - H(Y)] = H(X) + H(Y) - H(X,Y)$$

Par définition, l'entropie jointe H(X,Y) est symétrique, et donc :

$$I(X:Y) = H(X) + H(Y) - H(Y,X) = H(Y) - [H(Y,X) - H(X)]$$

On réutilise ici la règle de chaîne :

$$I(X:Y) = H(Y) - H(Y|X) = I(Y:X)$$

- Définition générale

La définition générale de l'entropie mutuelle, en termes de logarithmes, est donnée par :

$$\boxed{H(X:Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log_2\frac{p(x)p(y)}{p(x,y)}} \tag{1.17}$$

où

$$p(x) = \sum_{y\in\mathcal{Y}} p(x,y) \tag{1.18}$$

car, étant donné que

$$\sum_{y\in\mathcal{Y}} p(y|x) = 1$$

on a :

$$p(x) = p(x)\sum_{y\in\mathcal{Y}} p(y|x) = \sum_{y\in\mathcal{Y}} p(x)p(y|x) = \sum_{y\in\mathcal{Y}} p(x,y)$$

par la propriété (1.1) des probabilités. De même, on définit p(y) par un calcul similaire amenant à :

$$p(y) = \sum_{x\in\mathcal{X}} p(x,y)$$

Par la définition de l'espérance mathématique, on peut aussi dire que l'entropie mutuelle est équivalente à :

$$H(X:Y) = E\left[-\log_2\frac{p(x)p(y)}{p(x,y)}]_{(X,Y)\widetilde{}p(x,y)}\right]$$

Démonstration

Reprenons la définition intuitive (1.16) de l'entropie mutuelle afin de montrer comment arriver à la définition génerale (1.17) :

$$H(X:Y) = H(X) - H(X|Y) = -\sum_{x\in\mathcal{X}} p(x)\log_2 p(x) + \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log_2 p(x|y)$$

Utilisons ici la définition (1.18) et la propriété (1.1) :

$$H(X:Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log_2 p(x) + \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log_2\frac{p(x,y)}{p(y)}$$

$$= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)[\log_2 p(x) - \log_2\frac{p(x,y)}{p(y)}]$$

Utilisons maintenant la propriété des logarithmes (1.7) :

$$H(X:Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log_2\frac{p(x)p(y)}{p(x,y)}$$

- Propriétés
  - L'entropie est séparable en 2 : $H(X) = H(X|Y) + H(X:Y)$ (de même pour H(Y))
  - L'entropie mutuelle est positive : $I(X:Y) \geq 0$ (sera démontré plus loin)
  - L'entropie est la "self-information" de X. En effet, $I(X:X) = H(X) - H(X|X) = H(X)$, puisque $H(X|X)$ est nulle par définition.

1.5.5 Entropie relative

L'entropie relative, ou "distance" de Kullback, est définie par :

$$\boxed{D(p||q) = \sum_{x\in\mathcal{X}} p(x)\log_2\frac{p(x)}{q(x)}} \tag{1.19}$$

En utilisant la définition de l'espérance mathématique, on peut écrire l'entropie relative comme étant : $D(p||q) = E[\log_2\frac{p(x)}{q(x)}]$ On interprète cette distance de Kullback comme ceci. Si on croit que X est distribué selon q alors qu'elle l'est selon p, et qu'on passe cette faute dans un canal, on ne pourra pas coder X avec le même nombre de bits et on sera pénalisé par une relation proportionnelle à $D(p||q)$, qui est donc une mesure de l'inefficacité à utiliser q au lieu de p.
L'entropie relative a les propriétés d'une distance : $D(p||q) \geq 0$ et $D(p||q) = 0$ si $p = q$ (ces propriétés seront démontrées plus loin) mais elle n'en est pas une : elle n'est pas symétrique et ne satisfait pas l'inégalité triangulaire.

1.5.6 Entropie relative conditionnelle

L'entropie relative conditionnelle est définie par :

$$\boxed{D[p(y|x)||q(y|x)] = \sum_{x\in\mathcal{X}} p(x)\sum_{y\in\mathcal{Y}} p(y|x)\log_2\frac{p(y|x)}{q(y|x)} = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log_2\frac{p(y|x)}{q(y|x)}} \tag{1.20}$$

Il existe une règle de chaîne qui relie l'entropie relative à l'entropie relative conditionnelle. Elle s'écrit :

$$\boxed{D[p(x,y)||q(x,y)] = D[p(x)||q(x)] + D[p(y|x)||q(y|x)]} \tag{1.21}$$

Démonstration

Utilisons la définition de l'entropie relative (1.19) et appliquons-là aux distributions p(x,y) et q(x,y) :

$$D[p(x,y)||q(x,y)] = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log_2\frac{p(x,y)}{q(x,y)}$$

Par la propriété (1.1) des probabilités, $p(x,y) = p(x)p(y|x)$, et donc :

$$D[p(x,y)||q(x,y)] = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log_2\frac{p(x)p(y|x)}{q(x)q(y|x)}$$

Servons-nous ensuite de la propriété (1.7) des logarithmes pour séparer la somme en deux :

$$D[p(x,y)||q(x,y)] = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x)p(y|x)\log_2\frac{p(x)}{q(x)} + \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log_2\frac{p(y|x)}{q(y|x)}$$

Remarquons que le deuxième terme n'est rien d'autre que la définition de l'entropie relative conditionnelle (1.20). Dans le premier terme, la somme sur les y ne s'applique que sur l'élément p(y|x) et nous savons que $\sum_{y\in\mathcal{Y}} p(y|x) = 1$. Ainsi :

$$D[p(x,y)||q(x,y)] = \sum_{x\in\mathcal{X}} p(x)\log_2\frac{p(x)}{q(x)}\sum_{y\in\mathcal{Y}} p(y|x) + D[p(y|x)||q(y|x)]$$

Par la définition de l'entropie relative (1.19), on obtient finalement :

$$D[p(x,y)||q(x,y)] = D[p(x)||q(x)] + D[p(y|x)||q(y|x)]$$

1.6 Inégalités fondamentales

1.6.1 Inégalités de Jensen

Si f est convexe ($f'' \geq 0$), alors $E[f(x)] \geq f(E[x])$
Si f est strictement convexe ($f'' > 0$), alors $E[f(x)] = f(E[x])$ (et donc X est une constante)
On peut aussi dire qu'une fonction est convexe sur $[a,b]$ si

$$\forall x_1, x_2 \in [a,b], \forall\lambda \in [0,1] : f[\lambda x_1 + (1-\lambda)x_2] \leq \lambda f(x_1) + (1-\lambda)f(x_2)$$

Si f est concave ($f'' \leq 0$), alors $E[f(x)] \leq f(E[x])$
Si f est strictement concave ($f'' > 0$), alors $E[f(x)] = f(E[x])$ (et donc X est une constante)
On peut aussi dire qu'une fonction est concave sur $[a,b]$ si

$$\forall x_1, x_2 \in [a,b], \forall\lambda \in [0,1] : f[\lambda x_1 + (1-\lambda)x_2] \geq \lambda f(x_1) + (1-\lambda)f(x_2)$$

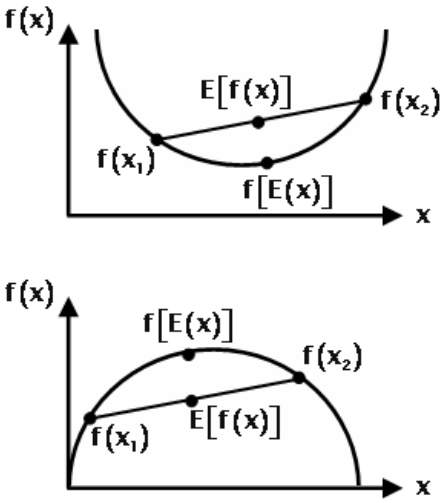

FIG. 1.5 – Représentation graphique des inégalités de Jensen

1.6.2 Information Inequality

Si $p(x)$ et $q(x)$ sont des distributions de probabilité sur le même domaine X, alors :

- L'entropie relative est positive :

$$\boxed{D(p||q) \geq 0} \tag{1.22}$$

Démonstration

Utilisons la définition de l'entropie relative (1.19) :

$$D(p(x)||q(x)) = \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{q(x)}$$

On va ici définir le support A de $p(x)$, sous-ensemble de $\mathcal{X}$ qui comprend tous les x tels que la probabilité $p(x)$ n'est pas nulle. On note donc

$$A = \{x | p(x) > 0\}$$

Ainsi, la somme utilisée dans la définition de l'entropie relative peut se limiter à ce support sans fausser l'égalité :

$$D(p||q) = \sum_{x \in A} p(x) \log_2 \frac{p(x)}{q(x)}$$

Prenons l'opposé des deux membres, utilisons la définition de l'espérance mathématique et rappelons que $-\log \frac{a}{b} = \log \frac{b}{a}$. Nous obtenons :

$$-D(p||q) = \sum_{x \in A} p(x) \log_2 \frac{q(x)}{p(x)} = E[\log_2 \frac{q(x)}{p(x)}]_{X p(x)}$$

Le logarithme étant une fonction concave, nous pouvons ici appliquer les inégalités de Jansen vues au point (1.6.1) :

$$-D(p||q) = E[\log_2 \frac{q(x)}{p(x)}]_{X\overline{p(x)}} \leq \log_2 E[\frac{q(x)}{p(x)}$$

avec, par définition de l'espérance mathématique :

$$\log_2 E[\frac{q(x)}{p(x)} = \log_2 [\sum_{x\in A} p(x)\frac{q(x)}{p(x)}] = \log_2 [\sum_{x\in A} q(x)]$$

Puisque A est un ensemble compris dans X, la somme des x sur l'ensemble A est forcément plus petite ou égale à la somme des x sur l'ensemble X. De ce fait :

$$\log_2 E[\frac{q(x)}{p(x)} = \log_2 [\sum_{x\in A} q(x)] \leq \log_2 [\sum_{x\in \mathcal{X}} q(x)]$$

Puisque la somme de toutes les probabilités de l'ensemble X vaut 1 par défaut, le logarithme s'annulle, ce qui donne finalement :

$$\log_2 E\left[\frac{q(x)}{p(x)}\right] \leq 0$$

Enfin, puisque

$$-D(p||q) \leq log_2 E\left[\frac{q(x)}{p(x)}\right] \leq 0$$

on a bien :

$$D(p||q) \geq 0$$

- L'entropie relative s'annulle si et seulement si $p = q$:

$$\boxed{D(p||q) = 0 \Leftrightarrow p = q} \tag{1.23}$$

Démonstration

Pour montrer cette double implication, montrons les deux implications séparément.
Pour l'implication $\Leftarrow$, on sait que $p = q$, et donc, par la définition de l'entropie relative (1.19) :

$$D(p||q) = \sum_{x\in\mathcal{X}} p(x)\log_2 \frac{p(x)}{q(x)} = 0$$

car le quotient $\frac{p(x)}{q(x)}$ vaut 1.
Pour l'implication $\Rightarrow$, on sait que

$$D(p||q) = E(\log_2 \frac{p}{q}) = 0$$

ainsi que

$$\log_2 [E(\frac{p}{q})] = 0$$

Par l'inégalité de Jansen vue au point (1.6.1), on a que

$$E(\log_2 \frac{p}{q}) = \log_2 [E(\frac{p}{q})]$$

implique que

$$\frac{p}{q} = constante = 1$$

car $\log_2$ est une fonction strictement concave. Ainsi, on a bien que $p = q$

Cette "Information Inequality" a quatre conséquences :

- La distribution uniforme sur X est celle qui maximise l'entropie :

$$H(X) \leq \log_2 |\mathcal{X}| \tag{1.24}$$

Démonstration

Considérons une distribution $p(x)$ quelconque et la distribution uniforme $q(x) = \frac{1}{|\mathcal{X}|}$. Reprenons la définition de l'entropie relative (1.19) et remplaçons-y la valeur de $q(x)$:

$$D(p||q) = \sum_{x\in\mathcal{X}} p(x) \log_2 \frac{p(x)}{q(x)} = \sum_{x\in\mathcal{X}} p(x) \log_2 p(x)|\mathcal{X}|$$

Utilisons ensuite la propriété (1.7) des logarithmes pour séparer la somme :

$$D(p||q) = \sum_{x\in\mathcal{X}} p(x) \log_2 p(x) + \sum_{x\in\mathcal{X}} p(x) \log_2 |\mathcal{X}|$$

Le terme $\log_2 |\mathcal{X}|$ étant constant, il peut sortir de la somme, laissant $\sum_{x\in\mathcal{X}} p(x)$ qui vaut 1 par définition. On a donc au final :

$$D(p||q) = \sum_{x\in\mathcal{X}} p(x) \log_2 p(x) + |\mathcal{X}| = -H(X) + |\mathcal{X}|$$

Comme l'entropie relative est positive par l'Information Inequality, on a :

$$-H(X) + |\mathcal{X}| \geq 0 \Rightarrow |\mathcal{X}| \geq H(X)$$

De plus, si $H(X) = \log_2 |\mathcal{X}|$, alors $D(p||q) = 0$ et donc $p(x) = q(x) = \frac{1}{|\mathcal{X}|}$
Ainsi, l'entropie atteint son maximum lorsque la distribution p(x) est uniforme.

- L'information mutuelle est une grandeur positive et s'annulle lorsque X et Y sont indépendantes.

$$I(X:Y) \geq 0 \quad et \quad I(X:Y) = 0 \Leftrightarrow X, Y \; indpendantes \tag{1.25}$$

Démonstration

Reprenons la définition de l'entropie mutuelle (1.17) :

$$I(X:Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log_2\frac{p(x)p(y)}{p(x,y)} = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log_2\frac{p(x,y)}{p(x)p(y)}$$

Si l'on compare cette dernière égalité à la définition de l'entropie relative (1.19), on peut écrire que :

$$I(X:Y) = D[p(x,y)||p(x)p(y)] \geq 0$$

Ceci démontre la première partie de la formule (1.25). Pour montrer la double implication, montrons d'abord l'implication $\Rightarrow$: si $I(X:Y) = 0$, alors $D[p(x,y)||p(x)p(y)] = 0$. Par l'Information Inequality (1.23), on peut affirmer que $p(x,y) = p(x)p(y)$, ce qui n'est vrai que si X et Y sont indépendantes.
Montrons ensuite l'implication $\Leftarrow$: si X et Y sont indépendantes, cela veut dire que $H(Y|X) = H(Y)$, comme nous l'avons vu à la formule (1.13). Reprenons ensuite la définition intuitive de l'entropie mutuelle (1.16), c'est-à-dire $I(X:Y) = H(Y) - H(Y|X)$, qui est égal à 0 dans ce cas-ci. Ainsi, $I(X:Y) = 0$.

- Le fait de " conditionner " (utiliser une entropie conditionnelle plutôt qu'une entropie simple) ne peut que réduire l'incertitude :

$$H(Y|X) \leq H(Y) \tag{1.26}$$

Démonstration

Nous avons montré précédemment que l'entropie mutuelle est positive : $I(X:Y) \geq 0$. Reprenons la définition de cette entropie mutuelle (1.16) : $I(X:Y) = H(Y) - H(Y|X) \geq 0$. Ceci implique donc bien que $H(Y) \geq H(Y|X)$.

Ceci n'est cependant vrai qu'en moyenne (tout comme pour toute la théorie de Shannon), et non pas symbole par symbole. On ne peut donc pas dire que $H(Y|X = x) \leq H(Y)\ \forall x$

- Montrons la subadditivité de l'entropie. Si l'on reprend la définition de l'entropie mutuelle (1.16) et qu'on y insère la règle de chaîne pour $H(X|Y)$, on obtient :

$$I(X:Y) = H(X) - H(X|Y) = H(X) - [H(X,Y) - H(Y)] = H(X) + H(Y) - H(X,Y)$$

Sachant par la deuxième conséquence de l'Information Inequality (1.25) que $I(X:Y) \geq 0$, on peut écrire que :

$$H(X) + H(Y) - H(X,Y) \geq 0 \quad \Rightarrow \quad H(X) + H(Y) \geq H(X,Y)$$

Aussi, on peut dire que $H(X) + H(Y) = H(X,Y)$ lorsque $I(X:Y) = 0$, c'est-à-dire lorsque X et Y sont indépendantes.

1.7 L'entropie des systèmes multipartites

Soit une collection de n variables aléatoires X_1, X_2, ..., X_n, caractérisée par une distribution $p(x_1, x_2, ..., x_n)$

1.7.1 Règle de chaîne de l'entropie

- Entropie jointe pour n variables
  L'entropie jointe peut être définie, pour n variables, par :

$$H(X_1, X_2, ..., X_n) = H(X_1) + H(X_2|X_1) + H(X_3|X_1, X_2) + ... + H(X_n|X_1, ..., X_{n-1})$$

  Ainsi, on obtient la règle de chaîne pour l'entropie jointe de n variables :

$$\boxed{H(X_1, ..., X_n) = \sum_{i=1}^{n} H(X_i|X_1, ..., X_{i-1})} \tag{1.27}$$

Démonstration

Montrons la formule (1.27) pour $n = 2$, puis pour $n = 3$, et montrons enfin que c'est vrai pour tout n.
Pour $n = 2$, on a, en appliquant la règle de chaîne (1.12), $H(X_1, X_2) = H(X_1) + H(X_2|X_1)$
Pour $n = 3$, on a $H(X_1, X_2, X_3) = H[X_1, (X_2, X_3)]$. On applique donc la règle de chaîne en considérant (X_2, X_3) comme une seule variable :

$$H(X_1, X_2, X_3) = H(X_1) + H(X_2, X_3|X_1) = H(X_1) + H(X_2|X_1, X_3|X_1)$$

Le deuxième membre de cette dernière égalité est également une entropie d'un système bipartite auquel on peut appliquer à nouveau la règle de chaîne :

$$H(X_1, X_2, X_3) = H(X_1) + H(X_2|X_1) + H(X_3|X_1, X_2)$$

Pour tout n, on a donc finalement :

$$H(X_1, ..., X_n) = \sum_{i=1}^{n} H(X_i|X_1, ..., X_{i-1})$$

- Subadditivité pour n variables
  La subadditivité de l'entropie pour n variables s'exprime comme ceci :

$$\boxed{H(X_1, ..., X_n) \leq \sum_{i=1}^{n} H(X_i)} \tag{1.28}$$

  et

$$H(X_1, ..., X_n) = \sum_{i=1}^{n} H(X_i) \quad \Leftrightarrow \quad X_i \textit{ sont indpendantes } \forall i$$

1.7.2 Règle de chaîne pour l'information

- Règle de chaîne

  La règle de chaîne pour l'information entre un ensemble $X_1,\ldots,X_n$ et Y est définie par :

$$H(X_1,...,X_n:Y) = H(X_1:Y) + H(X_2:Y|X_1) + ... + H(X_n:Y|X_1,...,X_{n-1})$$

  Au final, on obtient :

$$\boxed{H(X_1,...,X_n:Y) = \sum_{i=1}^{n} H(X_i:Y|X_1,...X_{i-1})} \tag{1.29}$$

Exemple

Ci-dessous est représenté un diagramme de Venn pour un exemple lorsque $n = 3$, avec les quatre variables X_1, X_2, X_3 et Y

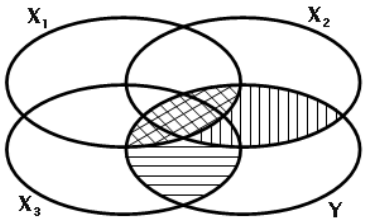

FIG. 1.6 – Diagramme de Venn pour 4 variables à information commune

- La surface hachurée double est l'information que Y partage avec X_1, c'est-à-dire $H(X_1 : Y)$
- La surface hachurée verticalement est l'information que Y partage avec X_2 sans la moindre information venant de X_1, c'est-à-dire $H(X_2 : Y|X_1)$
- La surface hachurée horizontalement est l'information que Y partage avec X_3 sans la moindre information venant de X_1 et X_2, c'est-à-dire $H(X_3 : Y|X_1, X_2)$

On voit que la somme de ces trois surfaces donne l'information que Y partage avec l'ensemble des X_i (pour i allant de 1 à 3). On note donc :

$$H(X_1, X_2, X_3 : Y) = H(X_1 : Y) + H(X_2 : Y|X_1) + H(X_3 : Y|X_1, X_2)$$

ce qui correspond bien à la formule (1.29) pour $n = 3$.

Démonstration

Par la définition de l'information mutuelle (1.16), on sait que

$$H(X_1,...X_n : Y) = H(X_1,...,X_n) - H(X_1,...X_n|Y)$$

Or, par la règle de chaîne de l'entropie (1.27), on sait que

$$H(X_1, ..., X_n) = \sum_{i=1}^{n} H(X_i|X_1, ..., X_{i-1})$$

et que donc :

$$H(X_1, ..., X_n|Y) = \sum_{i=1}^{n} H(X_i|X_1, ..., X_{i-1}, Y)$$

En rassemblant ces deux équations dans la première, on obtient :

$$H(X_1, ...X_n : Y) = \sum_{i=1}^{n} H(X_i|X_1, ..., X_{i-1}) - H(X_i|X_1, ..., X_{i-1}, Y)$$

Par la définition de l'information mutuelle (1.16), on peut affirmer que cette équation est équivalente à :

$$H(X_1, ...X_n : Y) = \sum_{i=1}^{n} H(X_i : Y|X_1, ..., X_{i-1})$$

- Entropie mutuelle conditionnelle (tripartite)
  Il y a une réduction de l'incertitude sur X amenée par la connaissance de Y conditionnellement à Z. C'est de là que part la définition de l'entropie mutuelle conditionnelle. Pour un système tripartite, voici ce que cela donne :

$$H(X : Y|Z) = H(X|Z) - H(X|Y, Z) \tag{1.30}$$

où :
- $H(X : Y|Z)$ est l'entropie commune à X et Y en enlevant toute contribution de l'entropie de Z (partie hachurée double).
- $H(X|Z)$ est l'entropie de X en enlevant toute contribution de l'entropie de Z.
- $H(X|Y, Z)$ est l'entropie de X en enlevant toute contribution des entropies de Y et de Z (parties hachurée double et hachurée horizontale).

