

Cours 1
Rappels et Compléments de codage source

—
ESIGETEL

Abdellatif ZAIDI

Septembre 2003

Plan du cours

- Codage source des sources discrètes
 - Introduction
 - Codes préfixes
 - Codes arithmétiques
 - Régulation du débit
 - Codes universels
- Codage source des sources continues
 - Courbes débit-distorsion
 - Quantification scalaire
 - Quantification vectorielle

1 Codage source des sources discrètes

On considère des sources *discrètes*
(symboles provenant d'un alphabet fini)

Sources présentent en général de la redondance



Codage source : codage éliminant cette redondance

1.1 Introduction

Deux types de méthodes :

1. Celles nécessitant un modèle de la source
codage de Huffman, codage arithmétique, ...
2. Celles ne nécessitant pas d'information *a priori*
⇒ estimation des statistiques de la source
codage universel, codage de Liv-Zempel, ...

Meilleur sera le modèle \implies meilleure sera le taux de compression

Une source discrète délivre une suite de variables aléatoires

$$\dots, X_{-2}, X_{-1}, X_0, X_1, X_2, \dots$$

Suite infinie de variables aléatoires \implies modèle :

- source discrète sans mémoire
- source discrète de Markov
- source discrète de Markov d'ordre m

1.1.1 Source discrète sans mémoire

Les symboles appartiennent à un alphabet à J symboles

$$\{a_0, \dots, a_{J-1}\}$$

A chaque coup d'horloge, un symbole X est produit

↪ probabilité $p(X = a_j) = p_j$

↪ *indépendance* par rapport aux symboles déjà émis

↪ *stationnarité* (sous-entendue) : p_j ne dépend pas du temps.

Ainsi, on a

$$\Pr(X_1 = a_{j_1}, \dots, X_n = a_{j_n}) = \prod_{\ell=1}^n p(X_\ell = a_{j_\ell}).$$

L'information associée à chaque symbole d'une source discrète sans mémoire de distribution de probabilité

$$\mathbf{p} = (p_0, \dots, p_{J-1})^T$$

est

$$I(p_j) = \log_2 \left(\frac{1}{p_j} \right) = -\log_2(p_j) \text{ (en bit/symbole).}$$

L'*entropie* d'une source est la quantité moyenne d'information apportée par les symboles pouvant être émis.

$$H(X) = -\sum_{j=0}^{J-1} p_j \log_2(p_j) \text{ (en bit/symbole).}$$

Pour un alphabet à J symboles,

$$H(X) \leq \log_2 J.$$

L'égalité est atteinte pour des symboles équiprobables.

L'entropie d'une source émettant des blocs de N symboles sera N fois l'entropie d'une source émettant un symbole après l'autre.

Exemples :

1. Une source binaire émet des symboles X appartenant à l'alphabet $\{0, 1\}$ avec une probabilité $\mathbf{p} = (0.1, 0.9)$:

$$H(X) = -0.1 \log 0.1 - 0.9 \log 0.9 = 0.469 \text{ bit/symbole}$$

2. Une nouvelle source émet des groupes de 2 symboles $\mathbf{X} = (X^1, X^2)$ appartenant à l'alphabet $\{(0, 0), (0, 1), (1, 0), (1, 1)\}$ avec une probabilité $\mathbf{p} = (0.01, 0.09, 0.09, 0.81)$:

$$\begin{aligned} H(\mathbf{X}) &= -0.01 \log 0.01 - 0.09 \log 0.09 \\ &\quad - 0.09 \log 0.09 - 0.81 \log 0.81 \\ &= 0.838 \text{ bit}/2 \text{ symboles} \\ &= 0.469 \text{ bit/symbole} \end{aligned}$$

3. Une source émet des symboles X parmi $\{-3, -1, 1, 3\}$ avec une probabilité $\mathbf{p} = (0.1, 0.4, 0.4, 0.1)$:

$$\begin{aligned} H(X) &= -0.1 \log 0.1 - 0.4 \log 0.4 - 0.4 \log 0.4 - 0.1 \log 0.1 \\ &= 1.722 \text{ bit/symbole} \end{aligned}$$

1.1.2 Source discrète de Markov

Source dont la probabilité d'émettre le symbole a_j dépend seulement du symbole précédemment émis.

$$\Pr (X_0 = a_j | X_{-1}, X_{-2}, \dots) = \Pr (X_0 = a_j | X_{-1}).$$

Comportement décrit de manière compacte par une matrice de transition \mathbf{P} dont les éléments sont définis par

$$P_{j|i} = \Pr (X_0 = a_j | X_{-1} = a_i).$$

Dans le cas général de sources de Markov d'ordre m , la probabilité d'émettre le symbole a_j dépend des m derniers symboles émis

$$\Pr (X_0 = a_j | X_{-1}, X_{-2}, \dots) = \Pr (X_0 = a_j | X_{-1}, X_{-2}, \dots, X_{-m}).$$

Exemple :

On considère une source binaire discrète de Markov d'ordre 1 et de matrice de transition

$$\mathbf{P} = \begin{pmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{pmatrix} = \begin{pmatrix} p(0|0) & p(1|0) \\ p(0|1) & p(1|1) \end{pmatrix}.$$

1. Si $X_{-1} = 0$,

$$p(X_0 = 0|0) = 0.8 \text{ et } p(X_0 = 1|0) = 0.2$$

2. Si $X_{-1} = 1$,

$$p(X_0 = 0|1) = 0.3 \text{ et } p(X_0 = 1|1) = 0.7$$

Shannon : il existe un code source permettant d'atteindre
en moyenne $H(\mathbf{p})$ bits/symbole (borne inférieure)

1.2 Codes préfixes

On considère une source discrète

- émettant des symboles X appartenant à un alphabet de J symboles
- distribution de probabilité \mathbf{p}
- entropie $H(\mathbf{p})$ bits/symbole

Construction des codes de Huffman

∈

codes longueur fixe \longrightarrow longueur variable

codes préfixes

Exemple :

Une source discrete sans mémoire peut émettre huit type de symboles

$$p_0 = p_1 = p_2 = p_3 = \frac{1}{32} \quad p_6 = \frac{1}{4}$$
$$p_4 = p_5 = \frac{1}{16} \quad p_7 = \frac{1}{2}$$

L'entropie est $H(X) = 2.125$ bits/symbole.

Un code à longueur variable permettant d'atteindre l'entropie est

Symbole	Probabilité	Mot de code	Longueur du mot de code
0	2^{-5}	00000	5
1	2^{-5}	00001	5
2	2^{-5}	00010	5
3	2^{-5}	00011	5
4	2^{-4}	0010	4
5	2^{-4}	0011	4
6	2^{-2}	01	2
7	2^{-1}	1	1

Longueur moyenne d'un mot de code

$$\bar{\ell} = \sum_j p_j \ell_j = 2.125.$$

\implies l'entropie $H(X)$ est atteinte.

Propriétés :

- Le débit moyen nécessaire sera de 2.125 bits/symbole après codage.
- Une séquence codée peut être décodée de manière unique

000100100101 ... \rightarrow 2647 ...

- Pas besoin de séparer les mots de code (ponctuation inutile)
 \hookrightarrow propriété principale des codes *préfixes*
- Dans le cas général, l'entropie n'est pas atteinte

Définition 1 *Un code préfixe associé à un alphabet $\{a_0, a_1, \dots, a_{J-1}\}$ est une collection de mots de longueur variable associés à chaque symbole de l'alphabet de sorte qu'aucun mot ne soit égal au début (au préfixe) d'un autre mot.*

$$\begin{array}{cc}
 \mathcal{C}_0 = \left\{ \begin{array}{l} 00 \\ 01 \\ 10 \\ 110 \\ 1110 \\ 1111 \end{array} \right. & \mathcal{C}_1 = \left\{ \begin{array}{l} 00 \\ 01 \\ 10 \\ 010 \\ 1110 \\ 1111 \end{array} \right. \\
 \text{code préfixe} & \text{code non préfixe}
 \end{array}$$

Si la chaîne

011101110110...

a été générée avec \mathcal{C}_0 , on peut reconstituer les mots de code

01, 110, 1110, 110, ...

Définition 2 *Un code préfixe est complet si toute chaîne semi-infinie composée de symboles appartenant l'alphabet de mots de code peut être décomposée en mots de codes (ponctuée) de manière unique.*

\mathcal{C}_0 est un code préfixe complet car, par exemple,

011011000100000...

peut être ponctuée

01, 10, 110, 00, 10, 00, 00, ...