Ceci s'observe sur le schéma ci-dessous.

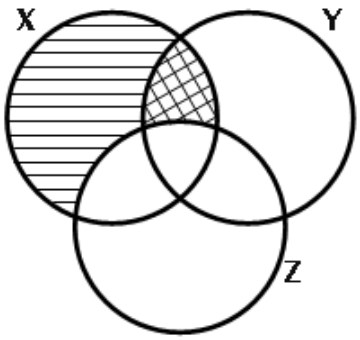

FIG. 1.7 – Diagramme de Venn pour un système tripartite X,Y,Z

Remarquons qu'en utilisant la règle de chaîne de l'entropie (1.27) sur le dernier terme de l'équation, on obtient :

$$H(X:Y|Z) = H(X|Z) - [H(X,Y,Z) - H(Z) - H(Y|Z)]$$

où $H(X,Y,Z)$ est l'entropie totale du système tripartite.

- Subadditivité forte
  Pour les systèmes binaires, on avait la subadditivité (1.25), impliquant que l'entropie relative est une grandeur positive par l'Information Inequality (1.22). Pour les systèmes tripartites, on a la subadditivité forte, exprimée par :

$$\boxed{H(X:Y|Z) \geq 0 \quad \Leftrightarrow \quad D[p(x|y)||q(x|y)] \geq 0} \tag{1.31}$$

Démonstration

Reprenons la définition (1.19) de l'entropie relative et appliquons-là aux distributions $p(y|x)$ et $q(y|x)$:

$$D[p(y|x)||q(y|x)] = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p(x,y) \log_2 \frac{p(y|x)}{q(y|x)}$$

En utilisant la propriété (1.1) des probabilités, on obtient :

$$D[p(y|x)||q(y|x)] = \sum_{x\in\mathcal{X}} p(x) \sum_{y\in\mathcal{Y}} p(y|x) \log_2 \frac{p(y|x)}{q(y|x)}$$

Il faut ensuite remarquer que la somme sur les y n'est autre que la définition de l'entropie relative pour les distributions p et q ayant un x fixé :

$$D[p(y|x)||q(y|x)] = \sum_{x\in\mathcal{X}} p(x) D[p(y|x_{\text{fixé}})||q(y|x_{\text{fixé}})]$$

Etant donné que l'entropie relative présente dans cette somme est positive par l'Information Inequality (1.22), on peut affirmer que la somme sur les x ci-dessus est également positive (par définition d'une probabilité). On sait donc que

$$D[p(y|x)||q(y|x)] \geq 0$$

propriété qui nous servira dans la suite de cette démonstration.
Utilisons maintenant la définition de l'entropie conditionnelle (1.11) :

$$\begin{aligned} &H(X:Y|Z) = H(X|Z) - H(X|Y,Z) \\ &= -\textstyle\sum_{x\in\mathcal{X}} \sum_{z\in\mathcal{Z}} p(x,z) \log_2 p(x|z) - \Big[-\sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \sum_{z\in\mathcal{Z}} p(x,y,z) \log_2 p(x|y,z)\Big] \end{aligned}$$

On va ici utiliser un artifice mathématique et ajouter l'élément unitaire

$$\sum_{y\in\mathcal{Y}} p(y|x,z) = 1$$

au premier terme ci-dessus :

$$H(X:Y|Z) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}\sum_{z\in\mathcal{Z}} p(x,z)p(y|x,z)\log_2 p(x|z) - \left[-\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}\sum_{z\in\mathcal{Z}} p(x,y,z)\log_2 p(x|y,z)\right]$$

Par la propriété (1.1) des probabilités, on sait que $p(x,y,z) = p(x,z)p(y|x,z)$, et donc :

$$H(X:Y|Z) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}\sum_{z\in\mathcal{Z}} p(x,y,z)\log_2 p(x|z) - \left[-\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}\sum_{z\in\mathcal{Z}} p(x,y,z)\log_2 p(x|y,z)\right]$$

Rassemblons ensuite les deux sommes en une par la propriété des logarithmes (1.7) :

$$H(X:Y|Z) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}\sum_{z\in\mathcal{Z}} p(x,y,z)[\log_2 p(x|z) - \log_2 p(x|y,z)]$$

$$H(X:Y|Z) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}\sum_{z\in\mathcal{Z}} p(x,y,z)\log_2 \frac{p(x|z)}{p(x|y,z)}$$

Comme $p(x,y,z) = p(x|y,z)p(y,z)$, on obtient :

$$H(X:Y|Z) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}\sum_{z\in\mathcal{Z}} p(x,y,z)\log_2 \frac{p(x|z)p(y,z)}{p(x,y,z)}$$

Ensuite, utilisons le fait que $p(x,y,z) = p(x,y|z)p(z)$ et $p(y,z) = p(y|z)p(z)$, ce qui donne :

$$H(X:Y|Z) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}\sum_{z\in\mathcal{Z}} p(z)p(x,y|z)\log_2 \frac{p(x|z)p(y|z)p(z)}{p(x,y|z)p(z)}$$

Les termes $p(z)$ du logarithme se simplifiant, la somme sur les z ne s'applique maintenant plus qu'au premier $p(z)$, et donne 1 par définition. On a donc :

$$H(X:Y|Z) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y|z)\log_2 \frac{p(x,y|z)}{p(x|z)p(y|z)}$$

Cette dernière équation n'est rien d'autre que la définition de l'entropie relative : $D[p(x,y|z)||p(x|z)p(y|z)]$, et nous avons trouvé précédemment que cette grandeur était positive.
De plus, $H(X:Y|Z) = D[p(x,y|z)||p(x|z)p(y|z)] = 0$, et donc par l'Information Inequality (1.23), on a que $p(x,y|z) = p(x|z)p(y|z)$, c'est-à-dire uniquement lorsque X et Y sont indépendantes conditionnellement à Z.

Chapitre 2

Equipartition asymptotique

2.1 Séquences typiques

2.1.1 Introduction

Soit une très longue séquence de symboles, émise par une source, $\overline{x} = x_1, x_2, ..., x_n$, de manière à pouvoir considérer que n tend vers l'infini. La source est ainsi caractérisée par les variables aléatoires $X_1, X_2, ..., X_n$ indépendantes (un symbole émis ne dépend pas des précédents) et identiquement distribuées ($X_i \tilde{} p(x) \forall i$). On dit que les variables sont alors IID.

2.1.2 Application de la loi des grands nombres

Rappelons la loi des grands nombres :

$$\frac{1}{n}\sum_{i=1}^{n} x_i \overset{n\to\infty}{\to} E[x] = \sum_{x\in\mathcal{X}} xp(x)$$

et appliquons-là à la théorie de l'information. Partons de :

$$-\frac{1}{n}\log_2 p(x_1, ..., x_n)$$

Puisque les X_i sont indépendantes, $p(x_1, ..., x_n)$ est égal au produit de toutes les probabilitiés de chaque x_i :

$$p(x_1, ..., x_n) = \prod_{i=1}^{n} p(x_i)$$

Comme le logarithme d'un produit est égal à la somme des logarithmes, on peut noter :

$$-\frac{1}{n}\log_2 p(x_1, ..., x_n) = -\frac{1}{n}\log_2 \prod_{i=1}^{n} p(x_i) = -\frac{1}{n}\sum_{i=1}^{n} \log_2 p(x_i)$$

Appliquons ici la loi des grands nombres :

$$-\frac{1}{n}\log_2 p(x_1, ..., x_n) = -\frac{1}{n}\sum_{i=1}^{n} \log_2 p(x_i) \overset{n\to\infty}{\to} E[-\log_2 p(x)]$$

Par la définition de l'espérance mathématique, nous savons que

$$E[-\log_2 p(x)] = -\sum_{x\in\mathcal{X}} p(x)\log_2 p(x) = H(X)$$

Ainsi, on obtient au final, lorsque n tend vers l'infini :

$$-\frac{1}{n}\log_2 p(x_1,...,x_n) = H(X)$$

ce qui donne :

$$\log_2 p(x_1,...,x_n) = -nH(X)$$

c'est-à-dire :

$$p(x_1,...,x_n) = n^{-nH(X)}$$

On verra par la suite que la probabilité qu'une séquence $\overline{x} = x_1,...,x_n$ ait pour distribution $p(x_1,...,x_n) = 2^{-n[H(X)\pm\epsilon]}$ est quasiment unitaire.

2.1.3 Définition

Une séquence typique est une séquence telle que

$$2^{n[H(X)+\epsilon]} \leq p(\overline{x}) \leq 2^{n[H(X)-\epsilon]} \tag{2.1}$$

2.2 Théorème d'équipartition asymptotique (AEP)

2.2.1 Théorème

Si $X_1,...,X_n$ sont des variables aléatoires IID (indépendantes et identiquement distribuées), alors

$$-\frac{1}{n}\log_2 p(\overline{x}) \overset{n\to\infty}{\to} H(X) \tag{2.2}$$

2.2.2 Ensemble typique

Un ensemble typique $A_\epsilon^{(n)}$ est un ensemble de séquences $\overline{x}$ telles que

$$2^{-n[H(X)+\epsilon]} \leq p(\overline{x}) \leq 2^{-n[H(X)-\epsilon]} \tag{2.3}$$

c'est-à-dire :

$$A_\epsilon^{(n)} = \left\{ \overline{x}\ tq\ | -\frac{1}{n}\log_2 p(\overline{x}) - H(X)| \leq \epsilon \right\}$$

Cet ensemble est représenté sur le schéma ci-dessous.

2.2.3 Propriétés

- Propriété n°1
  La probabilité qu'une séquence émise par la source soit typique est arbitrairement proche de 1 :

$$\boxed{\forall \varepsilon > 0, \exists n_0 : \forall n \geq n_0, P\left[\overline{x} \in A_\varepsilon^{(n)}\right] \geq 1-\varepsilon} \tag{2.4}$$

  ε étant la tolérance

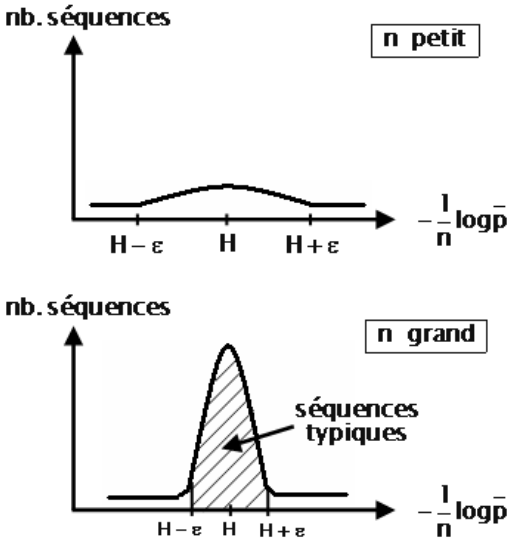

FIG. 2.1 – Représentation de l'ensemble typique

- Propriété n°2
  La taille de l'ensemble typique $\left| A_\varepsilon^{(n)} \right|$ est bornée supérieurement :

$$\boxed{\left| A_\varepsilon^{(n)} \right| \leq 2^{n\,(H+\varepsilon)}} \tag{2.5}$$

Démonstration

Sachant, par la définition d'une probabilité, que

$$\sum_{\overline{x} \in \mathcal{X}^n} p(\overline{x}) = 1$$

et que l'ensemble typique $\mathcal{X}^n$ est inclus dans l'ensemble $A_\varepsilon^{(n)}$, alors :

$$\sum_{\overline{x} \in A_\varepsilon^{(n)}} p(\overline{x}) \leq \sum_{\overline{x} \in \mathcal{X}^n} p(\overline{x}) = 1$$

De plus, comme on l'a précédemment vu pour les séquences typiques, $p\,(\overline{x}) \geq\ 2^{-n\,(H+\varepsilon)}$, ce qui implique que :

$$2^{-n\,(H+\varepsilon)} \sum_{\overline{x} \in A_\varepsilon^{(n)}} 1 \leq \sum_{\overline{x} \in A_\varepsilon^{(n)}} p\,(\overline{x}) \leq 1$$

où

$$\sum_{\overline{x} \in A_\varepsilon^{(n)}} 1 = \left| A_\varepsilon^{(n)} \right|$$

On a donc finalement que :

$$\left| A_\varepsilon^{(n)} \right| \, 2^{-n\,(H+\varepsilon)} \leq 1$$

Ce qui implique :

$$\left| A_\varepsilon^{(n)} \right| \leq 2^{n\,(H+\varepsilon)}$$

- Propriété n°3

La taille de l'ensemble typique est bornée inférieurement :

$$\boxed{\left| A_\varepsilon^{(n)} \right| > 2^{n\,(H-\varepsilon)}\,(1-\varepsilon)} \tag{2.6}$$

Démonstration

Par la propriété n°1 des séquences typiques (2.4), nous savons que :

$$1-\varepsilon < P\left[\overline{x} \in A_\varepsilon^{(n)}\right]$$

avec

$$P\left[\overline{x} \in A_\varepsilon^{(n)}\right] = \sum_{\overline{x} \in A_\varepsilon^{(n)}} p\,(\overline{x})$$

Puisque $p(\overline{x}) \leq 2^{-n(H-\varepsilon)}$, on a que :

$$1-\varepsilon < \sum_{\overline{x} \in A_\varepsilon^{(n)}} p\,(\overline{x}) \leq \left| A_\varepsilon^{(n)} \right| 2^{-n(H-\varepsilon)}$$

Ceci implique directement que :

$$(1-\varepsilon)\,2^{n(H-\varepsilon)} \leq \left| A_\varepsilon^{(n)} \right|$$

2.2.4 Conclusion

Au final, par les propriétés 2 (2.5) et 3 (2.6), on obtient :

$$(1-\varepsilon)\,2^{n(H-\varepsilon)} < \left| A_\varepsilon^{(n)} \right| \leq 2^{n(H+\varepsilon)}$$

Lorsque n tend vers l'infini et que ε tend vers 0, on trouve que :

$$\left| A_\varepsilon^{(n)} \right| \to 2^{nH} \ll 2^{n \log_2 |\,\mathcal{X}\,|}$$

Ainsi, l'ensemble typique est fortement restreint par rapport à la totalité des séquences $\mathcal{X}^n$, mais il concentre la quasi-totalité de la probabilité. On a en effet vu que

$$P\left[\overline{x} \in A_\varepsilon^{(n)}\right] \approx 2^{nH} 2^{-nH} = 1$$

car on trouve la probabilité totale en multipliant la densité de probabilité de l'ensemble par le "volume" de cet ensemble.

On verra par la suite que c'est cette propriété qui rend la compression possible : si toutes les séquences étaient typiques, alors il n'y aurait aucune compression possible.

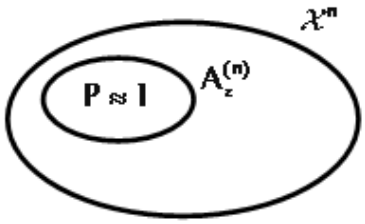

FIG. 2.2 – Représentation de l'ensemble typique et de l'ensemble $\mathcal{X}^n$

2.3 Utilisation du théorème pour la compression des données

2.3.1 Introduction

Divisons l'ensemble des séquences possibles $\mathcal{X}^n$ en l'ensemble des séquences typiques $A_\varepsilon^{(n)}$ et l'ensemble des séquences atypiques $\overline{A_\varepsilon^{(n)}}$, et numérotons-les. Voyons comment sont les tailles de ces deux ensembles :

- Séquences typiques
  On a vu précédemment que :
  $$\left| A_\varepsilon^{(n)} \right| \leq 2^{n(H+\varepsilon)}$$
  donc le nombre de bits $\ell(\overline{x})$ pour coder le numéro d'une séquence est inférieur à $n(H+\varepsilon)$, auquel on rajoute un bit pour aller à la puissance de 2 supérieure, et un bit appelé bit de "Flag" permettant de distinguer une séquence typique d'une atypique. Ainsi, le nombre de bit doit respecter :
  $$\ell(\overline{x}) \leq n(H+\varepsilon) + 2 \tag{2.7}$$
- Séquences atypiques
  On a vu précédemment que :
  $$\left| \overline{A_\varepsilon^{(n)}} \right| \leq 2^{n \log_2 |\mathcal{X}|}$$
  donc le nombre de bits pour coder le numéro d'une séquence $\ell(\overline{x})$ est inférieur à $n \log_2 |\mathcal{X}| + 2$:
  $$\ell(\overline{x}) \leq n \log_2 |\mathcal{X}| + 2 \tag{2.8}$$
  et on a rajouté deux bits pour la même raison que ci-dessus.

On obtient alors un schéma d'encodage à correspondance biunivoque, c'est-à-dire qu'à tout x_i correspond un mot-code et inversement. Ce schéma est efficace mais cependant inutilisable, puisqu'on suppose connaître toutes les séquences typiques.

Prenons l'exemple suivant : une source binaire envoit des bits avec les probabilités suivantes :

$$\begin{cases} p(x=0) = 1-p \\ p(x=1) = p \end{cases}$$

La source envoyant des séquences de n bits, on devrait avoir environ 2^{nH} séquences typiques avec un $n(1-p)$ zéros et np uns, chaque séquence typique ayant une probabilité d'environ 2^{-nH}. En effet, la probabilité qu'une séquence soit typique est donnée par :

$$p(\overline{x}) = P[\text{séquence contient np bits à 1 et } (1-p)\,\text{n bits à 0}]$$

Il s'agit donc d'une densité de probabilité de type binomiale, c'est-à-dire :

$$p\left(\overline{\mathrm{x}}\right) = p^{np}\left(1-p\right)^{n-np} = 2^{\log_2\left[p^{np}(1-p)^{n(1-p)}\right]} = 2^{\log_2[p^{np}]+\log_2\left[(1-p)^{n(1-p)}\right]}$$

et donc :

$$p\left(\overline{\mathrm{x}}\right) = 2^{np\log_2 p\, +\, n(1-p)\log_2(1-p)} = 2^{n[p\log_2 p\, +\, (1-p)\log_2(1-p)]} = 2^{-nH(X)}$$

Il est aussi possible de montrer le nombre de séquence typiques. En effet, ce nombre est le nombre de possibilité de mettre np bits à 1 parmi n bits :

$$N = \left(\begin{array}{c} n \\ np \end{array}\right) = \frac{n!}{(np)!\left[n\left(1-p\right)\right]!} \approx \frac{2^{n\log n}}{2^{np\log(np)}\, 2^{n(1-p)\log[n(1-p)]}} = 2^{n\log n - np\log(np) - n(1-p)\log[n(1-p)]}$$

En développant ceci, on obtient finalement :

$$N = 2^{n\log n - np\log n - np\log p - n(1-p)\log n - n(1-p)\log(1-p)} = 2^{n[-p\log p - (1-p)\log(1-p)]} = 2^{nH(X)}$$

2.3.2 Théorème fondamental pour le codage (longueur moyenne)

Pour toute source sans mémoire, il existe un code à décodage unique qui transforme l'alphabet de cette source en messages binaires de longueur moyenne proche de l'entropie de la source :

$$\boxed{E\left[\frac{1}{n}\ell\left(\mathcal{X}^n\right)\right] \leq H\left(X\right) + \varepsilon} \tag{2.9}$$

Démonstration

La définition de la longueur moyenne est donnée par l'espérance mathématique de la longueur des séquences :

$$E\left[\ell\left(\overline{x}\right)\right] = \sum_{\overline{x}\in\mathcal{X}^n} p\left(\overline{x}\right)\ell\left(\overline{x}\right)$$

Séparons l'ensemble $\mathcal{X}^n$ en deux ensembles vus au point précédent : les séquences typiques et les séquences atypiques.

$$E\left[\ell\left(\overline{x}\right)\right] = \sum_{\overline{x}\in A_\varepsilon^{(n)}} p\left(\overline{x}\right)\ell\left(\overline{x}\right) + \sum_{\overline{x}\in \overline{A_\varepsilon^{(n)}}} p\left(\overline{x}\right)\ell\left(\overline{x}\right)$$

On a vu que la longueur des séquences typiques ne dépassait pas $n\ \left(H+\varepsilon\right)+2$ bits et que celle des séquences atypiques ne dépassait pas $n\ \log_2\left|\mathcal{X}\right|+2$, cf. formules (2.7) et (2.8). Ainsi, on a :

$$E\left[\ell\left(\overline{x}\right)\right] \leq \sum_{\overline{x}\in A_\varepsilon^{(n)}} p\left(\overline{x}\right)\left[n\left(H+\varepsilon\right)+2\right] + \sum_{\overline{x}\in \overline{A_\varepsilon^{(n)}}} p\left(\overline{x}\right)\left[n\log_2\left|\mathcal{X}\right|+2\right]$$