Théorème 1 (inegalite de Kraft) *Pour un alphabet de taille K , il existe un code préfixe*

- à M mots de code*
 - chacun ayant une longueur ℓ_m pour $m = 0, \dots, M - 1$*
- si et seulement si*

$$\sum_{m=0}^{M-1} K^{-\ell_m} \leq 1.$$

Ce code préfixe est complet lorsque l'égalité est atteinte.

Théorème 2 (Shannon) *Pour toute source discrète de distribution de probabilité \mathbf{p} , il existe un code binaire préfixe tel que la longueur moyenne $\bar{\ell}$ des mots de code satisfait*

$$H(\mathbf{p}) \leq \bar{\ell} \leq H(\mathbf{p}) + 1$$

De plus, pour tout $\varepsilon > 0$, en réorganisant les symboles de la source, il existe un code binaire préfixe tel que

$$H(\mathbf{p}) \leq \bar{\ell} \leq H(\mathbf{p}) + \varepsilon$$

\implies Bornes sur les performances atteignables.

Théorème 3 *Un code binaire préfixe dont la longueur moyenne des mots de codes est minimale pour une distribution de probabilité \mathbf{p} a les propriétés suivantes*

1. *Si $p_j \geq p_{j'}$ alors $l_j \leq l_{j'}$*
2. *Les mots de code associés aux deux symboles les moins probables de la source ont la même longueur*
3. *Si au moins deux mots de code ont la même longueur, alors deux de ces mots de code sont identiques partout sauf pour le bit de poids le plus faible*

Construction d'un code préfixe optimal :

On considère une source à J symboles et on veut construire le code \mathcal{C} minimal pour cette source

1. Prendre les deux symboles a_j et a_k les moins probables et les combiner en un symbole fictif a_{jk} de probabilité $p_j + p_k$
 \implies nouvelle source de longueur $J - 1$
2. Trouver le code \mathcal{C}_1 minimal pour cette nouvelle source
3. Le mot de code associé à a_j et à a_k est obtenu en ajoutant 0 et 1 au mot de code associé au symbole fictif a_{jk}
 $\implies \mathcal{C}$ est alors déduit simplement de \mathcal{C}_1 .

Théorème 4 *Si \mathcal{C}_1 est minimal pour la source à $J - 1$ symboles alors \mathcal{C} est minimal pour la source à J symboles.*

Exemple 1 *On a une source émettant des symboles appartenant à*

$$\{A, B, C, D, E, F, G\}$$

et de distribution de probabilité

$$\mathbf{p} = \left(\frac{3}{8}, \frac{3}{16}, \frac{3}{16}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{32} \right)^T$$

On obtient un code pour lequel

$$\begin{aligned} l &> H(\mathbf{p}) \\ 2.44 &> 2.37 \end{aligned}$$

Amélioration des performance \implies coder des groupes de symboles.

Symbole	Probabilité	Mot de code	Longueur du mot de code
<i>A</i>	$3/8$	1	1
<i>B</i>	$3/16$	011	3
<i>C</i>	$3/16$	010	3
<i>D</i>	$1/8$	001	3
<i>E</i>	$1/16$	0001	4
<i>F</i>	$1/32$	00001	5
<i>G</i>	$1/32$	00000	5



Exemple 2 *On a une source emettant des symboles appartenant à*

$$\{A, B, C\}$$

et de distribution de probabilité

$$\mathbf{p} = \left(\frac{3}{4}, \frac{3}{16}, \frac{1}{16} \right)^T$$

On obtient un code pour lequel

$$\begin{aligned} \ell &> H(X) \\ 1.25 &> 1.012 \end{aligned}$$

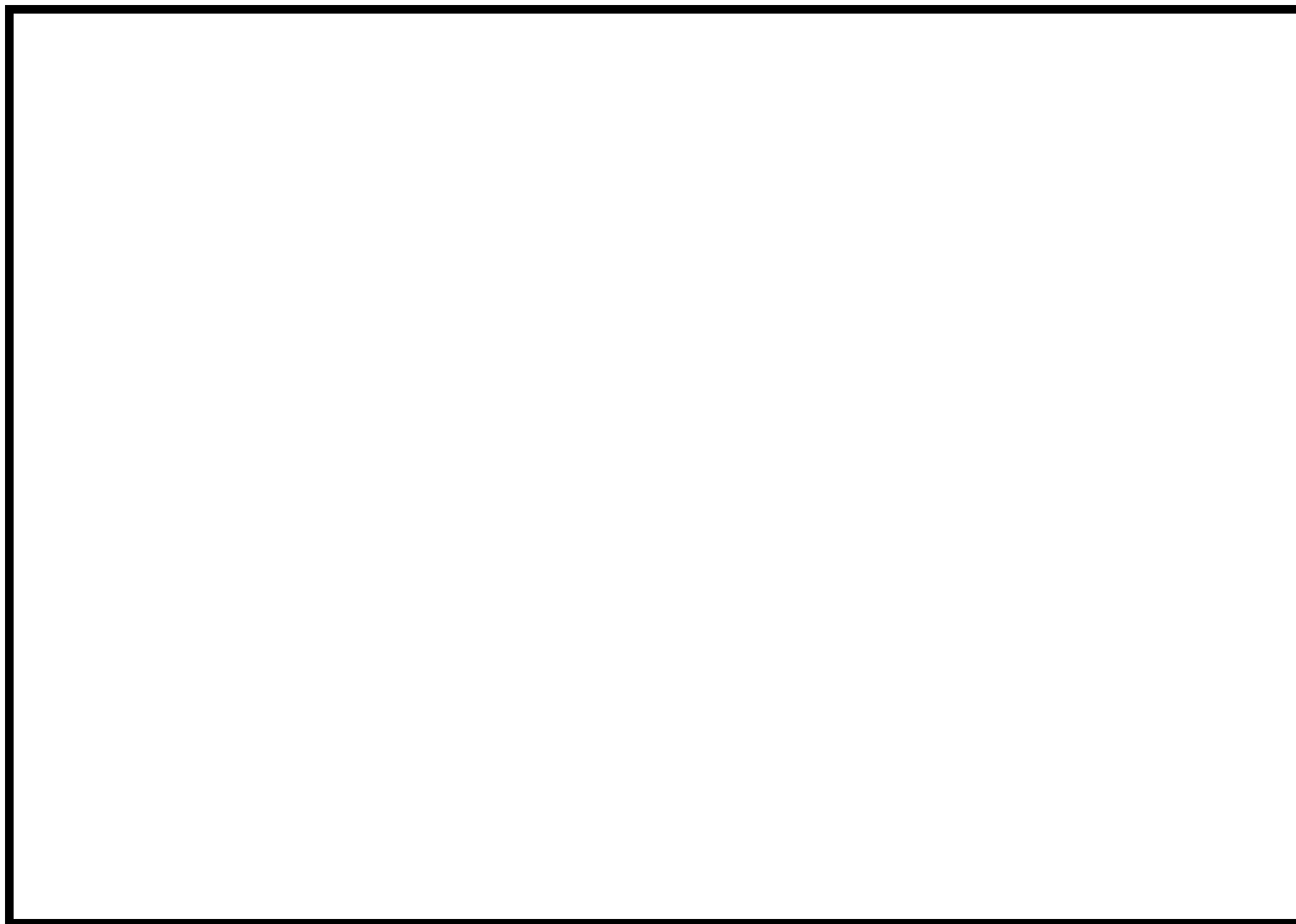


Exemple 3 (suite) *En regroupant les symboles par groupes de deux, on peut obtenir un meilleur code*

<i>Symbole</i>	<i>Probabilité</i>	<i>Mot de code</i>	<i>Longueur du mot de code</i>
<i>AA</i>	0.5625	1	1
<i>AB</i>	0.1406	011	3
<i>AC</i>	0.0469	0011	4
<i>BA</i>	0.1406	010	3
<i>BB</i>	0.0352	0001	4
<i>BC</i>	0.0117	00001	5
<i>CA</i>	0.0469	0010	4
<i>CB</i>	0.0117	000001	6
<i>CC</i>	0.0039	000000	6

\Rightarrow *performances obtenues*

$$1.035 \text{ bits/symbole} > H(X) = 1.012 \text{ bits/symbole}$$



Remarque :

Entropie d'une source discrète :

$$H(\mathbf{p}) = - \sum_j p_j \log_2 p_j.$$

Longueur moyenne d'un mot de code préfixe :

$$\bar{\ell} = \sum_j p_j \ell_j.$$

\implies Codage optimal réalisé lorsque

$$\ell_j = -\log_2 p_j.$$

Solutions :

1. Coder des blocs, de manière à réduire les p_j
2. Introduire des symboles fictifs :
 A avec $p_A = 3/4 \longrightarrow \{A_1, A_2, A_3\}$ avec $\mathbf{p} = (1/2, 1/8, 1/16)$
3. Utiliser d'autres types de codes...