En faisant sortir des sommes les éléments qui sont constants, on obtient :

$$E\left[\ell\left(\overline{x}\right)\right] \leq \left[n\left(H+\varepsilon\right)+2\right]\sum_{\overline{x}\in A_\varepsilon^{(n)}} p\left(\overline{x}\right) + \left[n\log_2\left|\mathcal{X}\right|+2\right]\sum_{\overline{x}\in \overline{A_\varepsilon^{(n)}}} p\left(\overline{x}\right)$$

On peut ici remplacer les sommes par leur définition, à savoir :

$$\sum_{\overline{x} \in A_\varepsilon^{(n)}} p\left(\overline{x}\right) = P\left(\overline{x} \in A_\varepsilon^{(n)}\right)$$
$$\sum_{\overline{x} \in \overline{A_\varepsilon^{(n)}}} p\left(\overline{x}\right) = P\left(\overline{x} \in \overline{A_\varepsilon^{(n)}}\right)$$

On a donc :

$$E\left[\ell\left(\overline{x}\right)\right] \leq n\ \left(H+\varepsilon\right) P\left(\overline{x} \in A_\varepsilon^{(n)}\right) + n\ \log_2\left|\mathcal{X}\right| P\left(\overline{x} \in \overline{A_\varepsilon^{(n)}}\right) + 2\underbrace{\left[P\left(\overline{x} \in A_\varepsilon^{(n)}\right) + P\left(\overline{x} \in \overline{A_\varepsilon^{(n)}}\right)\right]}_{1}$$

On sait également que :

$$P\left(\overline{x} \in A_\varepsilon^{(n)}\right) \approx 1$$
$$P\left(\overline{x} \in \overline{A_\varepsilon^{(n)}}\right) \leq \varepsilon$$

Que l'on insère dans l'équation précédente :

$$E\left[\ell\left(\overline{x}\right)\right] \leq n\left(H+\varepsilon\right) + n\varepsilon\ \log_2\left|\mathcal{X}\right| + 2$$

Mettons n en évidence :

$$E\left[\ell\left(\overline{x}\right)\right] \leq n\left[H + \varepsilon + \varepsilon\log_2\left|\mathcal{X}\right| + \frac{2}{n}\right]$$

Mettons ensuite ε en évidence :

$$E\left[\ell\left(\overline{x}\right)\right] \leq n\left[H + \varepsilon\left(1 + \log_2\left|\mathcal{X}\right|\right) + \frac{2}{n}\right]$$

Ainsi, entre crochet, l'entropie H est additionnée à deux termes très petits, que l'on va appeler ε'. On a donc finalement, lorsque n tend vers l'infini et ε tend vers 0, ε' tend vers 0 et :

$$E\left[\ell\left(\overline{x}\right)\right] \leq nH$$

Une compression est donc possible, au lieu d'avoir $\ell\left(\overline{x}\right) = n\log\left|\mathcal{X}\right|$, on aurait $\ell\left(\overline{x}\right) = nH$

Chapitre 3

Codage de source

3.1 Introduction

3.1.1 But et exemple

En compressant, on utilise mieux la capacité d'un canal sans erreur. Pour cela, on remplace les messages émis par la source (alphabet $\mathcal{X}$) par des messages dans un autre alphabet (alphabet $\mathcal{D}$). Un exemple de compression est le morse, où l'alphabet classique $\mathcal{X} = \{A, B, ..., Z\}$ est codé en un autre alphabet $\mathcal{D} = \{., -, espace\}$ de la manière la plus efficace qui soit (la lettre la plus fréquente, E, est codée avec un seul symbole, tandis que la plus rare, Q, est codée avec 4 symboles).

3.1.2 Définitions

- Un code source C est un mapping (conversion) entre l'alphabet $\mathcal{X}$ et l'alphabet $\mathcal{D}$. On note C(x) le mot-code associé au symbole x, et $\ell(x)$ la longueur de ce mot-code. Un code source doit posséder des propriétés telles que l'information de base puisse être reconstruite, et doit utiliser au mieux la capacité d'un canal (cf. morse ci-dessus).
- La longueur moyenne d'un code source C pour une source X est donnée par :

$$\boxed{L(C) = \sum_{x \in \mathcal{X}} p(x)\ell(x)} \tag{3.1}$$

Prenons l'exemple de codage suivant pour l'alphabet $\mathcal{X} = \{a, b, c, d\}$: Les probabilités d'avoir un certain symbole en provenance de la source sont données par :

$$\begin{cases} P(X = a) = \frac{1}{2} \\ P(X = b) = \frac{1}{4} \\ P(X = c) = \frac{1}{8} \\ P(X = d) = \frac{1}{8} \end{cases}$$

On décide de coder chaque symbole en bits. Par exemple, $C(a)$ est le codage de a. On note donc :

$$\begin{cases} C(a) = 0 \\ C(b) = 10 \\ C(c) = 110 \\ C(d) = 111 \end{cases}$$

Ce qui donne les longueurs suivantes :

$$\begin{cases} \ell(a) = 1 \\ \ell(b) = 2 \\ \ell(c) = 3 \\ \ell(d) = 3 \end{cases}$$

A partir de ces informations, on peut donc calculer l'entropie de la source et la longueur moyenne du codage. L'entropie vaut :

$$H(X) = \frac{1}{2}\underbrace{\log_2 2}_{1} + \frac{1}{4}\underbrace{\log_2 4}_{2} + \frac{2}{8}\underbrace{\log_2 8}_{3} = 1,75\ bits/symbole$$

et la longueur moyenne vaut :

$$L(C) = \frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \frac{3}{8} = \frac{7}{4} = 1,75\ bits/symbole$$

On a dans ce cas-ci un codage optimal, et en plus la longueur moyenne est égale à l'entropie, ce qui sont deux critères qui ne sont pas forcément toujours compatibles. C'est le cas du codage suivant. On a les probabilités suivantes :

$$\begin{cases} P(X = a) = \frac{1}{3} \\ P(X = b) = \frac{1}{3} \\ P(X = c) = \frac{1}{3} \end{cases}$$

avec le codage et les longueurs suivantes :

$$\begin{cases} C(a) = 0 \\ C(b) = 10 \\ C(c) = 11 \end{cases} \quad \begin{cases} \ell(a) = 1 \\ \ell(b) = 2 \\ \ell(c) = 2 \end{cases}$$

On obtient alors la longueur moyenne

$$L(C) = \frac{1}{3}1 + \frac{2}{3}2 = \frac{5}{3} = 1,667\ bits/symbole$$

et l'entropie

$$H(X) = \log_2 3 = 1,58\ bits/symbole$$

On voit ici que L est différent de H. Pourtant, il ne faut pas avoir fait 5 ans en polytech pour voir que le codage est optimal.

3.2 Classes de codes

- Codes non singuliers
  C est un code non singulier si :

  $$\forall x_i, x_j \in X : \{x_i \neq x_j \Rightarrow C(x_i) \neq C(x_j)\}$$

  Contrairement aux codes singuliers, ils ne sont pas ambigus.

- Codes décodables de façon unique (DFU)
  C est un code DFU si son extension, c'est-à-dire la concaténation des mots-code individuels $C^*(x_1, ..., x_n) = C(x_1)C(x_2)...C(x_n)$, est non singulière.

- Codes instantanés : C est un code instantané (ou prefix code, ou self-ponctuating code) s'il vérifie la condition de préfixe : "aucun mot-code ne peut être le préfixe d'un autre mot-code".

Exemple

Prenons le codage suivant :

$$\begin{cases} C(a) = 0 \\ C(b) = 10 \\ C(c) = 111 \\ C(d) = 110 \end{cases}$$

Introduire 1110 serait rendre le code non instantané puisque 111 serait son préfixe. Si on reçoit la séquence 01011111010, on peut directement, bit après bit, décoder l'arrivée de nouveaux symboles, et traduire directement en abcdb sans qu'il ait fallu envoyer de bits pour indiquer la séparation entre les symboles, ou sans qu'il ait fallu attendre plusieurs bit excédentaires pour connaître l'identité d'un symbole.

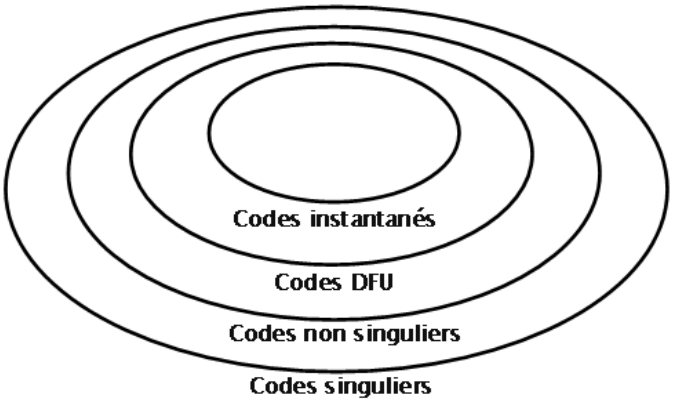

FIG. 3.1 – Schéma reprenant les différentes classes de codes

Ci-dessus est reprśenté un schéma comprenant les différentes classes de codes, groupées en ensembles. D'autres exemples sont donnés dans le tableau ci-dessous.

	Singulier	Non singulier	DFU	Instantané
X=a	0	0	10	0
X=b	0	010	00	10
X=c	0	01	11	110
X=d	0	10	110	111

FIG. 3.2 – Exemples de codes pour chaque classe

3.3 Inégalité de Kraft

L'inégalité de Kraft est une condition nécessaire et suffisante d'existence des codes instantanés. On dit qu'un code est instantané si et seulement si

$$\boxed{\sum_{i=1}^{n} D^{-\ell_i} \leq 1} \tag{3.2}$$

Démonstration

Pour démontrer cette inégalité, on utilise l'arbre de décodage, qui est une manière de décoder le message fourni par la source. Cet arbre est dit d'ordre D lorsque chaque noeud peut donner naissance à D branches. Un exemple d'arbre binaire (ordre D=2) est représenté ci-dessous.

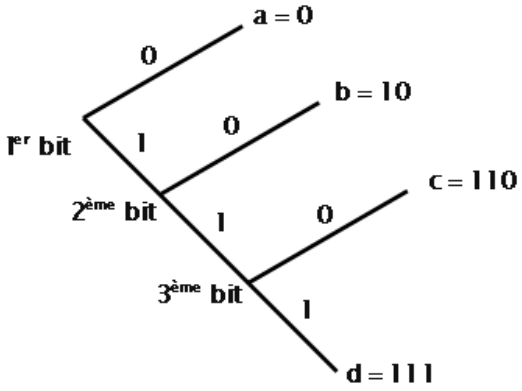

FIG. 3.3 – Arbre binaire de décodage (D=2)

Si le code se veut instantané, alors il existe des zones interdites pour placer les différents symboles. En effet, pour qu'il n'y ait aucun code qui soit le suffixe d'un autre, les zones en pointillés ci-dessous ne peuvent être utilisées.

Soit ℓ_{max} la longueur du mot-code le plus long (dans l'exemple ci-dessus, il s'agit de $\ell_{max} = 3$). Si on se place sur un noeud quelconque, alors il existe $D^{\ell_{max}-\ell_i}$ descendants entre ce noeud et le niveau ℓ_{max}. La somme de ces descendants doit être inférieure au nombre de descendants maximal, donné par $D^{\ell_{max}}$, c'est-à-dire :

$$\sum_{i=1}^{n} D^{\ell_{\max}-\ell_i} \leq D^{\ell_{\max}}$$

En simplifiant, on obtient donc :

$$\sum_{i=1}^{n} D^{-\ell_i} \leq 1$$

Plus on s'éloigne de 1, moins le codage est optimal. Sur l'exemple ci-dessus, le codage est optimal puisque

$$\sum_{i=1}^{n} D^{-\ell_i} = 1$$

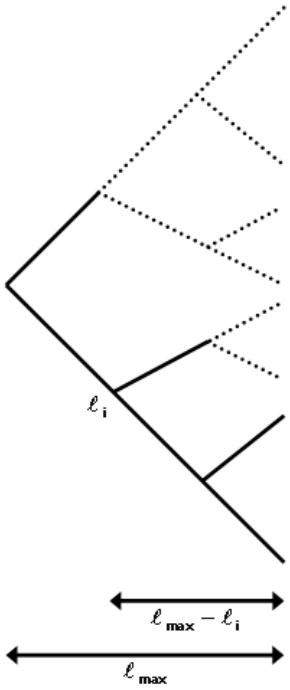

FIG. 3.4 – Exemple d'arbre de décodage avec zones interdites

3.4 Codes instantanés optimaux

On voudrait minimiser la longueur moyenne L(C) d'un mot-code tout en respectant l'inégalité de Kraft (c'est-à-dire en utilisant un code instantané). Pour cela, on doit recourir aux multiplicateurs de Lagrange. On doit cependant faire deux hypothèses : les ℓ_i sont réels, et le code est optimal, c'est-à-dire que l'inégalité de Kraft est saturée :

$$\sum_{i=1}^{n} D^{-\ell_i} = 1$$

Démonstration

Montrons comment minimiser la longueur moyenne à partir des multiplicateurs de Lagrange. Premièrement, il faut poser un opérateur de Lagrange J, défini par la grandeur à minimiser additionnée d'une condition multipliée par les multiplicateurs de Lagrange.
La condition est ici l'optimalité, c'est-à-dire :

$$\sum_{i=1}^{n} D^{-\ell_i} - 1$$

Cette expression s'annulle lorsque la condition est respectée. On a donc l'expression suivante :

$$J = \sum_{i=1}^{n} p_i \ell_i + \lambda(\sum_{i=1}^{n} D^{-\ell_i} - 1)$$

où λ est le multiplicateur de Lagrange. Essayons ensuite de minimiser J en annulant ses dérivées par rapport aux différentes variables ℓ_i et λ :

$$\begin{cases} 0 = \frac{\partial J}{\partial \ell_i} = 0 + 0 + ... + p_i + 0 + ... + \lambda \frac{\mathrm{d}\left(\mathrm{D}^{-\ell_\mathrm{i}}\right)}{\mathrm{d}\ell_\mathrm{i}} \\ 0 = \frac{\partial J}{\partial \lambda} = \sum\limits_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{D}^{-\ell_\mathrm{i}} - 1 \quad \Rightarrow \quad \sum\limits_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{D}^{-\ell_\mathrm{i}} = 1 \end{cases}$$

La dérivée de $D^{-\ell_i}$ est donnée par $(-\ell_i)' D^{-\ell_i} \log_2 D$, ce qui donne :

$$\begin{cases} 0 = p_i + \lambda \left(-\ell_\mathrm{i}\right)' \mathrm{D}^{-\ell_\mathrm{i}} \log_2 D = p_i - \lambda \, \mathrm{D}^{-\ell_\mathrm{i}} \log_2 D \quad \Rightarrow \quad \mathrm{D}^{-\ell_\mathrm{i}} = \frac{p_i}{\lambda \log_2 D} \\ \sum\limits_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{D}^{-\ell_\mathrm{i}} = 1 \end{cases}$$

Si on remplace $\mathrm{D}^{-\ell_\mathrm{i}}$ dans la deuxième équation, on obtient :

$$1 = \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{D}^{-\ell_\mathrm{i}} = \sum_{\mathrm{i}=1}^{\mathrm{n}} \frac{p_i}{\lambda \log_2 D} = \frac{1}{\lambda \log_2 D} \sum_{\mathrm{i}=1}^{\mathrm{n}} p_i \quad \Rightarrow \quad \sum_{\mathrm{i}=1}^{\mathrm{n}} p_i = \lambda \log_2 D$$

Par la définition d'une probabilité, on sait que la somme des probabilité vaut 1, donc :

$$\sum_{\mathrm{i}=1}^{\mathrm{n}} p_i = \lambda \log_2 D = 1 \quad \Rightarrow \quad \lambda = \frac{1}{\log_2 D}$$

Connaissant la valeur de λ, on peut la remplacer dans l'expression trouvée précédemment :

$$\mathrm{D}^{-\ell_\mathrm{i}} = \frac{p_i}{\lambda \log_2 D} = \frac{p_i \log_2 D}{\log_2 D} = p_i \tag{3.3}$$

Prenons le logarithme des deux membres, on obtient :

$$\log_D \mathrm{D}^{-\ell_\mathrm{i}} = \log_D p_i$$

et puisque $\log_D \mathrm{D}^{-\ell_\mathrm{i}} = -\ell_\mathrm{i} \log_D \mathrm{D} = -\ell_\mathrm{i}$, on a finalement :

$$\boxed{\ell_\mathrm{i}^{\min} = -\log_D p_i} \tag{3.4}$$

Etant donné que le logarithme est négatif, on obtient bien une longueur minimale positive. Remarquons qu'en pratique, les longueurs minimales sont réelles et on a pour habitude de prendre l'entier supérieur le plus proche.
Remarquons aussi que la longueur moyenne minimale devient :

$$L^{\min}(C) = \sum_{i=1}^{n} p_i \, \ell_\mathrm{i}^{\min} = -\sum_{i=1}^{n} p_i \log_D p_i = H_D(X)$$

On remarque donc que l'entropie est la valeur minimale de la longueur moyenne, ce que nous verrons au point suivant.

3.5 Borne inférieure de L(C)

3.5.1 Théorème

La longueur moyenne de tout code instantané L(C) d'alphabet de taille D pour la source X est supérieure ou égale à l'entropie de la source :

$$\boxed{L(C) \geq H_D(X)} \tag{3.5}$$

Démonstration

Prenons la différence entre la longueur moyenne et l'entropie, et utilisons leur définition (1.4) et (3.1) :

$$L\,(C) - H_D\,(X) = \sum_{i=1}^{n} p_i\,\ell_i + \sum_{i=1}^{n} p_i\,\log_D p_i$$

Insérons ensuite la propriété (3.3) dans l'égalité (3.4), ce qui donne : $\ell_i = \log_D\, D^{\ell_i} = -\log_D\, D^{-\ell_i}$. En posant cela dans l'égalité ci-dessus, on obtient :

$$L\,(C) - H_D\,(X) = -\sum_{i=1}^{n} p_i \log_D\, D^{-\ell_i} + \sum_{i=1}^{n} p_i\,\log_D p_i$$

Afin d'obtenir une distribution de probabilité dont la somme est unitaire, on va poser la distribution normée suivante :

$$q_i = \frac{D^{-\ell_i}}{\sum\limits_{i=1}^{n} D^{-\ell_i}} = \frac{D^{-\ell_i}}{C}$$

où on pose que $\sum\limits_{i=1}^{n} D^{-\ell_i} = C$

Ainsi, on a :

$$L\,(C) - H_D\,(X) = -\sum_{i=1}^{n} p_i \log_D\,(C\,q_i) + \sum_{i=1}^{n} p_i\,\log_D p_i$$

et si on utilise la propriété des logarithmes (1.7), on peut modifier la première somme :

$$L\,(C) - H_D\,(X) = -\sum_{i=1}^{n} p_i \log_D\, C - \sum_{i=1}^{n} p_i \log_D\, q_i + \sum_{i=1}^{n} p_i\,\log_D p_i$$

Rassemblons ensuite les deux derniers termes par la même propriété :

$$L\,(C) - H_D\,(X) = -\sum_{i=1}^{n} p_i \log_D\, C + \sum_{i=1}^{n} p_i\,\log_D \frac{p_i}{q_i}$$

Le terme $\log_D C$ dans le premier terme peut sortir de la somme puisqu'il est constant, en laissant une somme sur les p_i qui est unitaire. Remarquons également que le deuxième terme correspond à la définition de l'entropie relative (1.19), c'est-à-dire :

$$L(C) - H_D(X) = -\log_D C + D(p||q) \tag{3.6}$$

Remarquons que, comme $C = \sum_{i=1}^{n} D^{-\ell_i}$, C est une grandeur inférieure à 1 étant donné l'inégalité de Kraft (3.2). Le logarithme de C est donc négatif, précédé d'un moins. On a vu également que l'entropie relative était une grandeur positive (1.22). Ceci implique que $-\log_D C$ et $D(p||q)$ sont positifs, et que donc

$$L(C) - H_D(X) \geq 0$$

c'est-à-dire :

$$L(C) \geq H_D(X)$$

3.5.2 Distribution D-ADIC et saturation de l'inégalité

Pour saturer l'égalité $L(C) \geq H_D(X)$, et avoir $L(C) = H_D(X)$, il faut, d'après légalité (3.6), que :

$$\begin{cases} \log_D C = 0 \\ D(p||q) = 0 \end{cases}$$

ce qui n'arrive que si :

$$\begin{cases} C = 1 \\ p_i = q_i = \frac{D^{-\ell_i}}{C} \end{cases}$$

Pour que la première égalité soit vérifiée, il faut que l'inégalité de Kraft soit saturée. Pour la deuxième égalité, comme on a posé que $C = 1$, on a que $p_i = D^{-\ell_i}$. Une distribution telle est appelée distribution D-ADIC, distribution dans laquelle p_i est une puissance de 1/D.
Pour ces distributions et pour une inégalité de Kraft saturée, le codage est donc optimal. Ainsi, en théorie, pour trouver un code optimal de manière théorique, il faut :

- trouver une distribution q_i " proche " (au sens de l'entropie relative D) du p_i D-ADIC
- choisir des mots-codes de longueur $\ell_i^{\min} = -\log_D q_i$ entière

C'est ce que fait le code de Shannon du paragraphe suivant.