1.3 Codes arithmétiques

Limitation du codage de Huffman :

Sources d'entropie $H(X) \ll 1$
 $\implies \bar{\ell}$ sera limité à 1 sur l'alphabet original
(exemple : fax, images NB, ...)

Codage en arbre : codage de séquence de longueur infinie

\implies réponse à cette limitation

1.3.1 Code d'Elias

On considère une source binaire sans mémoire
d'alphabet $\{a_0, a_1\}$ avec $\mathbf{p} = (p_0, 1 - p_0) = (0.7, 0.3)^T$.

↔ Entropie $H(X) = 0.88$ bits/symboles.

La source génère une suite infinie de symboles, par exemple

.001101101001011...

⇒ nombre décimal codé binaire ρ de longueur infinie.

(rappel : $0.5_{10} \longrightarrow 0.1_2 = 1 * 2^{-1}$)

Codage d'Elias :

codeur : $\rho \longrightarrow \rho_c$

décodeur : $\rho_c \longrightarrow \rho$

Rappel sur le codage d'un nombre réel :

$$0.7_{10} = 0.101\dots_2 = 1 * 2^{-1} + 0 * 2^{-2} + 1 * 2^{-3} + \dots$$

Principe :

Le codage d'Elias partitionne l'intervalle $[0, 1)$ en sous-intervalles de taille p_0 et $1 - p_0$.

Pour l'exemple considéré, $\mathbf{p} = (0.7, 0.3)^T$, l'intervalle obtenu sera
 $[0, 0.7)$ et $[0.7, 1)$.

Ensuite,

- si premier symbole émis est 0, $\rho_c \leftrightarrow [0, 0.7)$,
- si premier symbole émis est 1, $\rho_c \leftrightarrow [0.7, 1)$.

La procédure est itérée,

- soit sur $[0, 0.7)$, scindé en $[0, 0.49)$ et $[0.49, 0.7)$
- soit sur $[0.7, 1)$, scindé en $[0.7, 0.91)$ et $[0.91, 1)$

\implies encadrement associé à ρ_c de plus en plus précis.



A l'instant $n - 1$,

$$\rho_c \leftrightarrow [A_{n-1}, B_{n-1}].$$

A l'instant n , en remplaçant 0.7 par p_0 , on obtient l'encadrement suivant

– si le $n^{\text{ème}}$ symbole vaut 0

$$A_n = A_{n-1}$$

$$B_n = A_{n-1} + p_0 (B_{n-1} - A_{n-1}),$$

– si le $n^{\text{ème}}$ symbole vaut 1

$$A_n = A_{n-1} + p_0 (B_{n-1} - A_{n-1})$$

$$B_n = B_{n-1}.$$

L'intervalle construit va se resserrer autour de ρ_c .

Lorsque L_n bits de poids fort seront communs à A_n et B_n , ces bits seront également communs à ρ_c

\implies ils pourront être envoyés vers le canal.

Exemple 4 Si $A_n = 0.\underline{01000}110\dots$ et $B_n = 0.\underline{0100}1100\dots$ alors

$$\rho_c = .0100\dots$$

Remarques :

- Le calcul de L_n se fait simplement en comptant le nombre de zéros de poids fort dans de $B_n - A_n$.
- Le nombre de bits s générés à chaque itération est $s = L_n - L_{n-1}$.
- Le codage d'Elias nécessite une précision infinie.

1.3.2 Version de Pasco-Jones du code d'Elias

Résout le problème de la précision infinie nécessaire :

- Les calculs sont réalisés en précision finie sur m bits
 \hookrightarrow variables entières allant de 0 à $2^m - 1$
- Après chaque découpage et sélection d'un intervalle sont réalisés
 - une dilatation de l'intervalle obtenu
 - un arrondi des bornes

Dilatation :

Soient

$$C_n = 2^{L_n+m} A_n$$

et

$$W_n = 2^{L_n+m} (B_n - A_n)$$

où L_n est un entier tel que $2^{m-1} \leq W_n < 2^m$.