3.6 Code de Shannon

3.6.1 Définition

Le code de Shannon consiste à poser les longueurs de codages optimales vues au paragraphe précédent, desquelles on prend le pallier afin d'avoir des longueurs entières :

$$\ell_S = -\lceil \log_D q_i \rceil$$

où $x \leq \lceil x \rceil \leq x + 1$. On a alors pour propriété fondamentale suivante :

$$\boxed{H_D(X) \leq L_{Shannon} \leq H_D(X) + 1} \tag{3.7}$$

Il est donc possible d'approcher la borne inférieure de L(C) à un bit près, et ce, sans recourir à l'extension de la source (c'est-à-dire sans avoir des séquences de n bits avec n très grand).

Démonstration

Commençons par montrer que ce code existe pour toute source. On a $x \leq \lceil x \rceil \leq x+1$, c'est-à-dire, en prenant la première inégalité :

$$D^{-\lceil x \rceil} \leq D^{-x}$$

En remplaçant x par $\log_D \frac{1}{p_i}$ et en sommant le tout, on obtient :

$$\sum_{i=1}^{m} D^{-\left\lceil \log_D \frac{1}{p_i} \right\rceil} \leq \sum_{i=1}^{m} D^{-\log_D \frac{1}{p_i}} = \sum_{i=1}^{m} p_i = 1$$

Ainsi, on voit par cette dernière égalité que l'inégalité de Kraft est bien vérifiée. Ceci montre donc que le code (instantané) existe pour toute source.

Démonstration

Partons à nouveau de $x \leq \lceil x \rceil \leq x+1$ et remplaçons x par $-\log_D p_i$:

$$-\log_D p_i \leq \lceil -\log_D p_i \rceil \leq -\log_D p_i + 1$$

On sait que $\ell_i = \lceil -\log_D p_i \rceil$, donc :

$$-\log_D p_i \leq \ell_i \leq -\log_D p_i + 1$$

Multiplions ensuite les trois membres par p_i :

$$-p_i \log_D p_i \leq p_i \ell_i \leq -p_i \log_D p_i + p_i$$

Sommons ensuite les termes de ces trois membres sur l'ensemble des i :

$$-\sum_{i=1}^{m} p_i \log_D p_i \leq \sum_{i=1}^{m} p_i \ell_i \leq -\sum_{i=1}^{m} p_i \log_D p_i + \sum_{i=1}^{m} p_i$$

En utilisant la définition de l'entropie (1.4) et de la longueur moyenne (3.1), et en sachant que la somme des probabilité vaut 1, on obtient finalement :

$$H_D(X) \leq L_{Shannon} \leq H_D(X) + 1$$

Remarquons que la longueur optimale L* du code C vérifie également la propriété fondamentale ci-dessus : $H_D(X) \leq L^* \leq H_D(X) + 1$. En effet, on sait que cette longueur est inférieure ou égale à la longueur de Shannon, et qu'elle est aussi supérieure ou égale à l'entropie. On a donc au final :

$$H_D(X) \leq L^* \leq L_{Shannon} \leq H_D + 1$$

3.6.2 Premier théorème de Shannon (codage par blocs)

On a vu au paragraphe précédent qu'on perdait au pire 1 bit de codage par rapport au cas optimal où $H_D(X) = L_{Shannon}$. On va ici tenter de répartir cette perte de 1 bit sur n symboles $(X_1, ..., X_n) \in X^n$ (où les X_i sont indépendantes et identiquement distribuées, IID), en s'intéressant cette fois-ci à l'extension de la source. Le premier théorème de Shannon exprime qu'il existe un code de compression tel que la longueur moyenne des mots-codes, donnée par

$$L_n = \frac{\sum p(x_1, ...x_n)\ell(x_1, ..., x_n)}{n} = \frac{E[\ell(x_1, ...x_n)]}{n} \tag{3.8}$$

soit arbitrairement proche de H lorsque n tend vers l'infini. On aurait alors en effet :

$$\boxed{H\left(X\right) \leq L_n \leq H\left(X\right) + \frac{1}{n}} \tag{3.9}$$

Démonstration

La propriété fondamentale (3.7) vue au point précédent se généralise à toute extension de la source $(x_1, ...x_n)$. Notons $H_D\left(X_1\ ,\ ...\ ,\ X_n\right)$ et $L\left(X_1\ ,\ ...\ ,\ X_n\right)$ l'entropie et la longueur moyenne de cette extension. En généralisant, on obtient :

$$H_D\left(X_1\ ,\ ...\ ,\ X_n\right) \leq L\left(X_1\ ,\ ...\ ,\ X_n\right) \leq H_D\left(X_1\ ,\ ...\ ,\ X_n\right) + 1 \tag{3.10}$$

avec, par définition, $L\left(X_1\ ,\ ...\ ,\ X_n\right) = \sum p\left(x_1\ ,\ ...\ ,\ x_n\right) \ell\left(x_1\ ,\ ...\ ,\ x_n\right) = \ E\left[\ell\left(x_1\ ,\ ...\ ,\ x_n\right)\right]$
Par la définition (3.8) de la longueur moyenne, on a donc : $L\left(X_1\ ,\ ...\ ,\ X_n\right) = nL_n$

Comme les variables X_i sont indépendantes et identiquement distribuées (IID), on peut écrire :

$$H_D\left(X_1\ ,\ ...\ ,\ X_n\right) = H_D\left(X_1\right) + H_D\left(X_2\right) + ... + H_D\left(X_n\right) = \sum_{i=1}^{n} H_D\left(X_i\right)$$

car les variables sont indépendantes. Comme elles sont identiquement distribuées, on a :

$$H_D\left(X_1\ ,\ ...\ ,\ X_n\right) = \sum_{i=1}^{n} H_D\left(X_i\right) = nH_D\left(X\right)$$

En plaçant ces définition de l'entropie et de la longueur moyenne dans la propriété fondamentale généralisée (3.10), on obtient :

$$nH_D\left(X\right) \leq n\ L_n \leq nH_D\left(X\right) + 1$$

En divisant le tout par n, on trouve finalement :

$$H_D\left(X\right) \leq L_n \leq H_D\left(X\right) + \frac{1}{n}$$

Ainsi, si n tend vers l'infini, alors L_n tend vers $H(X)$.

Ce théorème peut se généraliser à des variables non indépendantes et identiquement distribuées, ou des distributions inconnues :

- Les variables ne sont pas indépendantes et identiquement distribuées :
  On peut reprendre la propriété fondamentale généralisée (3.10) mais on ne peut plus remplacer l'entropie par ce que nous avons trouvé ci-dessus :

$$H_D(X_1\ ,\ \ldots\ ,\ X_n) \leq n\, L_n \leq H_D(X_1\ ,\ \ldots\ ,\ X_n) + 1$$

  Ainsi, en divisant par n, on n'obtient plus le même résultat (3.9) que précédemment. On va donc introduire la notion de taux entropique :

$$H(\mathcal{X}) = \frac{H_D(X_1\ ,\ \ldots\ ,\ X_n)}{n} \tag{3.11}$$

  On généralise donc le premier théorème de Shannon aux X_i non IID :

$$H(\mathcal{X}) \leq L_n \leq H(\mathcal{X}) + \frac{1}{n} \tag{3.12}$$

- La distribution est inconnue :
  Soit p la distribution de la source et q une estimation de p. En tentant de créer un code de Shannon basé sur q en posant

$$\ell(x) = \left\lceil \log_D \frac{1}{q(x)} \right\rceil$$

  le théorème se généralise en :

$$H\left(X_{p(x)}\right) + D(p||q) \leq L_n\left(X_{p(x)}\right) \leq H\left(X_{p(x)}\right) + D(p||q) + 1$$

Démonstration

Prenons la propriété du pallier $x \leq \lceil x \rceil \leq x + 1$, et remplaçons x par $\log_D \frac{1}{q(x)}$:

$$\log_D \frac{1}{q(x)} \leq \left\lceil \log_D \frac{1}{q(x)} \right\rceil \leq \log_D \frac{1}{q(x)} + 1$$

avec $\ell(x) = \left\lceil \log_D \frac{1}{q(x)} \right\rceil$. On a donc :

$$\log_D \frac{1}{q(x)} \leq \ell(x) \leq \log_D \frac{1}{q(x)} + 1$$

En multipliant les trois membres par $p(x)$ et en les sommant sur l'ensemble des x, on obtient :

$$\sum_{x\in\mathcal{X}} p(x) \log_D \frac{1}{q(x)} \leq \sum_{x\in\mathcal{X}} p(x)\,\ell(x) \leq \sum_{x\in\mathcal{X}} p(x) \log_D \frac{1}{q(x)} + \underbrace{\sum_{x\in\mathcal{X}} p(x)}_{1}$$

On va ici utiliser un artifice mathématique en multipliant certains termes par $p(x)/p(x)$:

$$\sum_{x\in\mathcal{X}} p(x) \log_D \frac{1}{q(x)}\frac{p(x)}{p(x)} \leq \sum_{x\in\mathcal{X}} p(x)\,\ell(x) \leq \sum_{x\in\mathcal{X}} p(x) \log_D \frac{1}{q(x)}\frac{p(x)}{p(x)} + 1$$

Ensuite, servons-nous de la propriété des logarithmes (1.7) :

$$\sum_{x\in\mathcal{X}} p(x)\log_D\frac{1}{p(x)}+\sum_{x\in\mathcal{X}} p(x)\log_D\frac{p(x)}{q(x)} \leq \sum_{x\in\mathcal{X}} p(x)\,\ell(x) \leq \sum_{x\in\mathcal{X}} p(x)\log_D\frac{1}{p(x)}+\sum_{x\in\mathcal{X}} p(x)\log_D\frac{p(x)}{q(x)}+1$$

Par les définitions (1.4), (1.19) et (3.1), on peut poser les éléments suivants :

$$\begin{cases} H\left(X_{p(x)}\right) = \sum\limits_{x\in\mathcal{X}} p(x)\log_D\frac{1}{p(x)} \\ D(p||q) = \sum\limits_{x\in\mathcal{X}} p(x)\log_D\frac{p(x)}{q(x)} \\ L(X) = \sum\limits_{x\in\mathcal{X}} p(x)\,\ell(x) \end{cases}$$

Ce qui donne finalement :

$$H\left(X_{p(x)}\right)+D(p||q) \leq L_n\left(X_{p(x)}\right) \leq H\left(X_{p(x)}\right)+D(p||q)+1$$

On a ici une nouvelle interprétation de ce qu'est l'entropie relative : il s'agit bien du nombre de bits perdus par rapport à une longueur moyenne optimale, si on avait utilisé la distribution $p(x)$ au lieu de $q(x)$.

Le codage de Shannon ne peut cependant pas nous permettre de descendre sous l'entropie H. Est-ce qu'on aurait un meilleur résultat avec les codes DFU ? Non, car le théorème de Mac-Millan stipule que ceux-ci satisfont l'inégalité de Kraft, et se doivent donc de respecter également la propriété fondamentale (3.7) vue précédemment. En pratique, le code de Shannon est optimal mais difficile à mettre en place : on utilise donc plutôt le code de Huffman ci-dessous.

3.7 Code de Huffman

3.7.1 Définition

Le principe du codage est de regrouper progressivement les symboles de la source en fonction de leur probabilité jusqu'à obtenir un ensemble contenant tous les symboles qui correspondent à la racine de l'arbre. Voyons cette méthode sur plusieurs exemples :

- Code binaire
  Le principe est le suivant : on trie les symboles de la probabilité la plus haute à la plus basse, et on regroupe les deux derniers en une paire de symboles (dont la probabilité est l'addition des probabilités des symboles groupés). On effectue ceci, en attribuant un bit à chaque symbole regroupé, jusqu'à arriver à deux groupes de symboles.

  On retrouve alors le codage de chaque symbole en reparcourant l'arbre de droite à gauche. On a la longueur moyenne et l'entropie suivantes :

$$L_{Huffman} = \sum p_i\,\ell_i = 2\,.\,0,25+2\,.\,0,25+2\,.\,0,20+3\,.\,0,15+3\,.\,0,15 = 2,3\ bits$$
$$H(X) = -\sum p_i\,\log_2 p_i = -2\left[0,25\log_2(0,25)\right]-0,2\log_2(0,2)-2\left[0,15\log_2(0,15)\right] = 2,2855\ bits$$

  Si on veut se rapprocher davantage de H, il faut utiliser un codage par blocs de Shannon.

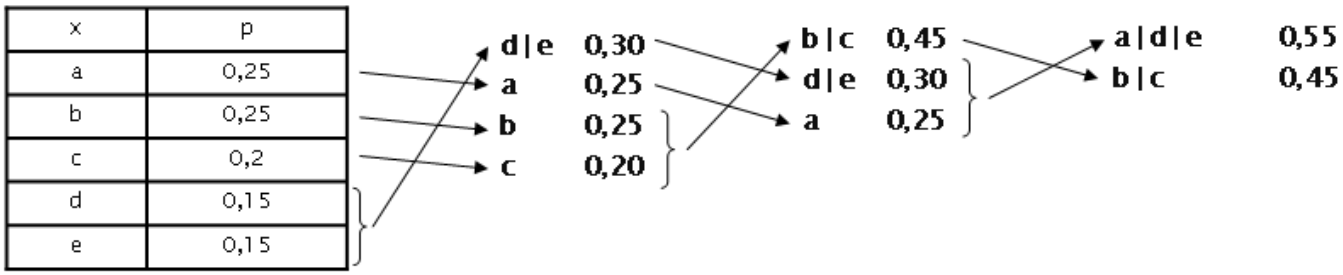

FIG. 3.5 – Exemple de code de Huffman : décomposition

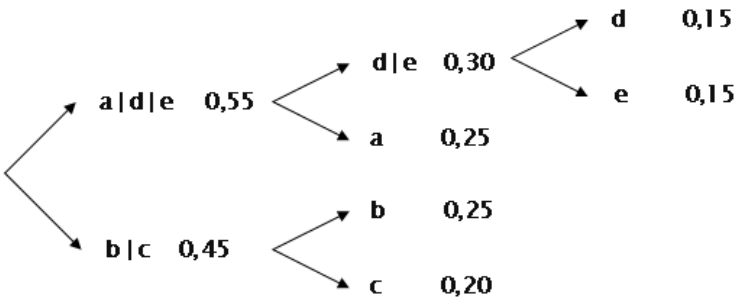

FIG. 3.6 – Exemple de code de Huffman : arbre des symboles

- Code avec D>2
  Pour qu'on arrive à la fin de l'arbre avec un nombre D d'éléments, il faut que le nombre de symboles M soit tel que
  $$M = k(D-1)+1$$
  Si ce n'est pas le cas, on va arriver au bout de l'arbre avec trop peu de symboles à coder. L'astuce est alors d'ajouter des symboles bidons, de probabilité nulle, afin de respecter l'égalité ci-dessus. Ces symboles ne seront bien sûr pas codés.

3.7.2 Optimalité du code de Huffman

Tout code DFU construit avec le même alphabet ne peut avoir une longueur moyenne inférieure à celle du code de Huffman :

$$\boxed{L(C_{Huffman}) \leq L(C)} \tag{3.13}$$

Un code optimal n'est cependant pas unique : on ne change rien en permutant les 1 et les 0, ou bien en échangeant les mots-code de même longueur. Les codes optimaux ont cependant pour propriétés :

- Si, pour deux mots-codes j et k, $p_j > p_k$ alors $\ell_j \leq \ell_k$

Démonstration

Soit C_m un code optimal et C'_m le même code où on a permuté les mots-codes j et k : j prend le code, et donc la longueur de k, et inversemment (la longueur moyenne est donc plus grande car le code est moins optimal). On a donc :

$$L(C'_m) \leq L(C_m)$$

c'est-à-dire :

$$0 \leq L(C'_m) - L(C_m)$$

En utilisant la définition de la longueur moyenne (3.1), on peut développer cette inégalité :

$$0 \leq L\left(C'_m\right) - L\left(C_m\right) = \sum p_i \ell'_i - \sum p_i \ell_i$$

Parmi tous les termes de ces deux sommes, il n'y a que les termes j et k qui diffèrent, si bien que les autres s'annullent :

$$0 \leq \sum p_i \ell'_i - \sum p_i \ell_i = \underbrace{\sum_{\substack{i \neq j \\ i \neq k}} p_i \ell'_i - \sum_{\substack{i \neq j \\ i \neq k}} p_i \ell_i}_{=0} + p_j \ell_k + p_k \ell_j - p_j \ell_j - p_k \ell_k$$

En mettant en évidence les termes restants, on obtient :

$$0 \leq p_j \ell_k + p_k \ell_j - p_j \ell_j - p_k \ell_k = \left(p_j - p_k\right)\left(\ell_k - \ell_j\right)$$

Comme on pose comme hypothèse que $p_j > p_k$, cela veut dire que $p_j - p_k > 0$, et que donc on doit avoir $\ell_k - \ell_j \geq 0$, donc :

$$\ell_k \geq \ell_j$$

- Les D mots-code les plus longs ont la même longueur.
  Si ce n'était pas le cas, alors une des deux (D=2) feuilles de profondeur maximum de l'arbre incomplet ne serait pas associé à un symbole. On pourrait alors attribuer le symbole source au nœud du père, ce qui réduirait la longueur du code : cela signifie bien que l'on n'avait pas un code optimal à la base.
- Ces mots-code diffèrent seulement par leur dernier D-it.

3.7.3 Remarques

- L'arbre canonique peut être vu comme étant une suite de questions oui/non, permettant de " deviner " quel est le symbole qui a été envoyé sur un canal.
  Pour l'arbre ci-dessous, la suite de questions serait :
  "Est-ce un C ?" Si la réponse est oui, c'est un C, sinon on se demande : "est-ce un B ?". Si la réponse est oui, c'est un B, sinon c'est un A.
- Malgré le fait que certains symboles puissent avoir des codages plus courts dans le code de Shannon, la longueur moyenne du codage de Shannon sera toujours supérieure à celle de Huffman :

$$\boxed{H(X) \leq L_H \leq L_S \leq H(X) + 1} \tag{3.14}$$

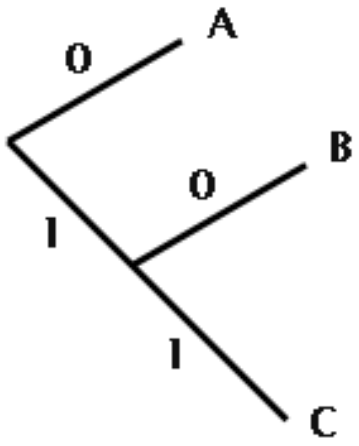

FIG. 3.7 – Exemple d'arbre canonique

Chapitre 4

Capacité d'un canal

4.1 Schéma de principe

La transmission de M messages différents $W = \{1, ..., M\}$ à travers un canal se fait via un encodeur qui crée des mots-code de n bits X^n (séquences d'inputs) qui sortent bruités en tant que séquences d'output Y^n. Un décodeur permet alors de donner une estimation $\widehat{W}$ du message envoyé W.

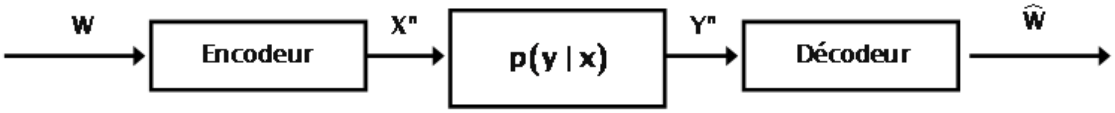

FIG. 4.1 – Schéma de principe d'encodage/décodage

Le canal est caractérisé par une probabilité conditionnelle $p(y|x)$, c'est-à-dire la probabilité d'avoir y à la sortie sachant qu'on a x à l'entrée. Il y a ici un problème de recouvrement, c'est-à-dire des mots-codes qui ont été encodés avec des bits en commun, et qui peuvent donc être éventuellement confondus à la sortie à cause du bruit. Par exemple, à la figure 4.2, on a une information commune entre W=5 et W=4, due au fait qu'on pourrait par exemple les encoder comme 5=101 et 4=100 (seul le dernier bit est différent). Pour éviter ce recouvrement, on va s'arranger pour que la probabilité d'avoir une sortie correspondant à l'entrée soit proche de 1.
Le principe de la construction des mots-code correcteurs d'erreurs est donc de construire des séquences X^n particulières " écartées " les unes des autres, afin d'éviter au maximum qu'elles se recouvrent.