\implies dilatation de l'intervalle $[A_n, B_n)$ par 2^{L_n+m} .

Choix de m arbitraire à condition que

1. $p_0 \geq 2^{-m}$
2. $p_1 = 1 - p_0 \geq 2^{-m}$.

Des valeurs très grandes de m permettent d'obtenir une compression des données très proche de l'entropie.

Dilatation et arrondi :

Initialisation : $C_0 = 0$ et $W_0 = 2^{m-1}$.

Itération :

Si le $n^{\text{ème}}$ symbole est 0 :

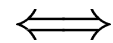
$$\begin{aligned} C_n &= 2^s C_{n-1} \\ W_n &= 2^s \left[W_{n-1} p_0 + \frac{1}{2} \right]. \end{aligned}$$

Si le $n^{\text{ème}}$ symbole est 1 :

$$\begin{aligned} C_n &= 2^s \left(C_{n-1} + \left[W_{n-1} p_0 + \frac{1}{2} \right] \right) \\ W_n &= 2^s \left(W_{n-1} - \left[W_{n-1} p_0 + \frac{1}{2} \right] \right). \end{aligned}$$

L'entier s est choisi afin que $2^{m-1} \leq W_n < 2^m$.

Multiplication par 2^s dans le calcul de C_n



décalage à gauche de s bits

\implies débordement de s bits

\implies bits de code générés à l'itération n et envoyés vers le canal.

Le nombre de bits ayant été envoyés vers le canal est

$$L_n = L_{n-1} + s$$

Initialisation par $L_0 = 0$.

On peut retrouver à chaque n

$$\begin{aligned}A_n &= 2^{-(m+L_n)} C_n \\B_n &= A_n + 2^{-(m+L_n)} W_n.\end{aligned}$$

On peut montrer que :

1. $[A_n, B_n)$ reste un sous-intervalle propre de $[0, 1)$,
2. W_n reste toujours positif,
3. il faut que $p_0 > 2^{-m}$,
4. il faut que $p_1 > 2^{-m}$.

1.3.3 Cas des sources discrètes non binaires

Extension du code d'Elias aisée.

Soit une source d'alphabet

$$\{a_0, \dots, a_{J-1}\}$$

et de densité de probabilité

$$\mathbf{p} = (p_0, p_1, \dots, p_{J-1}).$$

Soit F_j les probabilités cumulées

$$F_j = \sum_{k=0}^{j-1} p_k.$$

Le code d'Elias découpe récursivement les intervalles en J sous-intervalles de taille proportionnelle à p_j .

Code d'Elias :

Initialisation :

$$A_0 = 0 \text{ et } B_0 = 1.$$

Récursion :

Si le $n^{\text{ème}}$ symbole émis est a_j , alors

$$A_n = A_{n-1} + F_j (B_{n-1} - A_{n-1})$$

$$B_n = A_{n-1} + F_{j+1} (B_{n-1} - A_{n-1})$$

Version de Pasco-Jones du code d'Elias :

Initialisation :

$$C_0 = 0 \text{ et } W_0 = 2^m - 1.$$

Récursion :

Si le $n^{\text{ème}}$ symbole émis est a_j , alors

$$C_n = 2^s (C_{n-1} + \lfloor F_j W_{n-1} + 1/2 \rfloor)$$

$$W_n = 2^s (\lfloor F_{j+1} W_{n-1} + 1/2 \rfloor - \lfloor F_j W_{n-1} + 1/2 \rfloor)$$

L'entier s est choisi afin que $2^{m-1} \leq W_n < 2^m$.



1.3.4 Exemple

Une source ayant pour alphabet $\{a_0, a_1\}$ avec $\mathbf{p} = (0.7, 0.3)$.

Elle émet

.01100110001 ...