4.2 Définitions

4.2.1 Canal discret sans mémoire

On dit qu'un canal discret n'a pas de mémoire lorsque la probabilité d'avoir une certaine séquence à la sortie pour une certaine séquence d'entrée est le produit des probabilités bit par bit :

$$\boxed{p\left(y^n|x^n\right) = \prod_{i=1}^{n} p\left(y_i|x_i\right)} \tag{4.1}$$

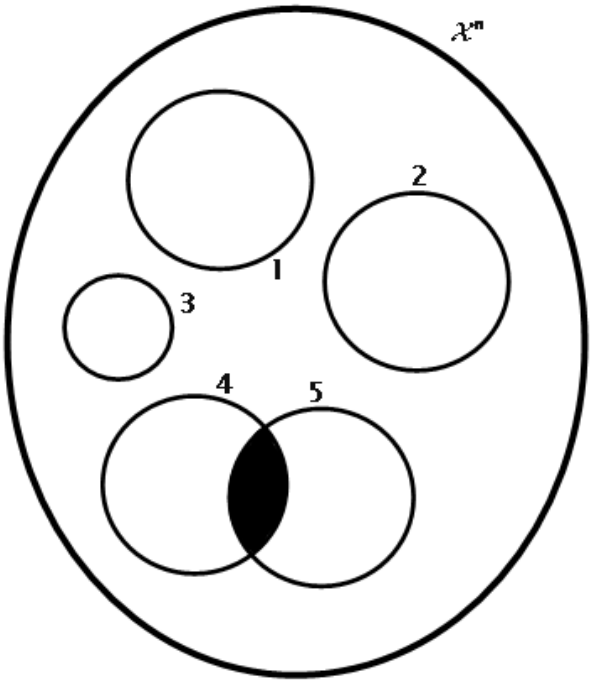

FIG. 4.2 – Recouvrement d'ensembles

4.2.2 Capacité d'un canal discret sans mémoire

Il existe deux définitions de la capacité :

- Capacité informationnelle :

$$\boxed{C_{\text{inf}} = \max_{p(x)} I(X:Y)} \tag{4.2}$$

- Capacité opérationnelle :
  Il s'agit du taux de transfert maximal pour lequel la probabilité d'erreur est inférieure à une certaine tolérance ε.

$$\boxed{C_{op} = \max_{p_e<\varepsilon} R} \tag{4.3}$$

avec le taux de transmission R défini par :

$$\boxed{R = \frac{\text{bits reçus}}{\text{bits transmis}}} \tag{4.4}$$

Le second théorème de Shannon stipule que ces deux capacités sont égales, comme on le verra par la suite.

4.2.3 Exemples de canaux discrets sans mémoire

- Canal binaire sans bruit

  Le principe du canal binaire sans bruit est représenté sur le schéma ci-dessous : tout bit est parfaitement transmis à la sortie du canal.

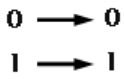

FIG. 4.3 – Canal binaire sans bruit

  - Capacité opérationnelle : comme il n'y a pas d'erreur possible, on a besoin de transmettre un seul bit pour recevoir un bit : $C_{op} = 1$ bit/utilisation du canal
  - Capacité informationnelle : en utilisant la définition de la capacité informationnelle (4.2) et la définition de l'entropie mutuelle (1.16), on a :

$$C_{\inf} = \max_{p(x)} [I(X:Y)] = \max_{p(x)} [H(X) - H(Y|X)]$$

    D'après la définition du canal binaire sans bruit, si on connait X (l'entrée), on connait automatiquement Y (la sortie). Ainsi, il n'existe aucune incertitude sur Y lorsque l'on connait X, et $H(Y|X) = 0$. On a donc :

$$C_{\inf} = \max_{p(x)} H(X)$$

    On a vu précédemment que H(X) est une fonction concave dont la valeur s'annulle en $p(x) = 0$ et $p(x) = 1$, et est maximale en $p(x) = \frac{1}{2}$. En effet, on a vu que l'entropie était maximale lorsque la distribution était égale à 1 divisé par la taille de l'ensemble $\mathcal{X}$, ici égal à 2 puisque $\mathcal{X} = \{0, 1\}$
    L'entropie $H(X)$ est donc maximale pour $p(x) = \{\frac{1}{2}, \frac{1}{2}\}$, et vaut alors 1, tout comme la capacité opérationnelle :

$$C_{\inf} = \max_{p(x)} [H(X)] = 1 \ \textit{bit/utilisation du canal}$$

- Canal bruyant sans recouvrement
  Ici, on obtient des sorties faussées mais on sait toujours à quelle entrée elles correspondent. Les capacités C_{inf} et C_{op} sont identiques au cas précédent.

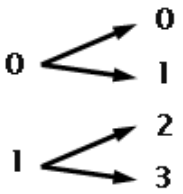

FIG. 4.4 – Canal binaire sans recouvrement

- Canal bruyant avec recouvrement
  Ici, une entrée peut correspondre à plusieurs sorties différentes (comme le cas précédent), mais

une sortie peut également correspondre à plusieurs entrées. Cette source d'erreur s'appelle le recouvrement.

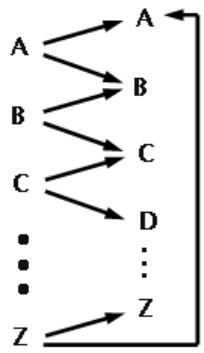

FIG. 4.5 – Canal binaire avec recouvrement

– Capacité opérationnelle : pour éviter le recouvrement, il est nécessaire de prendre une lettre sur deux. On voit alors ci-dessous qu'il n'est plus possible de confondre deux entrées différentes.

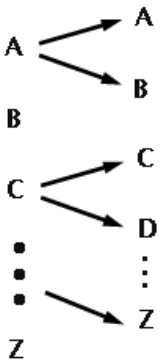

FIG. 4.6 – Canal binaire avec recouvrement : une entrée prise sur deux

On a donc un nombre de mots-codes égal à 13 (la moitié de l'alphabet). Ainsi, pour transmettre un symbole (bits transmis : 1), on doit recevoir k symboles (bits reçus). Pour recevoir $M = 13$ symboles diffŕents, on a en effet besoin de k bits tels que $M = 2^k$, c'est-à-dire $k = \log_2 M$ bits. On a donc un taux de transmission

$$R = \frac{\text{bits reçus}}{\text{bits transmis}} = \frac{k}{1} = \log_2 M = \log_2 13 \; bits$$

Ainsi,

$$C_{op} = \log_2 13$$

– Capacité informationnelle : Calculons d'abord l'entropie mutuelle :

$$I(X:Y) = H(X) - H(Y|X)$$

En utilisant les définitions (1.4) et (1.11), on obtient :

$$I(X:Y) = -\sum_{y\in\mathcal{Y}} p(y)\log_2 p(y) + \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log_2 p(y|x)$$

Puisque les variables sont IID, cela veut dire qu'elles sont identiquement distribuées ($p(y) = \frac{1}{26}$) et indépendantes ($p(x,y) = p(x)p(y) = \frac{1}{26^2}$).
Etant donné la distribution représentée sur le schéma ci-dessus du canal bruyant avec recouvrement, on sait aussi que $p(y|x) = \frac{1}{2}$, ce qui donne :

$$I(X:Y) = -\sum_{y\in\mathcal{Y}} \frac{1}{26}\log_2\frac{1}{26} + \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} \frac{1}{26}\log_2\frac{1}{2}$$

Donc :

$$I(X:Y) = \underbrace{\log_2 26}_{\log_2(2.13)=1+\log_2 13} - \frac{1}{26^2}\underbrace{\log_2 2}_{1}\underbrace{\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} 1}_{26^2} = \log_2 13$$

Ce qui donne finalement :

$$C_{\text{inf}} = \max_{p(x)}\left[I(X:Y)\right] = \log_2 13$$

- Canal binaire symétrique (CBS)
On a ici une probabilité p d'avoir un bit inversé à la sortie. La matrice de transition du canal est donc :

$$p(y|x) = \begin{array}{c} \\ 1-p \\ p \end{array}\begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix} \end{array}\begin{array}{c} \\ 0 \\ 1 \end{array}$$

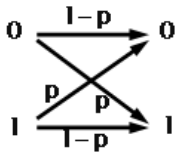

FIG. 4.7 – Schéma d'un canal binaire symétrique (CBS)

On a que $p(y^n|x^n) > 0$, mais on peut se rapprocher de 0 en jouant sur les probabilités. Par exemple, en utilisant le code à répétition (on envoi des séquences de n fois le même bit pour transmettre un seul bit d'information). Ainsi, si on veut transmettre un 0, on aura $\underbrace{00....0}_{n}$ et la probabilité d'erreur, c'est-à-dire d'avoir un 1 à la sortie $\underbrace{11....1}_{n}$ est donnée par p^n, qui tend vers 0 lorsque n tend vers l'infini. On a donc une probabilité d'erreur nulle, mais un taux de transmission affaibli :

$$R = \frac{\text{bits reçus}}{\text{bits transmis}} = \frac{1}{n}$$

Calculons la capacité informationnelle, en commençant par calculer l'entropie mutuelle :

$$I(X:Y) = H(Y) - H(Y|X)$$

D'après la définition de l'entropie conditionnelle (1.11), on a que :

$$I(X:Y) = H(Y) - H(Y|X) = H(Y) - \sum_{x\in\{0,1\}} p(x)\,\mathrm{H}(\mathrm{Y}|\mathrm{X}=\mathrm{x})$$

Ainsi, si on a 1 à l'entrée, on peut avoir à la sortie 0 avec une probabilité p ou 1 avec une probabilité $1-p$. Cette incertitude est représentée par $H_2(p, 1-p)$. On note donc :

$$\mathrm{H}(\mathrm{Y}|\mathrm{X}=\mathrm{x}) = \mathrm{H}_2(\mathrm{p}, 1-\mathrm{p})$$

Ce qui nous donne :

$$I(X:Y) = H(Y) - \mathrm{H}_2(\mathrm{p}, 1-\mathrm{p}) \underbrace{\sum_{x\in\{0,1\}} p(x)}_{1}$$

Etant donné que le maximum de l'entropie est donné par (1.5), on sait que $H(Y) \leq \log_2 2 = 1$, et donc :

$$I(X:Y) = H(Y) - \mathrm{H}_2(\mathrm{p}, 1-\mathrm{p}) \leq 1 - \mathrm{H}_2(\mathrm{p}, 1-\mathrm{p})$$

Ce qui nous donne finalement :

$$C_{\inf} = \max_{p(x)} I(X:Y) = 1 - \mathrm{H}_2(\mathrm{p})$$

et donc :

$$\boxed{C_{CBS} = 1 - \mathrm{H}_2(\mathrm{p})} \tag{4.5}$$

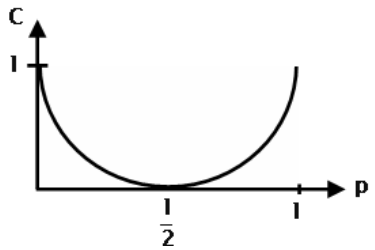

FIG. 4.8 – Graphique de la capacité d'un canal binaire symétrique (CBS)

- Canal binaire à effacement (CBE)
  On a ici une probabilité π d'avoir 1 et $1-\pi$ d'avoir 0, et chacun a une probabilité α d'être effacé le long du canal, et une chance $1-\alpha$ d'être correctement transmis sans inversion. Les sources d'erreurs sont donc localisées : elles proviennent des cas où on n'obtient aucun bit à la sortie. Calculons la capacité informationnelle, en calculant d'abord l'entropie mutuelle :

$$I(X:Y) = H(Y) - H(Y|X)$$

Etant donné que l'on a affaire à un alphabet ternaire (il existe trois types de sortie : 0, 1 ou "effacé"), on pourrait penser que $H(Y) \leq \log_2 3$, mais ce n'est pas le cas, car il existe plusieurs chemins possibles, représentés par le schéma ci-dessous.

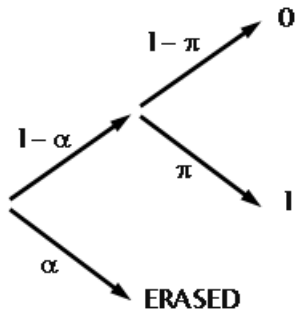

FIG. 4.9 – Diffrentes possibilités dans un CBE

Il faut donc utiliser le théorème de regroupement sur la suite des choix ci-dessus, afin de les prendre tous en compte. L'entropie $H(Y)$ est donc une entropie appliquée à trois choix différents, c'est-à-dire :

- La sortie 0 : il faut d'abord avoir $1-\alpha$ chances de ne pas être supprimé, et ensuite avoir $1-\pi$ chances de valoir 0, donc $p(0) = (1-\pi)(1-\alpha)$
- La sortie 1 : il faut d'abord avoir $1-\alpha$ chances de ne pas être supprimé, et ensuite avoir π chances de valoir 1, donc $p(1) = \pi(1-\alpha)$
- La sortie "erased" : il faut avoir α chances d'être supprimé, donc $p(erased) = \alpha$

On a donc :

$$H(Y) = H_3[(1-\pi)(1-\alpha), \pi(1-\alpha), \alpha]$$

Le théorème de regroupement stipule que l'entropie sur les 3 résultats est la somme des entropies de chaque choix pondérés par les probabilités correspondantes :

$$H\left(Y\right) = H_2\left(\alpha\right) + \left(1-\alpha\right) H_2\left(\pi\right) + \alpha.0$$

Pour le troisième choix, on n'a aucune incertitude : on sait que c'est ERASED, d'où la multiplication par 0. En replaçant dans la définition de l'entropie mutuelle ci-dessus, on obtient :

$$I\left(X:Y\right) = H\left(Y\right) - \mathrm{H}_2\left(\alpha\right) = H_2\left(\alpha\right) + \left(1-\alpha\right) H_2\left(\pi\right) - \mathrm{H}_2\left(\alpha\right) = \left(1-\alpha\right) H_2\left(\pi\right)$$

Si la densité de probabilité $p(x)$ donne $\pi = \frac{1}{2}$, alors la valeur de $H_2(\pi)$ atteint son maximum, à savoir $\log_2 2 = 1$. Ainsi, le maximum de l'entropie mutuelle $I(X:Y)$ ci-dessus est donné par :

$$C_{CBE} = \max_{p(x)} I\left(X:Y\right) = \max_{p(x)} \left[\left(1-\alpha\right) H_2\left(\pi\right)\right] = 1-\alpha$$

et donc :

$$\boxed{C_{CBE} = 1-\alpha} \tag{4.6}$$

Ainsi, la capacité est uniquement diminuée par la présence des symboles perdus : ils n'apportent aucune information.

- Canal binaire à effacement et feedback

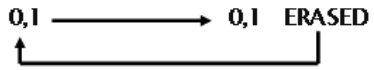

FIG. 4.10 – Principe du canal binaire à effacement avec feedback

On pourrait penser utiliser un CBE avec feedback, c'est-à-dire un CBE qui, une fois un ERASED détecté, réessaye de transmettre le même bit. Tentons de calculer sa capacité en regardant le nombre d'usages du canal nécessaires pour transmettre un bit :

$$N = \underbrace{(1-\alpha)}_{\substack{\text{le bit est}\\ \text{transmis}}} \underbrace{1}_{\substack{\text{un usage}\\ \text{du canal}\\ \text{requis}}} \underbrace{+}_{ou} \underbrace{\alpha}_{\substack{\text{le bit a été}\\ \text{supprimé}\\ \text{une fois}}} \underbrace{(1-\alpha)}_{\substack{\text{le bit est}\\ \text{transmis}}} \underbrace{2}_{\substack{\text{deux usages}\\ \text{du canal ont}\\ \text{été requis}}} \underbrace{+}_{ou} \underbrace{\alpha^2}_{\substack{\text{le bit a été}\\ \text{supprimé}\\ \text{2 fois}}} \underbrace{(1-\alpha)}_{\substack{\text{le bit est}\\ \text{transmis}}} \underbrace{3}_{\substack{\text{trois usages}\\ \text{du canal ont}\\ \text{été requis}}} +...$$

Nous pouvons ensuite généraliser ceci à l'aide de sommes infinies :

$$N = (1-\alpha)\sum_{k=1}^{\infty} k\alpha^{k-1}$$

Il faut ensuite remarquer que $k\alpha^{k-1}$ n'est rien d'autre que la dérivée de α^k. Donc :

$$N = (1-\alpha)\frac{\partial}{\partial\alpha}\left(\sum_{k=1}^{\infty}\alpha^k\right)$$

Utilisons maintenant la propriété suivante :

$$\sum_{k=1}^{\infty}\alpha^k = \frac{\alpha}{1-\alpha}$$

Qui nous permet d'écrire :

$$N = (1-\alpha)\frac{\partial}{\partial\alpha}\left(\frac{\alpha}{1-\alpha}\right) = (1-\alpha)\frac{(1-\alpha)-(-\alpha)}{(1-\alpha)^2} = (1-\alpha)\frac{1}{(1-\alpha)^2} = \frac{1}{1-\alpha}$$

Comme on a besoin de $\frac{1}{1-\alpha}$ usages du canal pour transmettre 1 bit, un seul usage du canal permet de transmettre $1-\alpha$ bits, et donc on peut écrire que $C = 1-\alpha$. On remarque donc que le feedback ne change rien à la capacité du canal :

$$\boxed{C_{feedback} = C_{sans\ feedback}} \tag{4.7}$$

Le schéma ci-dessous reprend les différentes capacités précédemment calculées :

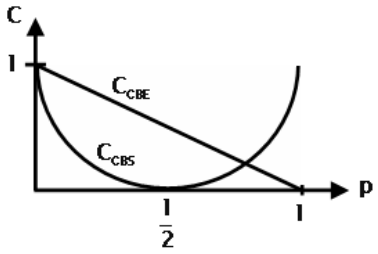

FIG. 4.11 – Capacités de différents canaux

4.3 Propriétés de C

- $C \geq 0$ puisque $I(X:Y) \geq 0$
- $C \leq \log|\mathcal{X}|$ puisque $I(X:Y) = H(X) - \underbrace{H(Y|X)}_{\geq 0} \leq H(X) \leq \log|\mathcal{X}|$
- $C \leq \log|\mathcal{Y}|$ pour la même raison
- $I(X:Y)$ est une fonction continue et concave de p(x) : il ne peut exister plusieurs maxima globaux, ce qui implique qu'il ne peut exister qu'un seul C global, que l'on peut ainsi estimer numériquement (il n'existe pas de solution analytique simple). Ceci est illustré ci-dessous.

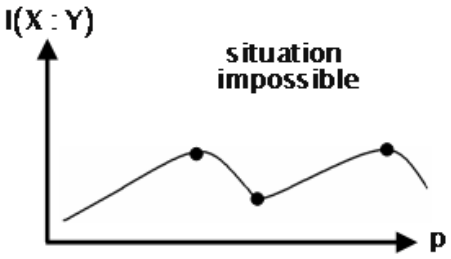

FIG. 4.12 – Cas impossible de plusieurs maxima de I(X :Y)

4.4 Canaux symétriques

Un canal est dit symétrique lorsque les lignes et les colonnes de la matrice de transition sont des permutations les unes des autres. On dit qu'un canal est faiblement symétrique lorsque les colonnes de cette matrice ont la même somme :

$$\sum_x p(y|x) = constante$$

$$p(y|x) = \begin{pmatrix} . & . & . \\ . & . & . \\ . & . & . \end{pmatrix} \leftarrow \vec{r}$$

Pour un canal (faiblement) symétrique, on a la relation

$$\boxed{C_{CS} = \log|\mathcal{Y}| - H(\vec{r})} \tag{4.8}$$

où $\vec{r}$ est n'importe quelle ligne de $p(y|x)$: la capacité est atteinte lorsque la distribution de l'entrée est uniforme.

Démonstration

Partons de la définition de l'entropie mutuelle (1.16) :

$$I(X:Y) = H(Y) - H(Y|X) = H(Y) - \sum_{x\in\mathcal{X}} p(x)\, H(Y|X=x)$$

Etant donné la symétrie de la matrice $p(y|x)$, on peut écrire que :

$$H(Y|X=x) = H(\vec{r}) \quad \forall x$$

qui est une constante, et peut donc sortir de la somme :

$$I(X:Y) = H(Y) - H(\vec{r}) \underbrace{\sum_{x\in\mathcal{X}} p(x)}_{=1}$$

On a vu que le maximum d'une entropie était donné par le logarithme de la taille de l'ensemble considéré : $H(Y) \leq \log|\mathcal{Y}|$. Ainsi, on a :

$$I(X:Y) = H(Y) - H(\vec{r}) \leq \log|\mathcal{Y}| - H(\vec{r})$$

Et donc, comme la capacité est donnée par le maximum de l'entropie mutuelle :

$$C_{CS} = \log|\mathcal{Y}| - H(\vec{r})$$

Ce maximum est atteint si la variable d'entrée est distribuée uniformément, c'est-à-dire si $p(x) = \frac{1}{|\mathcal{X}|}$. En effet, on a alors :

$$p(y) = \sum_x p(x)\, p(y|x) = \frac{1}{|\mathcal{X}|} \sum_x p(y|x)$$

Comme on sait que, d'après la définition du canal symétrique, $\sum_x p(y|x) = \textit{constante}$, on peut dire que $p(y)$ doit être constante :

$$p(y) = \frac{1}{|\mathcal{X}|} \underbrace{\sum_x p(y|x)}_{c^{te}} = \textit{constante}$$

et pour que $p(y)$ soit constante, il faut que la distribution soit uniforme, c'est-à-dire :

$$p(y) = \frac{1}{|\mathcal{Y}|}$$

Chapitre 5

Codage de canal

5.1 Théorème du codage de canal

5.1.1 Définitions

On va ici se servir de l'extension d'ordre n d'un canal en lui fournissant des séquences de n bits. Si n est suffisamment grand, on peut considérer ces séquences comme typiques. On obtient alors à la sortie un ensemble de $2^{nH(Y)}$ séquences typiques différentes, mais dont le volume $2^{nH(Y|X)}$ (non démontré) pourrait coïncider avec celui d'une autre séquence typique.