Codage d'Elias :

1. $[A_0, B_0) = [0, 1)$
2. $[0, 1) \rightarrow [0, 0.7), [0.7, 1)$
 1^{er} symbole est 0 $\implies [A_1, B_1) = [0_{10} = 0.0_2, 0.7_{10} = 0.1011\dots_2)$
3. $[0, 0.7) \rightarrow [0, 0.49), [0.49, 0.7)$
 $2^{\text{ème}}$ symbole est 1 $\implies [A_2, B_2) = [0.49_{10} = 0.0111\dots_2, 0.7_{10} = 0.1011\dots_2)$
4. $[0.49, 0.7) \rightarrow [0.49, 0.637), [0.637, 0.7)$
 $3^{\text{ème}}$ symbole est 1 \implies
 $[A_3, B_3) = [0.637_{10} = 0.1010\dots_2, 0.7_{10} = 0.1011\dots_2)$
 \implies **101** peut être envoyé dans vers le canal
5. $[0.637, 0.7) \rightarrow \dots$

Décodage d'Elias :

1.4 Régulation du débit

Codage à longueur variable



Problème de gestion du débit à l'émission

Structure de régulation permet d'obtenir un débit régulier :

1.5 Codes universels

Codes précédants nécessitent un modèle de la source connu du codeur et du décodeur.

Codes universels :

- estimation des statistiques de la source,
- initialisation peut être mauvaise,
- convergence doit être rapide

1.5.1 Codeur de Liv-Zempel-Welch

Construction progressive d'un dictionnaire

↔ contient les chaînes émises par la source

Dictionnaire construit à partir de la source



Bonne image des statistiques de la source

Initialisation : chaque symbole de l'alphabet est placé dans le dictionnaire

Récursion :

1. On recherche la plus longue chaîne du dictionnaire correspondant aux premiers symboles se trouvant dans le flot d'entrée.
2. L'index dans le dictionnaire de la chaîne ainsi trouvée sert de code
3. La chaîne trouvée est complétée du symbole suivant dans le flot d'entrée et placée dans le dictionnaire
4. Le processus est itéré

1.5.2 Exemple

On considère une source d'alphabet $\{A, B, C\}$ et un dictionnaire de taille 8

La source a émis :

ABCCABCBA...

Dictionnaire à l'itération 0 : *A/B/C*

Chaîne traitée à l'itération 1 : $A \implies$ envoi de 0000 dans le canal

Dictionnaire à l'itération 1 : *A/B/C/AB*

Symboles restant à traiter : *BCCABCBA...*

Chaîne traitée à l'itération 2 : $B \implies$ envoi de 0001 dans le canal

Dictionnaire à l'itération 2 : *A/B/C/AB/BC*

Symboles restant à traiter : *CCABCBA...*

Chaîne traitée à l'itération 3 : $C \implies$ envoi de 0010 dans le canal

Dictionnaire à l'itération 3 : *A/B/C/AB/BC/CC*

Symboles restant à traiter : *CABCBA...*

Initialisation : chaque symbole de l'alphabet est placé dans le dictionnaire

Récursion :

1. A l'aide de l'index transmis, on retrouve le mot codé dans le dictionnaire
ABCX
2. La première lettre de ce mot *A* sert de postfixe au mot codé à l'itération précédente *XCAB*
↔ le mot *XCABA* est placé dans le dictionnaire
3. Le processus est itéré

Remarque :

Un problème peut apparaître au décodage de séquence du type

$$X\omega X\omega X$$

où X est un symbole donné et ω est une chaîne.

Pour le codeur, on suppose que $X\omega$ appartient au dictionnaire.

$X\omega$ est traité et $X\omega X$ est placé dans le dictionnaire.

A l'itération suivante, $X\omega X$ est traité et l'index dans le dictionnaire est envoyé.

Pour le décodeur, $X\omega$ est décodé correctement.

Par contre l'adresse suivante ne correspondra à rien car $X\omega X$ n'aura pas été placé dans le dictionnaire

↪ Indique au décodeur que la chaîne à décoder commence de la même manière que la précédente.

1.5.3 Conclusions

1. Codes longueur variable \longrightarrow longueur fixe.
2. Performances excellents lorsque le dictionnaire est long.
3. Problème de dépassement de capacité du dictionnaire
 - Stratégies identiques pour le codeur et le décodeur doivent être employées. Exemple : remplacement des symboles les moins utilisés par les nouveaux symboles.
4. Possibilité de réaliser un codage adaptatif lorsque les statistiques de la source varient.

2 Codage source des sources continues

Sources continues servent à modéliser la sortie des codeurs à transformée

:

- Transformée en cosinus discrete (JPEG, H263, MPEG...)
- Transformée en ondelette (JPEG 2000)
- Transformée de Karhunen-Loeve
- ...