On définit ainsi le nombre d'ensembles disjoints qu'il est possible de transmettre, en divisant le nombre de séquence par le volume de ces séquences :

$$N \leq \frac{2^{n\,H(Y)}}{2^{n\,H(Y|X)}} = 2^{n\,[H(Y)-H(Y|X)]}$$

Par la définition de l'entropie mutuelle (1.16), on trouve donc que :

$$N \leq 2^{n\,[H(Y)-H(Y|X)]} = 2^{n\,I(X:Y)}$$

et donc :

$$\frac{\log_2 N}{n} \leq I\,(X:Y)$$

où $\frac{\log_2 N}{n}$ est l'information transmise par usage du canal.

Introduisons aussi les notions suivantes :

- Code correcteur d'erreur
  On note un code correcteur d'erreur (M,n), et il est défini par :
  - Un ensemble de messages à envoyer { 1 ... M }
  - Une fonction d'encodage $X^n : \{\, 1...M \,\} \rightarrow X^n\,(m)$
  - Une fonction de décodage : $g : \mathcal{Y}^n \rightarrow \{\, 1...M \,\}$

- Probabilité d'erreur conditionnelle
  La probabilité d'erreur, sachant que i est envoyé, s'écrit :

$$\boxed{\lambda_i = P\,[g\,(\mathcal{Y}^n) \neq i \;\mid\; \mathcal{X}^n = X^n\,(i)]} \tag{5.1}$$

- Probabilité d'erreur moyenne
  Il s'agit de la somme des probabilités d'erreur conditionnelles divisées par le nombre d'erreurs faisables (c'est-à-dire le nombre de messages M à envoyer) :

$$\boxed{P_e^{(n)} = \frac{1}{M}\sum_{i=1}^{M} \lambda_i = P\left[g\left(\mathcal{Y}^n\right) \neq \mathcal{I} \;\mid\; \mathcal{X}^n = X^n\left(\mathcal{I}\right)\right]} \tag{5.2}$$

- Taux de transmission d'un code (M,n)
  Pour envoyer M messages différents, k bits sont nécessaires tels que : $M = 2^k$, c'est-à-dire $k = \log_2 M$. Par contre, on enverra peut-être plus de bits à travers le canal pour améliorer la robustesse du message (voir chapitre 6). Ainsi, si on envoit n bits pour en transmettre k, on aura un taux de transmission donné par :

$$\boxed{R = \frac{k}{n} = \frac{\log M}{n} \quad \frac{\text{bits}}{\text{usage du canal}}} \tag{5.3}$$

  On peut le voir comme étant l'entropie par symbole de canal des messages d'entrée quand ceux-ci sont distribués équiprobablement, c'est-à-dire $H = \log_2 M$. Le taux R est atteignable s'il existe une suite de codes (M,n) avec $M = M(n) = \left\lceil 2^{nR} \right\rceil$ tels que la proabilité d'erreur moyenne s'annulle lorsque n tend vers l'infini : $\lim_{n\to\infty} P_e^{(n)} = 0$

5.1.2 Séquences conjointement typiques

L'ensemble $A_\varepsilon^{(n)}$ est l'ensemble des paires de séquences (X^n, Y^n) conjointement typiques, c'est-à-dire telles que l'entrée soit typique, la sortie soit typique, et que la jointure des deux soit également typique. Une paire de séquences (X^n, Y^n) doit donc respecter trois conditions pour être conjointement typique :

$$A_\varepsilon^{(n)} = \left\{ \; (X^n, Y^n) \in (\mathcal{X}^n, \mathcal{Y}^n) \;\; tq \;\; \left\{ \begin{array}{l} \left| -\frac{1}{n}\log p\left(\mathcal{X}^n\right) - H\left(X\right) \right| \leq \varepsilon \\ \left| -\frac{1}{n}\log p\left(\mathcal{Y}^n\right) - H\left(Y\right) \right| \leq \varepsilon \\ \left| -\frac{1}{n}\log p\left(\mathcal{X}^n, \mathcal{Y}^n\right) - H\left(X,Y\right) \right| \leq \varepsilon \end{array} \right\} \right. \tag{5.4}$$

Les séquences conjointement typiques ont, tout comme les séquences typiques, certaines propriétés :

- Propriété 1
  La probabilité qu'une paire de séquences quelconques soit conjointement typique est proche de 1, lorsque n tend vers l'infini :

$$\boxed{P\left[\left(X^n, Y^n\right) \in A_\varepsilon^{(n)}\right] \overset{n\to\infty}{>} 1 - \varepsilon} \tag{5.5}$$

- Propriété 2
  La taille de l'ensemble des séquences conjointement typiques tend vers $2^{nH(X,Y)}$ lorsque n tend vers l'infini :

$$\boxed{\left| A_\varepsilon^{(n)} \right| \overset{n\to\infty}{\to} 2^{n\,H(X,Y)}} \tag{5.6}$$

- Propriété 3
  Si $\left(\widetilde{X^n}, \widetilde{Y^n}\right)$ est une paire de séquences typiques indépendantes (distribuée comme $p\left(X^n\right) p\left(Y^n\right)$), alors la probabilité que cette paire appartienne à l'ensemble des séquences conjointement typiques tend vers 0 lorsque n tend vers l'infini :

$$\boxed{P\left[\left(\widetilde{X^n}, \widetilde{Y^n}\right) \in A_\varepsilon^{(n)}\right] \overset{n\to\infty}{\to} 2^{-n\, I(X:Y)}} \tag{5.7}$$

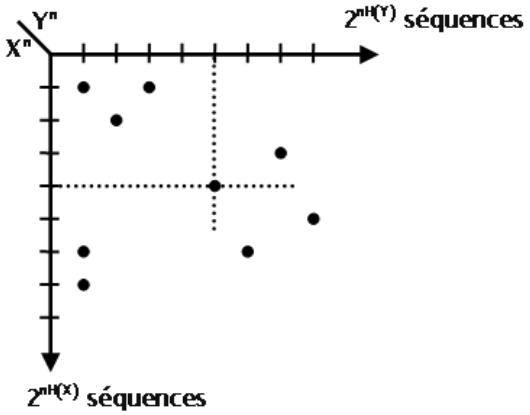

FIG. 5.1 – Ensembles typiques

On peut voir ça graphiquement ci-dessus : on a $2^{nH(X,Y)}$ séquences conjointement typiques, parmi les $2^{nH(X)}2^{H(Y)}$ séquences typiques comprises au total dans le maillage. On a donc, pour deux séquences typiques et indépendantes, la probabilité qu'elles soient conjointement typiques donnée par :

$$P = \frac{\text{cas favorables}}{\text{cas possibles}} = \frac{2^{n\,H(X,Y)}}{2^{n\,H(X)}\,2^{n\,H(Y)}} = 2^{n\,[H(X,Y)-H(X)-H(Y)]} = 2^{-n\,I(X:Y)}$$

Démonstration

Démontrons analytiquement la propriété 3 ci-dessus. Pour cela, on va montrer qu'il existe une borne supérieure et une borne inférieure à la probabilité P, probabilité que deux séquences typiques et indépendantes soient conjointement typiques.
La probabilité P est donnée par la somme des probabilités $p(X^n, Y^n)$ sur l'ensemble des entrées et sorties existantes, à savoir :

$$P = \sum_{(X^n,Y^n)} p\left(X^n, Y^n\right)$$

Comme les variables sont indépendantes, on peut écrire que $p\left(X^n, Y^n\right) = p\left(X^n\right) p\left(Y^n\right)$, et donc :

$$P = \sum_{(X^n,Y^n)} p\left(X^n\right) p\left(Y^n\right)$$

– Par la propriété (2.3) des séquences typiques, on sait que :

$$\begin{cases} p\left(X^n\right) \leq 2^{-n[H(X)-\varepsilon]} \\ p\left(Y^n\right) \leq 2^{-n[H(Y)-\varepsilon]} \end{cases}$$

Ainsi, on peut noter que :

$$P = \sum_{(X^n,Y^n)} p\left(X^n\right) p\left(Y^n\right) \leq 2^{-n[H(X)-\varepsilon]} 2^{-n[H(Y)-\varepsilon]} \underbrace{\sum_{(X^n,Y^n)} 1}_{\left|A_\varepsilon^{(n)}\right|} = 2^{-n[H(X)+H(Y)-2\varepsilon]} \left|A_\varepsilon^{(n)}\right|$$

Par la propriété 2 des ensembles typiques (2.5), on note : $\left|A_\varepsilon^{(n)}\right| \leq 2^{n[H(X,Y)+\varepsilon]}$, et donc :

$$P \leq 2^{-n[\,H(X)+H(Y)-2\varepsilon\,]} \left|A_\varepsilon^{(n)}\right| \leq 2^{-n[\,H(X)+H(Y)-2\varepsilon\,]} 2^{n[H(X,Y)+\varepsilon]} = 2^{-n[\,H(X)+H(Y)-H(X,Y)-3\varepsilon\,]}$$

En utilisant la définition de l'entropie mutuelle (1.16), on obtient finalement :

$$P \leq 2^{-n[\,I(X:Y)-3\varepsilon\,]}$$

Nous trouvons donc la borne supérieure pour P.

– Pour trouver la borne inférieure, repartons de la définition de P et de la définition d'une séquence typique. On sait que :

$$P = \sum_{(X^n,Y^n)} p\left(X^n\right) p\left(Y^n\right)$$

et que :

$$\begin{cases} p\left(X^n\right) \geq 2^{-n[\,H(X)+\varepsilon\,]} \\ p\left(Y^n\right) \geq 2^{-n[\,H(Y)+\varepsilon\,]} \end{cases}$$

donc :

$$P = \sum_{(X^n,Y^n)} p\left(X^n\right) p\left(Y^n\right) \geq 2^{-n[\,H(X)+H(Y)+2\varepsilon\,]} \left|A_\varepsilon^{(n)}\right|$$

Ensuite, on utilise la propriété 3 des séquences typiques (2.6), c'est-à-dire $\left|A_\varepsilon^{(n)}\right| > 2^{n\,(H-\varepsilon)}\,(1-\varepsilon)$. On a donc :

$$P = \sum_{(X^n,Y^n)} p\left(X^n\right) p\left(Y^n\right) \geq (1-\varepsilon)\, 2^{-n[\,H(X)+H(Y)-H(X,Y)+3\varepsilon\,]}$$

et par la définition de l'entropie mutuelle (1.16) :

$$P \geq (1-\varepsilon)\ 2^{-n[\,I(X:Y)+3\varepsilon\,]}$$

Il s'agit de la borne inférieure.

Au final, en combinant ces deux bornes, nous obtenons :

$$2^{-n[\,I(X:Y)+3\varepsilon\,]} \leq P \leq (1-\varepsilon)\ 2^{-n[\,I(X:Y)-3\varepsilon\,]}$$

Ainsi, lorsque n tend vers l'infini, nous avons bien :

$$P \overset{n\to\infty}{\to} 2^{-n\,I(X:Y)}$$

5.2 Second théorème de Shannon

Le théorème s'énonce comme ceci : si $R < C$, alors R est réalisable, c'est-à-dire qu'il existe un code tel que la probabilité d'erreur moyenne est nulle lorsque n tend vers l'infini : $\lim_{n\to\infty} P_e^{(n)} = 0$. Il existe deux réciproques à ce théorème, une réciproque faible et une réciproque forte :

- Réciproque faible : Si $R > C$, alors on a $\lim_{n\to\infty} P_e^{(n)} = p > 0$
- Réciproque forte : Si $R > C$, alors on a $\lim_{n\to\infty} P_e^{(n)} = 1$

Ces deux réciproques sont représentées sur le schéma ci-dessous.

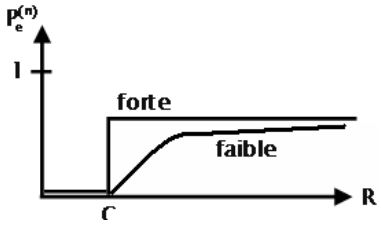

FIG. 5.2 – Représentation graphique des réciproques du second théorème de Shannon

Démonstration

Les hypothèses utilisées pour démontrer ce théorème sont les suivantes :

1. On a uns source dont on encode les messages sous forme de mots-codes X^n de longueur n tels que $R = \frac{\log_2 M}{n}$. Ainsi, on a $M = 2^{nR}$ messages provenant de la source (voir figure 5.3)

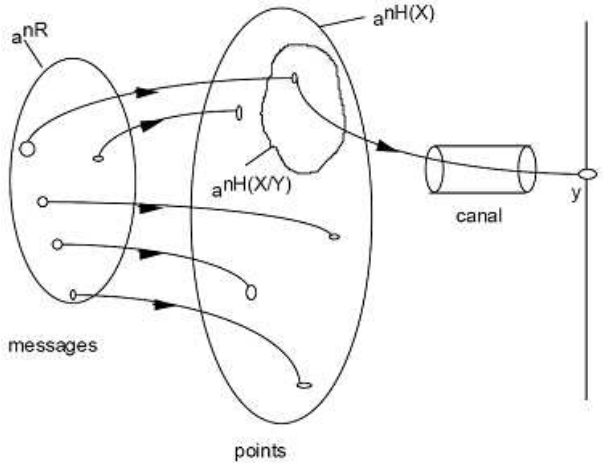

FIG. 5.3 – Cheminement de l'information et encodage

2. On réalise des mots-codes de longueur n suffisamment grande pour qu'on puisse utiliser les propriétés des séquences typiques (et des séquences conjointement typiques, vues précédemment).
3. On utilise un code aléatoire, c'est-à-dire que le code utilisé pour la transmission des messages sur le canal est choisi de façon aléatoire dans une classe de codes, la distribution $p(x)$ de la source étant connue. On calculera alors la probabilité d'erreur moyenne de l'ensemble des codes.

Si cette dernière tend vers 0, alors on conclura que parmi tous les codes de la classe considérée, il en existe au moins un pour lequel la probabilité d'erreur tend aussi vers 0.

4. On décode par "typicalité jointe". Le principe consiste à recevoir un output, et déduire d'après le tableau (de la figure 5.1) quelle est l'entrée conjointement typique qui y correspond. On verra qu'asymptotiquement (n tendant vers l'infini), il n'y a pas moyen de se tromper en utilisant ce procédé.

Quelles sont les sources qui pourraient fausser notre méthode de typicalité jointe ? Imaginons que l'on lise à la sortie du canal la séquence Y^n. On regarde dans le tableau à la colonne correspondante et on remarque un rond qui réfère à un X^n. Le problème est que :

- peut-être X^n et Y^n ne sont pas conjointement typiques. Or, d'après la propriété 1 des séquences conjointement typiques (5.5), on a $P\left[(X^n,Y^n)\in A_\varepsilon^{(n)}\right] \overset{n\to\infty}{>} 1-\varepsilon$, donc la probabilité de se tromper (c'est-à-dire de prendre une paire non conjointement typique) en sélectionnant un point dans le tableau est :

$$P_{e\,1} = 1 - P\left[(X^n,Y^n)\in A_\varepsilon^{(n)}\right] \leq 1-(1-\varepsilon) = \varepsilon \quad \Rightarrow \quad P_{e\,1}\leq\varepsilon$$

- ce Y^n correspond à un X^n lié à plusieurs messages différents (voir figure **??**). Il existe donc des messages "rivaux" qui ne sont pas les entrées conjointement typiques avec Y^n : elles sont donc indépendantes de Y^n, et ne sont que marginalement typiques avec Y^n. Or, on a trouvé, dans la démonstration de la propriété 3, que pour une entrée typique et une sortie typique indépendantes $\left(\widetilde{X^n},\widetilde{Y^n}\right)$, on a :

$$P\left[\left(\widetilde{X^n},\widetilde{Y^n}\right)\in A_\varepsilon^{(n)}\right] \leq (1-\varepsilon)\ 2^{-n[\,I(X:Y)-3\varepsilon\,]} \leq 2^{-n[\,I(X:Y)-3\varepsilon\,]}$$

Comme il y a 2^{nR} messages différents provenant de la source, il y a $2^{nR}-1$ messages rivaux du message correct. Ainsi, la probabilité de se tromper en choisissant un mauvais message plutôt que le bon est donnée par la multiplication de $2^{nR}-1$ (nombre total de rivaux) par la probabilité de choisir un rival, (c'est-à-dire quelque chose d'inférieur à $2^{-n[\,I(X:Y)-3\varepsilon\,]}$). On a donc :

$$P_{e\,2} \leq \left(2^{nR}-1\right)2^{-n[\,I(X:Y)-3\varepsilon\,]} \leq 2^{nR}2^{-n[\,I(X:Y)-3\varepsilon\,]} - 2^{-n[\,I(X:Y)-3\varepsilon\,]} \leq 2^{n[R-I(X:Y)+3\varepsilon]}$$

Au final, la probabilité de se tromper en appariant une sortie Y^n avec une entrée X^n en "décodant par typicalité jointe" est donnée par la somme des deux probabilités d'erreurs P_{e1} et P_{e2} mentionnées ci-dessus :

$$P_e^{(n)} = P_{e\,1} + P_{e\,2}$$

Ayant trouvé précédemment que :

$$\begin{cases} P_{e\,1} \leq \varepsilon \\ P_{e\,2} \leq 2^{n[R-I(X:Y)+3\varepsilon]} \end{cases}$$

on obtient donc :

$$P_e^{(n)} \leq \varepsilon + 2^{n[R-I(X:Y)+3\varepsilon]}$$

Remarquons que le terme $2^{n[R-I(X:Y)+3\varepsilon]}$ peut devenir arbitrairement petit seulement si l'exposant est négatif, c'est-à-dire lorsque $R-I\,(X:Y)+3\varepsilon<0$. On a alors, et seulement dans ce cas-là : $2^{n[R-I(X:Y)+3\varepsilon]}\leq\varepsilon$ (lorsque n tend vers l'infini). Ceci donne au final :

$$P_e^{(n)} \leq \varepsilon + \varepsilon = 2\varepsilon \quad \Leftrightarrow \quad R \ < \ I\left(X:Y\right) - 3\varepsilon$$

Comme la capacité C est le maximum de $I(X:Y)$, la condition devient :

$$R \ < \ I\left(X:Y\right) - 3\varepsilon \leq C - 3\varepsilon$$

Ainsi, si $R \ < C - 3\varepsilon$, alors $P_e^{(n)} \leq 2\varepsilon$. Dans le cas où n tend vers l'infini, alors ε tend vers 0, et on a :

$$P_e^{(n)} = 0 \quad \Leftrightarrow \quad R \ < C$$

Notons que, selon la troisème hypothèse, le résultat ci-dessus nous amène bien à une probabilité d'erreur moyenne nulle pour toute une classe de codes. En effet, cette probabilité est donnée par :

$$P_e^{(n)} = \sum_{\substack{\text{ensemble} \\ \text{des codes de} \\ \text{la classe}}} \underbrace{P\left(C\right)}_{\substack{\text{probabilité d'avoir} \\ \text{le code aléatoire C}}} \underbrace{P_e^{(n)}\left(C\right)}_{\substack{\text{probabilité d'erreur} \\ \text{moyenne sur tous les} \\ \text{mots-codes du code C}}}$$

Comme on utilise des codes aléatoires, ils ont tous la même probabilité d'apparaître, et donc $P(C) = \textit{constante}$. Finalement, on a :

$$P_e^{(n)} = P\left(C\right) \sum_{\substack{\text{ensemble} \\ \text{des codes de} \\ \text{la classe}}} P_e^{(n)}\left(C\right) = 0$$

c'est-à-dire

$$P_e^{(n)}\left(C\right) = 0 \quad \forall C$$

On en conclut que les codes de la classe considérée ont tous une probabilité d'erreur qui tend vers 0. On verra au chapitre 6 que cette classe est la classe des codes correcteurs d'erreurs, dans laquelle on ajoute des redondances aux messages à transmettre pour les rendre plus résistants aux erreurs. Par extension, on déduit qu'il existe un code C dans cette classe tel que $P_e^{(n)}\left(C\right) = 0$, ce qui démontre bien le second théorème de Shannon.

Chapitre 6

Codes correcteurs d'erreur

6.1 Introduction

On va ici suivre le cheminement représenté sur le schéma ci-dessous.

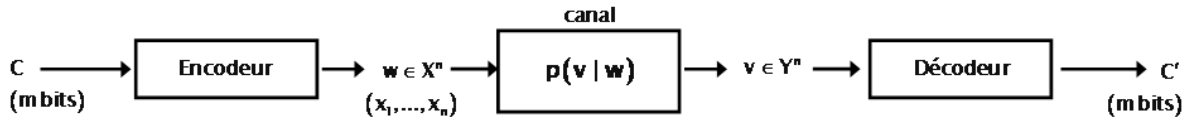

FIG. 6.1 – Encodage et décodage de l'information

Sur ce schéma, on :

- encode tous les symboles de C (de la taille de m bits) sous forme de chaines de n bits, avec en général $m \leq n$. Cet encodage se fait via un code noté code (M,n) qui ajoute donc des redondances pour augmenter la robustesse des messages transmis. L'alphabet C est constitué de M messages différents, tel que $M = 2^m$. On a donc un taux de transmission $R = \frac{\log M}{n} = \frac{m}{n}$, que l'on tente d'avoir proche de la capacité C mais tout en ayant des blocs de taille n pas trop grands.
- décode les messages à la sortie, c'est-à-dire on associe un w à toute séquence v_j possible (il peut en exister 2^n au total).

On fait pour hypothèse que les M mots-code d'entrée sont équiprobables (s'ils ne le sont pas, on compresse) : $p(w_i) = \frac{1}{M}$

6.2 Décodeur idéal

6.2.1 Définition générale

Un décodeur idéal est un décodeur qui associe w à la séquence v tel que la probabilité d'erreur soit minimale, c'est-à-dire tel que $p(w|v)$ soit maximal. Reprenons ici la formule de Bayes :

$$p(w)\,p(v|w) = p(v)\,p(w|v)$$

De plus, on sait que les distributions sont constantes :

$$\begin{cases} p(w) = \frac{1}{M} \\ \frac{1}{p(v)} = constante \end{cases}$$

et dans l'expression

$$\frac{1}{M} p\left(v|w\right) \frac{1}{p\left(v\right)} = p\left(w|v\right)$$

on remarque que maximiser $p\left(w|v\right)$ revient à maximiser $p\left(v|w\right)$. Le décodeur idéal doit donc maximiser $p\left(v|w\right)$.

6.2.2 Minimum distance decoder

La distance de Hamming $d(v_1, v_2)$ est le nombre de bits qui diffèrent entre v_1 et v_2. Elle vérifie les propriétés d'une distance :

- $d\left(v_1, v_2\right) \geq 0$ et $d\left(v_1, v_2\right) = 0 \quad \Leftrightarrow \quad v_1 = v_2$
- $d\left(v_1, v_2\right) = d\left(v_2, v_1\right)$
- $d\left(v_1, v_2\right) + d\left(v_2, v_3\right) \geq d\left(v_1, v_3\right)$

Si on en revient aux messages envoyés dans un canal, on peut dire que $d(w, v)$ est le nombre de bits erronés dus au bruit dans le canal. Ainsi, la probabilité d'avoir v à la sortie alors qu'on a w à l'entrée est donnée par l'expression binomiale suivante :

$$p\left(v|w\right) = p^{d(v,w)} \left(1-p\right)^{n-d(v,w)} \tag{6.1}$$

où p est la probabilité d'erreur. Le premier terme représente la probabilité que d bits soient inversés, et le deuxième terme représente la probabilité que les n-d autres bits ne soient pas inversés.

Soit deux mots-codes w_1 et w_2. La probabilité d'attribuer v à w_1 ou w_2 doit être fonction de la distance de Hamming : on aura une plus grande probabilité d'attribuer v à w_1 si v est plus proche de w_1. Le "minimum distance decoder" est donc le décodeur qui choisit le w le plus proche de v.

Démonstration

En reprenant l'égalité (6.1) et en l'appliquant sur w_1 et w_2, on obtient :

$$\begin{cases} p\left(v|w_1\right) = p^{d(v,w_1)} \left(1-p\right)^{n-d(v,w_1)} \\ p\left(v|w_2\right) = p^{d(v,w_2)} \left(1-p\right)^{n-d(v,w_2)} \end{cases}$$

Si on divise une égalité par l'autre, on obtient :

$$\frac{p\left(v|w_1\right)}{p\left(v|w_2\right)} = \frac{p^{d(v,w_1)} \left(1-p\right)^{n-d(v,w_1)}}{p^{d(v,w_2)} \left(1-p\right)^{n-d(v,w_2)}}$$

Notons

$$\begin{cases} d\left(v, w_1\right) = d_1 \\ d\left(v, w_2\right) = d_2 \end{cases}$$

et rassemblons les termes en p au dénominateur et les termes en $(1-p)$ au numérateur :

$$\frac{p\left(v|w_1\right)}{p\left(v|w_2\right)} = \frac{p^{d_1} \left(1-p\right)^{n-d_1}}{p^{d_2} \left(1-p\right)^{n-d_2}} = \frac{\left(1-p\right)^{d_2-d_1}}{p^{d_2-d_1}} = \left(\frac{1-p}{p}\right)^{d_2-d_1}$$

Si on suppose que p est compris entre 0 (cas certain) et $\frac{1}{2}$ (cas totalement aléatoire), alors on peut dire que

$$\left(\frac{1-p}{p}\right) \geq 1$$

Et donc, $\frac{p(v|w_1)}{p(v|w_2)}$ est également supérieur à 1 dans le cas où $d_2 - d_1 > 0$
On a donc bien finalement :

$$p\,(v|w_1) \geq p\,(v|w_2) \quad \Leftrightarrow \quad d_2 > d_1$$

6.3 Distance minimale d'un CCE et performances

Il existe un lien entre la performance d'un CCE (Code Correcteur d'Erreur) et sa distance minimale. Cette distance minimale, notée d, est la plus petite distance de Hamming entre deux mots-code différents. Ainsi, si cette distance est maximale, le code est plus performant : il y a moins de chances de confondre les mots-code étant donné qu'ils sont plus "eloignés". Cependant, plus les mots-code sont éloignés, plus $m < n$, et plus le taux de transmission R est bas : on doit donc trouver un compromis. Pour trouver à quel point un code est performant, on tente de déterminer le nombre d'erreurs qu'il est capable de corriger :

- Si $d \geq 2e + 1$, alors le CCE corrige des erreurs jusqu'à l'ordre e (jusqu'à e bits erronés)
  Réciproquement, tout code corrigeant jusqu'à e erreurs doit satisfaire $d \geq 2e + 1$
- Si $d \geq 2e$, alors le CCE corrige des erreurs jusqu'à l'ordre $e-1$, mais détecte cependant e erreurs. Une de ces erreurs est impossible à corriger : on se situe en effet à égale distance entre deux mots-code et il y a une ambigüité non résolvable.

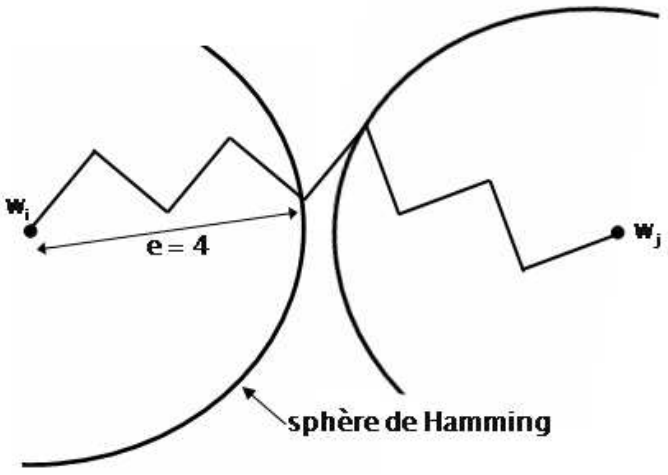

FIG. 6.2 – Sphères de Hamming avec d impair

Tout ceci est plus facile à représenter graphiquement (voir figure ci-dessus). Prenons deux mots-code w_i et w_j, éloignés d'une distance (la distance minimale) d. On représente une modification d'un bit par une arrête supplémentaire : chaque arrête correspond à une erreur. Si on pose qu'on désire corriger un nombre d'erreurs e = 4, on devra avoir $d \geq 2.4 + 1 = 9$. Les mots-code w_i et w_j doivent

donc être distants de 9 distances de Hamming, afin qu'il n'y ait aucune ambiguité : lorsque l'on fait 4 erreurs en partant de w_i, on reste toujours à une distance " plus près " de w_i que de w_j : on peut corriger l'erreur. Cette zone où le nombre d'erreurs ne dépasse pas 4 est une sphère de Hamming.

Est-ce réellement nécessaire que $d \geq 9$? Prenons un exemple où $d = 2.4 = 8$: il existe un mot-code v à égale distance de w_i et w_j, et donc : $d(w_i, v) + d(v, w_j) \geq d(w_i, w_j)$ avec $d(w_i, v) = e$ et $d(w_i, w_j) = 2e$. On peut donc avoir $d(v, w_j) = e$, mais il faut remarquer qu'à cette distance e, il y a une ambigüité : les sphères se touchent et, si on fait 4 erreurs, on ne sait plus si le mot-code émis est w_i ou w_j : impossible de corriger l'erreur (voir figure ci-dessous).

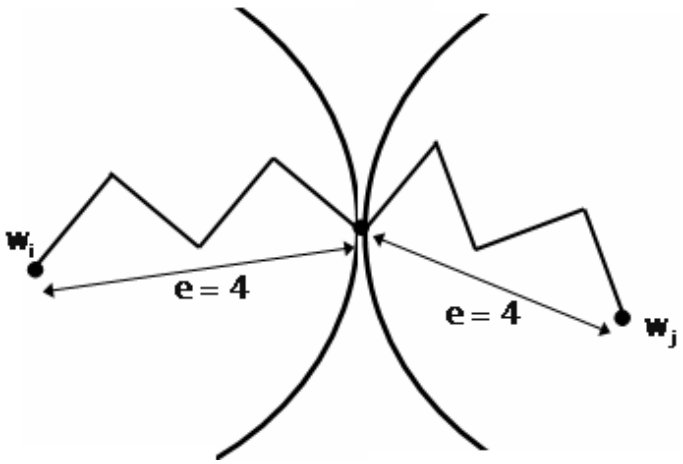

FIG. 6.3 – Sphères de Hamming avec d impair

Par contre, on détecte tout de même la présence de l'erreur.

6.4 Borne de Hamming

6.4.1 Théorème

La borne de Hamming donne le nombre de mots-code M maximal que l'on peut avoir dans un code qui corrige e erreurs, pour un n fixé. Elle est donnée par :

$$M \leq \frac{2^n}{\sum_{i=0}^{e} \binom{n}{i}}$$

et en général on prend le plancher de cette valeur :

$$\boxed{M \leq \left\lfloor \frac{2^n}{\sum_{i=0}^{e} \binom{n}{i}} \right\rfloor} \tag{6.2}$$

Démonstration

Pour que les sphères de Hamming ci-dessous soient toutes disjointes, une condition nécessaire (mais pas suffisante) est que le volume total des sphères ne dépasse pas le volume total autorisé, c'est-à-dire le nombre de combinaisons possibles de n bits.

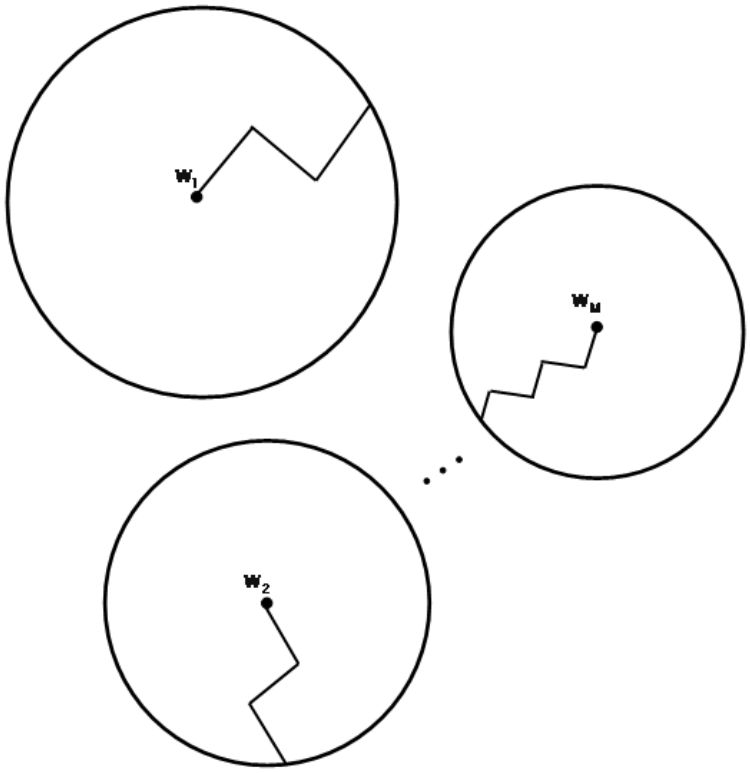

FIG. 6.4 – Sphères de Hamming disjointes

Ce nombre de combinaisons est donné par 2^n, et donc notre condition peut s'écrire :

$$\sum_{k=1}^{M} Vol_{\text{sphère k}} \leq 2^n$$

Le volume d'une sphère $Vol_{\text{sphère k}}$ est donné par le nombre de possibilités de s'écarter du point w_k tout en restant dans la sphère, c'est-à-dire la somme des éléments suivants :

- Le mot-code lui-même : 1
- Le nombre de possibilités de faire une erreur (on peut inverser un des n bits) : n
- Le nombre de possibilités de changer 2 bits parmi n bits : $\begin{pmatrix} n \\ 2 \end{pmatrix}$
- ...
- Le nombre de possibilités de changer e bits parmi n bits (on se retrouve sur le bord de la sphère) : $\begin{pmatrix} n \\ e \end{pmatrix}$

On a donc :

$$Vol_{\text{sphère k}} = 1 + n + \left(\begin{array}{c} n \\ 2 \end{array} \right) + ... + \left(\begin{array}{c} n \\ e \end{array} \right) = \sum_{i=0}^{e} \left(\begin{array}{c} n \\ i \end{array} \right)$$

La condition ci-dessus devient donc :

$$\sum_{k=1}^{M} Vol_{\text{sphère k}} = \sum_{k=1}^{M} \sum_{i=0}^{e} \left(\begin{array}{c} n \\ i \end{array} \right) \leq 2^n$$

Les sphères ayant toutes le même volume :

$$\sum_{k=1}^{M} Vol_{\text{sphère k}} = \sum_{k=1}^{M} \sum_{i=0}^{e} \left(\begin{array}{c} n \\ i \end{array} \right) = M \sum_{i=0}^{e} \left(\begin{array}{c} n \\ i \end{array} \right) \leq 2^n$$

On obtient donc finalement :

$$M \leq \frac{2^n}{\sum_{i=0}^{e} \left(\begin{array}{c} n \\ i \end{array} \right)}$$

6.4.2 Exemples

- Soit le code à répétition pour $n = 3$ répétitions : on représente donc 0 par "000" et 1 par "111", et on a $M = 2$ (on transmet 1 ou 0). La distance de Hamming minimale est $d = 3$. On en déduit par $d \geq 2e + 1$ que $e = 1$. On a donc, en utilisant la borne de Hamming (6.2) :

$$M = 2 \leq \left\lfloor \frac{2^n}{\sum_{i=0}^{e} \left(\begin{array}{c} n \\ i \end{array} \right)} \right\rfloor = \frac{2^3}{\left(\begin{array}{c} 3 \\ 0 \end{array} \right) + \left(\begin{array}{c} 3 \\ 1 \end{array} \right)} = \frac{2^3}{1 + \frac{3!}{(3-1)!\,1!}} = \frac{2^3}{1+3} = \frac{8}{4} = 2$$

  La borne de Hamming est donc saturée : l'espace des séquences est optimal.
- La borne de Hamming n'est pas une condition nécessaire pour ne pas avoir de recouvrement des sphères : il peut arriver qu'il n'existe pas de code qui sature la borne de Hamming. Par exemple, si on pose $n = 4$, $e = 1$, on sait que $d \geq 2e + 1 = 3$. On a donc, par la borne de Hamming :

$$M \leq \left\lfloor \frac{2^n}{\sum_{i=0}^{e} \left(\begin{array}{c} n \\ i \end{array} \right)} \right\rfloor = \left\lfloor \frac{2^4}{\left(\begin{array}{c} 4 \\ 0 \end{array} \right) + \left(\begin{array}{c} 4 \\ 1 \end{array} \right)} \right\rfloor = \left\lfloor \frac{16}{1 + \frac{4!}{(4-1)!\,1!}} \right\rfloor = \left\lfloor \frac{16}{5} \right\rfloor = \lfloor 3,2 \rfloor = 3$$

  Codons donc trois mots-code, pensant que ça va parfaitement marcher :

$$\begin{cases} w_1 = 0000 \\ w_2 = 0111 \\ w_3 = 1 \underbrace{...}_{x \text{ bits à } 1} \end{cases}$$

On a donc les distances de Hamming suivantes :

$$\begin{cases} d_{12} = 3 \\ d_{13} = 1 + x \\ d_{23} = 1 + (3 - x) \end{cases}$$

	$d_{1,3}$	$d_{2,3}$
x=0	1	4
x=1	2	3
x=2	3	2
x=3	4	1

FIG. 6.5 – Liste des possibilités de codage et distances de Hamming correspondantes (les cases en gris sont les distances non valides)

On a ici les 4 possibilités représentées dans le tableau ci-dessus, et on se rend compte qu'aucune de ces possibilités ne possède une distance minimale supérieure ou égale à 3. Ainsi, on ne peut coder ce w_3 d'aucune manière : M doit valoir 2, et la borne de Hamming n'est pas saturée.

6.5 Codes de Hamming

6.5.1 Introduction

Les codes de Hamming impliquent un décodage par multiplication par une matrice. On utilise ici un code correcteur d'erreur jouant sur la parité d'une chaine de n bits. On rajoute un bit en fin de chaine tel que le nombre de bits à 1 est pair. Par exemple : 010111 (les 5 premiers bits servant d'information, le dernier bit servant de bit de parité). Ainsi, on peut détecter (mais pas corriger) les erreurs en détectant une parité totale impaire. La condition pour ne pas avoir d'erreur est donc : $x_1 + x_2 + ... + x_n = 0$ (le "+" représentant le "ou exclusif").
Le principe des codes de Hamming est de généraliser tout ceci en testant, non plus la parité totale, mais la parité d'un sous ensemble de bits. On a alors un système d'équations linéaires :

$$\begin{cases} h_{11}x_1 + h_{12}x_2 + ... + h_{1n}x_n = 0 \\ ... \\ h_{m1}x_1 + h_{m2}x_2 + ... + h_{mn}x_n = 0 \end{cases}$$

En regroupant ce système sous forme de matrices, on obtient :

$$\begin{pmatrix} h_{11} & \cdots & h_{1n} \\ \vdots & \ddots & \vdots \\ h_{m1} & \cdots & h_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = 0$$

La matrice des coefficients h sera notée H (appelée matrice de Hamming ou matrice de parité) et la matrice colonne des x sera notée w (matrice "mot-code"). Ainsi, w est un mot-code si et seulement si

$$\boxed{Hw = 0} \tag{6.3}$$

On note t le rang de H, c'est-à-dire le plus souvent le nombre de lignes dans la matrice H. On note n le nombre de bits total, c'est-à-dire le nombre de colonnes de H. Ces colonnes se divisent en un

certain nombre de bits de parité (donné par t) et un certain nombre de bits d'information (donné par $k = n - t$). On note donc :

$$t = rang\,(H) \left\{ \underbrace{\begin{pmatrix} h_{11} & \cdots & h_{1n} \\ \vdots & \ddots & \vdots \\ h_{m1} & \cdots & h_{mn} \end{pmatrix}}_{n\ bits\ au\ total} \right.$$

$$\underbrace{}_{\substack{\text{t bits} \\ \text{de parité}}} \quad \underbrace{}_{\substack{k=(n-t)\ bits \\ d'information}}$$

Etant donné la présence de t bits de parité : $t = rang(H) \leq m < n$, on a un système avec plus d'inconnues (n) que d'équations (t ou m : en pratique ils sont les mêmes). Le système est donc indéterminé, ce qui indique qu'il existe plusieurs mots-code qui sont solution. Etant donné la présence de k bits d'information, on peut transmettre $M = 2^k$ mots-code : on a donc un taux de transmission de $R = \frac{\log M}{n} = \frac{k}{n}$

Exemple

Prenons un exemple où on a au total $n = 6$ bits, $m = t = 3$ bits de parité, et donc $k = n - m = 3$ bits d'information. Cela nous fait un nombre de mots-codes égal à $M = 2^k = 2^3 = 8$. La matrice H est donnée par :

$$H = \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{array} \right)$$

$$\underbrace{}_{m=3} \quad \underbrace{}_{k=3}$$

On va ici poser successivement des valeurs pour les bits d'information, et on résoudra ensuite le système de 3 équations à 3 inconnues pour trouver les bits de parité. Posons les valeurs :

$$\begin{cases} x_4 = 0 \\ x_5 = 1 \\ x_6 = 1 \end{cases}$$

Le système déquations donné par (6.3) se développe comme ceci :

$$\begin{cases} 1.x_1 + 0.x_2 + 0.x_3 + 0.0 + 1.1 + 0.1 = 0 \\ 0.x_1 + 1.x_2 + 1.x_3 + 1.0 + 0.1 + 1.1 = 0 \\ 1.x_1 + 0.x_2 + 1.x_3 + 1.0 + 0.1 + 1.1 = 0 \end{cases}$$

La première équation nous permet d'affirmer que $x_1 = 1$. La troisième indique que $x_1 + x_3 = 1$, c'est-à-dire que $x_3 = 0$. Enfin, la deuxième équation donne $x_2 + x_3 = 1$, c'est-à-dire $x_2 = 1$. On obtient donc le mot-code

$$w = 110011$$

En procédant de même pour d'autres valeurs de x_4, x_5, x_6, on trouve l'ensemble des mots-codes

ci-dessous. On trouve d=2 : le code ne corrige aucune erreur mais peut en détecter une.

$$\begin{cases} w_1 = 000000 \\ w_2 = 001001 \\ w_3 = 111010 \\ w_4 = 110011 \\ \vdots \\ w_6 = 000111 \end{cases}$$

6.5.2 Propriétés

La théorie qui tourne autour de la matrice H de Hamming possède certaines propriétés :

- Linéarité : un code est linéaire si la somme de 2 mots-code est un mot-code :

$$H(w_i + w_j) = Hw_i + Hw_j = 0$$

- Poids minimum d'un code : il s'agit du nombre w de 1 dans le mot-code où il y a le moins de 1 (sauf pour le mot-code 00...0 qui est toujours présent).
- Distance minimale d'un code : nombre minimum d de 1 par lequel diffère toute paire de mots-code. Etant donné l'existence systématique du mot-code 0, on peut dire que d est w pour les codes de Hamming :

$$\boxed{d = w} \tag{6.4}$$

6.5.3 Décodage du code de Hamming

On part de mots-code $w = x_1x_2...x_n$ et on a à la sortie $v = y_1y_2...y_n$. On pose le vecteur correcteur, ou vecteur syndrome : $C = Hv$. On sait donc quand il y a une erreur : c'est lorsque C n'est pas nul. On n'a pas d'erreur lorsque $C = Hv = Hw = 0$, mais l'inverse n'est pas vrai.
On pose $v = w + z$ où z est le vecteur d'erreur (error pattern vector), qui contient des 1 aux emplacements où les bits sont erronés, et 0 ailleurs. Si on remplace, on a donc :

$$C = Hv = H(w + z) = Hw + Hz$$

On sait que, selon léquation (6.3) des mots-codes, on a $Hw = 0$, et donc :

$$C = Hz = \sum_{j \in \{ j_1, \cdots, j_e \}} colonne_j\,(H)$$

Il peut cependant y avoir une ambigüité, comme dans l'exemple ci-dessous.

Exemple

Soit la matrice de Hamming

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

On a donc $n = 6$, $m = 3$ et $k = 3$. Une erreur simple en position 1 donnerait le syndrome

$$C = col_1(H) = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

alors qu'une erreur triple en 2, 3 et 5 donnerait

$$C = col_2(H) + col_3(H) + col_5(H) = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

ce qui correspond au même syndrome que pour l'erreur simple en 1. Le décodeur ne peut donc différentier les deux résultats que par la probabilité : une erreur simple étant la plus probable, il va traduire ce syndrome par une erreur simple.

6.5.4 Codes de Hamming "canoniques"

Voyons le codage et le décodage appliqué à un code de Hamming canonique (7,4) corrigeant des erreurs simples. On a la matrice de Hamming suivante :

$$H = \left(\underbrace{\begin{matrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{matrix}}_{\text{information}} \underbrace{\begin{matrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{matrix}}_{\text{parité}} \right)$$

car pour un code canonique, les différentes colonnes représentent les valeurs binaires du numéro de la colonne (colonne 1 : 001, colonne 2 : 010, colonne 3 : 011, ...). On trouve les mots-code suivants :

$$\begin{cases} w_0 = 0000 \;\; 000 \\ w_1 = 0001 \;\; 111 \\ w_2 = 0010 \;\; 110 \\ \vdots \\ w_{15=2^4-1} = 1111 \;\; 111 \end{cases}$$

avec $d = w = 3$ et donc $e = 1$.

Décrivons les étapes d'encodage te de décodage :

- Encodage : on a le système suivant :

$$\begin{cases} x_4 + x_5 + x_6 + x_7 = 0 \\ x_2 + x_3 + x_6 + x_7 = 0 \\ x_1 + x_3 + x_5 + x_7 = 0 \end{cases}$$

et on utilise le fait qu'en modulo 2, une somme ou une soustraction revient au même. On isole donc les bits de parité :

$$\begin{cases} x_4 = x_5 + x_6 + x_7 \\ x_6 + x_7 = x_2 + x_3 \\ x_5 + x_7 = x_1 + x_3 \end{cases}$$

En réinjectant $x_5 + x_7$ dans la première équation, on obtient :

$$\begin{cases} x_6 = x_1 + x_3 + x_4 \\ x_7 = x_2 + x_3 + x_6 = x_2 + x_3 + x_1 + x_3 + x_4 \\ x_5 = x_1 + x_3 + x_7 = x_1 + x_3 + x_2 + x_1 + x_4 \end{cases}$$

Dans la deuxième équation, les x_3 se simplifient et dans la troisième équation, les x_1 se simplifient. Le système devient donc :

$$\begin{cases} x_6 = x_1 + x_3 + x_4 \\ x_7 = x_1 + x_2 + x_4 \\ x_5 = x_2 + x_3 + x_4 \end{cases}$$

On est alors à même de réaliser différents circuits logiques avec des XOR (additions), comme ci-dessous pour x_5, ce qui représentera l'encodeur.

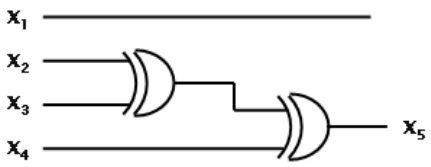

FIG. 6.6 – Circuit logique représentant une partie du système déquations

- Décodage : supposons que le j^{ème} bit soit erroné, on a la matrice z suivante :

$$z = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \leftarrow \text{ position j}$$

On aura donc comme syndrome :

$$C = Hz = col_j\,(H)$$

c'est-à-dire la position de l'erreur (j) exprimée en binaire. Le décodage est donc très facile : si

$$C = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

alors on n'a pas d'erreur simple, et si par exemple

$$C = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

alors le troisième bit est erroné. Par contre, ces codes canoniques ne fonctionnent que pour $e = 1$. Supposons $e = 2$ et qu'on a une erreur en 1 et en 2. Le syndrome devient :

$$C = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

et il est impossible de discerner cette erreur d'une erreur simple sur le troisième bit. Remarquons que ce code canonique particulier sature la borne de Hamming :

$$2^4 \leq \frac{2^7}{1+7} = 2^4$$

En réalité, ceci se généralise à tout code de Hamming canonique. Puisque $k = n - m$:

$$2^k = 2^{n-m} = \frac{2^n}{\sum \begin{pmatrix} n \\ i \end{pmatrix}} = \frac{2^n}{1+n}$$

En simplifiant les 2^n, on a :

$$2^{-m} = \frac{1}{1+n}$$

c'est-à-dire :

$$\boxed{2^m = 1 + n} \tag{6.5}$$

On trouve donc une condition pour que le code de Hamming canonique puisse toujours saturer la borne de Hamming. Le m joue alors le rôle de paramètre, puisqu'une fois fixé, on trouve n et k.
Cependant, il n'est pas forcément très pratique de ne corriger qu'une seule erreur.

6.5.5 Lien entre e et H

Le code de Hamming défini par H corrige jusqu'à e erreurs si et seulement si tous les ensembles de 2e colonnes de H sont constitués de colonnes linéairement indépendantes.

Démonstration

Le code corrige e erreurs si et seulement si tous les patterns (différentes matrices z possibles) de e erreurs correspondent à des vecteurs syndrome C différents.
Le vecteur C doit ainsi être différent pour tous les z possibles, donc une combili de e colonnes doit être différente d'une combili de e autres colonnes : tout ensemble de 2e colonnes doit être constitué de colonnes linéairement indépendantes.

Un bon moyen pour avoir des colonnes linéairement indépendantes est d'insérer une matrice identité dans H, comme pour le H ci-dessous.

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Si on y prend un groupe de 4 colonnes, on est sûr qu'elles sont linéairement indépendantes : le code que représente ce H corrige 2 erreurs. On peut le vérifier en jetant un coup d'œil à la borne de Hamming. Pour $e = 2$, on a

$$2^3 \leq \frac{2^{10}}{1 + 10 + \begin{pmatrix} 10 \\ 2 \end{pmatrix}} = 18,3$$

alors que pour $e = 3$, on a

$$2^3 \leq \frac{2^{10}}{1 + 10 + \begin{pmatrix} 10 \\ 2 \end{pmatrix} + \begin{pmatrix} 10 \\ 3 \end{pmatrix}} = 5,8$$

inégalité qui n'est pas respectée : ce code est incapable de corriger 3 erreurs, ce qui confirme ce que nous avons trouvé précédemment.

6.5.6 Bornes sur le nombre de bits de parité m

- Borne inférieure de Hamming
  Si n et e sont fixés, alors on a

$$M \leq \frac{2^n}{\sum \begin{pmatrix} n \\ i \end{pmatrix}}$$

  Comme $M = 2^k = 2^{n-m}$, on a :

$$2^{n-m} \leq \frac{2^n}{\sum \begin{pmatrix} n \\ i \end{pmatrix}}$$

  et donc :

$$2^{-m} \leq \frac{1}{\sum \begin{pmatrix} n \\ i \end{pmatrix}}$$

  c'est-à-dire :

$$\boxed{\sum_{i=0}^{e} \begin{pmatrix} n \\ i \end{pmatrix} \leq 2^m} \tag{6.6}$$

  Ceci définit une borne inférieure pour la valeur de m :

$$\log_2 \left[\sum \begin{pmatrix} n \\ i \end{pmatrix} \right] \leq m$$

Par exemple, pour l'exemple de $n = 10$ et $e = 2$, on a :

$$2^m \geq 1 + 10 + \begin{pmatrix} 10 \\ 2 \end{pmatrix} = 56$$

et donc

$$m \geq \left\lceil \underbrace{\log_2 56}_{5,8} \right\rceil = 6$$

- Borne supérieure de Hamming : si n et e sont fixés, alors on a

$$\boxed{\sum_{i=0}^{2e-1} \begin{pmatrix} n-1 \\ i \end{pmatrix} \leq 2^m} \tag{6.7}$$

ce qui constitue une condition suffisante pour construire un code de Hamming qui corrige e erreurs. Pour notre exemple ci-dessus, on aura donc :

$$2^m \geq \sum_{i=0}^{3} \begin{pmatrix} 9 \\ i \end{pmatrix} = 1 + 9 + \begin{pmatrix} 9 \\ 2 \end{pmatrix} + \begin{pmatrix} 9 \\ 3 \end{pmatrix} = 130$$

et donc :

$$m \geq \left\lceil \underbrace{\log 130}_{7,02} \right\rceil = 8$$

En conclusion, le code fonctionnera certainement si $m \geq 8$ (mais ne sera peut-être pas optimal), et ne fonctionnera certainement pas si $m < 6$. L'expérience nous apprend que le point optimal se situe à mi-chemin : en $m = 7$, ce que représente le schéma ci-dessous.

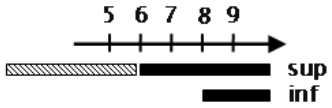

FIG. 6.7 – Représentation graphique des bornes de Hamming

6.6 Autres codes correcteurs d'erreur

6.6.1 Classification des CCE

Les codes correcteurs d'erreur se classent en deux catégories principales :

- Codes en blocs (ou codes algébriques) : on a une séquence à coder qui est coupée en blocs de longueur k, envoyées dans un canal puis décodés à la sortie
- Codes en treillis (ou tree-codes) : non vus dans ce cours

Ces codes peuvent chacun être linéaires, ou non linéaire. Les codes en treillis linéaires sont nommés codes convolutifs. Les codes linéaires en bloc se divisent encore en deux catégories : les codes cycliques et non cycliques. Parmi les codes cycliques, quelques-uns sont fréquemment utilisés :

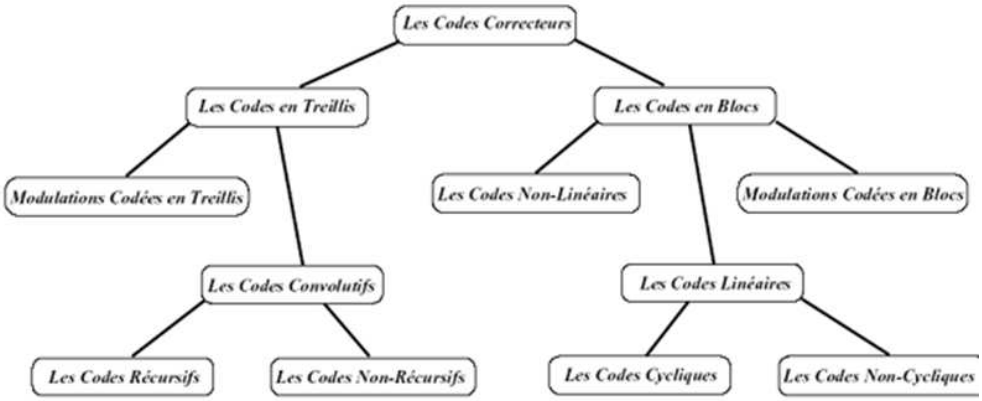

FIG. 6.8 – Ensemble des codes correcteurs d'erreur

- Codes BCH (Bose / Chaudhuri / Hocquenghen)
- Codes de Reed-Solomon : on les utilise par exemple dans les lecteurs CD. Ils sont "maximum distance separable", c'est-à-dire que pour n, k fixés, ils corrigent le plus d'erreurs possibles : ils saturent la borne de Singleton

6.6.2 Borne de Singleton

Pour n et k fixés, la borne de Singleton est définie par :

$$\boxed{d \leq n - k + 1} \tag{6.8}$$

ce qui fournit également le nombre d'erreurs maximum corrigeables. De plus, si on a un code qui corrige e erreurs, il corrige également $e' = 2e$ effacements.
Imaginons deux mots-code ci-dessous

$$\begin{cases} x_1\ x_2\ x_3\ x_4\ x_5 \\ x_1\ x_2\ x_3\ x_4\ x_5 \end{cases}$$

pour lesquels les deux premiers bits (x_1 et x_2) ont été effacés : il faut nécessairement qu'un des 3 bits restant soit différent d'un mot-code à l'autre, afin de toujours pouvoir distinguer les mots-code. On a donc la condition $d = e' + 1$. On sait aussi que $d = 2e + 1$, et on a donc bien $e' = 2e$.
Prenons l'exemple

$$\begin{cases} 011001 \\ 101000 \end{cases}$$

Si les deux premiers bits sont effacés, le dernier bit distingue toujours les deux mots-code.

Démonstration

L'inégalité ci-dessus se montre par le fait qu'il doit rester, après effacement, au moins tous les bits d'information :

$$n - e' \geq k$$

En isolant le e', on obtient :

$$e' \leq n - k$$

Comme on sait que $d = e' + 1$, on peut noter :

$$d - 1 \leq n - k$$

en isolant d, on a finalement :

$$d \leq n - k + 1$$

6.6.3 Codes cycliques

L'avantage de ce genre de codes, c'est qu'il existe une technique pour construire la matrice H de façon systématique, en se basant sur la notion de registre à décalage.

- Principe : on a n registres, et à chaque top d'horloge, on décale les bits vers la gauche, et la cellule x_{n-1} reçoit une combili des autres cellules.

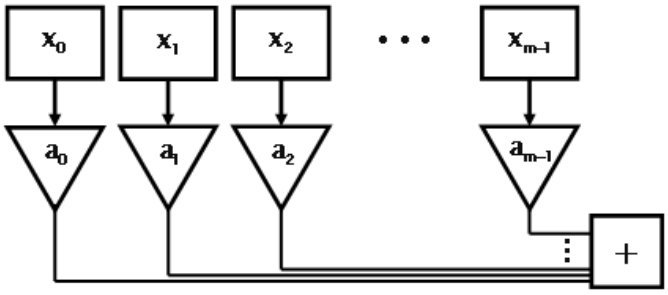

FIG. 6.9 – Représentation du registre à décalage

On a donc les liens suivants entre deux top d'horloge :

$$\begin{cases} x'_0 = x_1 \\ x'_1 = x_2 \\ \vdots \\ x'_{m-2} = x_{m-1} \\ x'_{m-1} = a_0x_0 + a_1x_1 + ... + a_{m-1}x_{m-1} \end{cases}$$

qui s'écrit, sous forme matricielle, $X' = TX$, avec

$$T = \begin{pmatrix} 0 & \left[\begin{matrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{matrix}\right] \\ a_0 & a_1 \quad \cdots \quad a_{m-1} \end{pmatrix}$$

où on a, entre crochets, une matrice identité, impliquant que $detT = a_0$, qui doit être différent de 0 si on veut inverser T pour avoir X à partir de X' : $X' = T^{-1}X'$. On doit donc poser

$$a_0 = 1$$

A chaque coup d'horloge, on tombe donc sur un nouvel état, et on obtient des séquences $X_0, X_1 = TX_0, X_2 = T^2X_0$, ce qui constitue au final un cycle de longueur n (à condition que X_0 ne soit

pas nul car on aurait alors un cycle de longueur 1, ce qui n'est pas très intéressant) puisque l'on revient sur un état "déjà visité". On a donc :

$$\boxed{n \leq 2^m - 1} \tag{6.9}$$

où 2^m représente tous les états possibles, et "1" représente l'état $X_0 = 0$ qui n'est pas intéressant.

Exemple

Pour $m = 4$, on pose les coefficients suivants :

$$\begin{cases} a_0 = 1 \quad \text{(imposé)} \\ a_1 = 0 \\ a_2 = 1 \\ a_3 = 0 \end{cases}$$

et l'état initial suivant :

$$\begin{cases} x_0 = 0 \\ x_1 = 0 \\ x_2 = 0 \\ x_3 = 1 \end{cases}$$

Typiquement, on pose comme état initial un ensemble de 0 sauf la dernière cellule qui est à 1. On va donc suivre la suite d'états ci-dessous, pour finalement retomber sur cet état 0001.

$$\begin{cases} 0001 = 1 \\ 0010 = 2 \\ 0101 = 5 \\ 1010 = 10 \\ 0100 = 4 \\ 1000 = 8 \\ 0001 = 1 \end{cases}$$

On a alors un cycle de longueur 6, représenté par le schéma ci-dessous. Il existe d'autres cycles si on part de l'état initial 3, 6 ou 0.

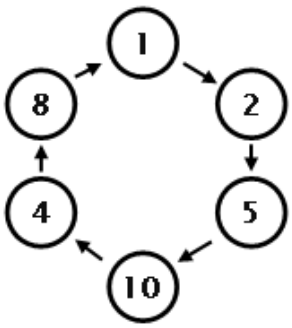

FIG. 6.10 – Cycle de longueur 6

- Définition : un code est cyclique s'il est tel que, si une séquence $w = (x_0\ x_1\ \ ...\ \ x_{m-1})$ est un mot-code, alors $w' = (x_{m-1}\ x_0\ \ ...\ \ x_{m-2})$ est aussi un mot-code.

- Théorème : si on construit un code de Hamming avec

$$H = \begin{pmatrix} X & TX & T^2X & \cdots & T^{n-2}X & T^{n-1}X \end{pmatrix}$$

alors ce code est cyclique.

Démonstration

Si $w = (x_0 ... x_{n-1})^T$ est un mot-code, alors, selon la condition (6.3), on a :

$$Hw = \begin{pmatrix} X & TX & T^2X & \cdots & T^{n-2}X & T^{n-1}X \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ x_{n-1} \end{pmatrix} = 0$$

Développons cette équation :

$$x_0X + x_1TX + x_2T^2X + ... + x_{n-1}T^{n-1}X = 0$$

En multipliant les deux membres par T :

$$x_0TX + x_1T^2X + x_2T^3X + ... + x_{n-2}T^{n-1}X + x_{n-1}T^nX = 0$$

et nous savons que T^n est égal à la matrice identité, puisque le cycle est de longueur n. On a donc, en réorganisant :

$$x_{n-1}X + x_0TX + x_1T^2X + x_2T^3X + ... + x_{n-2}T^{n-1}X = 0$$

c'est-à-dire, sous forme matricielle :

$$\begin{pmatrix} X & TX & T^2X & \cdots & T^{n-2}X & T^{n-1}X \end{pmatrix} \begin{pmatrix} x_{n-1} \\ x_0 \\ \vdots \\ x_{n-2} \end{pmatrix} = 0$$

Ce qui veut dire que

$$w' = \begin{pmatrix} x_{n-1} \\ x_0 \\ \vdots \\ x_{n-2} \end{pmatrix}$$

est un mot-code également.

Exemple

Reprenons le cycle obtenu pour l'exemple précédent, et notons à chaque fois le nombre binaire que chaque colonne représente :

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 5 & 10 & 4 & 8 \end{pmatrix}$$

avec donc $n = 6$, $m = 4$ et $k = 2$, ce qui donne $M = 2^2 = 4$. On trouve les mots-code ci-dessous

$$\begin{cases} w_0 = 000000 \\ w_1 = 101010 \\ w_2 = 010101 \\ w_3 = 111111 \end{cases}$$

On vérifie bien le théorème étant donné qu'un décalage cyclique amène à un autre mot-code : w_0 revient à w_0, w_1 devient w_2 et inversement, w_3 revient à w_3.

6.6.4 Codes BCH

Ce sont des codes cycliques très performants, avec n dépassant 1000. La matrice de transition T a une taille $q \times q$ et le code a donc une période maximale $2^q - 1$.
La matrice de parité H se construit en prenant des exposants impaires de T, en allant jusqu'à $2e - 1$:

$$H = \begin{pmatrix} X & TX & T^2X & T^3X & \cdots & T^{n-1}X \\ X & T^3X & T^6X & T^9X & \cdots & T^{3(n-1)}X \\ X & T^5X & T^{10}X & T^{15}X & \cdots & T^{5(n-1)}X \\ \vdots & \vdots & \vdots & \cdots & \ddots & \vdots \\ X & T^{2e-1}X & T^{2(2e-1)}X & T^{3(2e-1)}X & \cdots & T^{(n-1)(2e-1)}X \end{pmatrix}$$

On utilise alors la propriété que si tout bloc de 2e colonnes est constitué de colonnes linéairement indépendantes, alors le code corrige e erreurs.