Problèmes soulevés :

- comment associer une quantité d'information à de telles sources ?
- compression \longleftrightarrow quantification ?
- comment rendre la compression optimale ?
- optimale en quel sens ?
- bornes sur les performances atteignables ?

2.1 Entropie différentielle d'une source continue

Soit une variable aléatoire *continue* X , de distribution de probabilité $f_X(x)$.

Entropie *différentielle* de X

$$h(X) = - \int_{-\infty}^{+\infty} f_X(x) \log_2(f_X(x)) dx$$

A distinguer impérativement de l'entropie ordinaire ou *absolue*.

On peut considérer la variable aléatoire X comme la limite d'une variable aléatoire discrete $X_{\Delta x}$ prenant les valeurs

$$a_k = k.\Delta x, k \in \mathbb{Z}$$

avec $\Delta x \longrightarrow 0$.

Pour Δx petit, on a

$$p_k = \Pr (a_k \leq X < a_k + \Delta x) = f_X (a_k) .\Delta x.$$

Entropie absolue de la source continue X :

$$\begin{aligned} H (X) &= \lim_{\Delta x \rightarrow 0} H (X_{\Delta x}) . \\ &= \lim_{\Delta x \rightarrow 0} - \sum_{k=-\infty}^{k=+\infty} p_k \log_2 p_k \end{aligned}$$

$$\begin{aligned}
 H(X) &= \lim_{\Delta x \rightarrow 0} - \sum_{k=-\infty}^{k=+\infty} p_k \log_2 p_k \\
 &= \lim_{\Delta x \rightarrow 0} - \sum_{k=-\infty}^{k=+\infty} f_X(a_k) \Delta x \log_2 (f_X(a_k) \Delta x)
 \end{aligned}$$

$$\begin{aligned}
 H(X) &= \lim_{\Delta x \rightarrow 0} - \sum_{k=-\infty}^{k=+\infty} f_X(x_k) \Delta x \log_2 (f_X(x_k) \Delta x) \\
 &= \lim_{\Delta x \rightarrow 0} - \sum_{k=-\infty}^{k=+\infty} f_X(a_k) \log_2 (f_X(a_k)) \Delta x - \log_2 \Delta x \sum_{k=-\infty}^{k=+\infty} f_X(a_k) \Delta x \\
 &= - \int_{-\infty}^{+\infty} f_X(x) \log_2 (f_X(x)) dx - \lim_{\Delta x \rightarrow 0} \log_2 \Delta x \int_{-\infty}^{+\infty} f_X(x) dx \\
 &= h(X) - \lim_{\Delta x \rightarrow 0} \log_2 \Delta x.
 \end{aligned}$$

Tend vers $+\infty$, d'où le terme entropie différentielle pour $h(X)$.

Attention, $h(X)$ peut être positive ou négative.

Pour un vecteur aléatoire \mathbf{X} composé de n variables aléatoires continues X_1, X_2, \dots, X_n , l'entropie différentielle est

$$h(\mathbf{X}) = - \int_{-\infty}^{+\infty} f_{\mathbf{X}}(\mathbf{x}) \log_2(f_{\mathbf{X}}(\mathbf{x})) d\mathbf{x}$$

où $f_{\mathbf{X}}(\mathbf{x})$ est la distribution de probabilité jointe de \mathbf{X} .

Exemple 5 Soit X uniformément distribuée sur $[0, a]$. Montrer que

$$h(X) = \log a.$$

Exemple 6 Soit X une variable aléatoire gaussienne de densité de probabilité

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right).$$

Montrer que

$$h(X) = \frac{1}{2} \log_2(2\pi e\sigma^2),$$

ou e est le nombre d'Euler.

Exemple 7 Soit X et Y deux variables aléatoires continues, de même moyenne μ et de même variance σ^2 . En outre, on suppose que X est gaussienne

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

En utilisant la propriété

$$\int_{-\infty}^{+\infty} f_Y(x) \log_2\left(\frac{f_X(x)}{f_Y(x)}\right) dx \leq 0,$$

montrer que l'entropie différentielle de Y est bornée par l'entropie différentielle de la variable aléatoire X .

Modèle gaussien

\Leftrightarrow

Modèle *conservatif* d'une source de distribution mal connue

Information mutuelle entre deux variables aléatoires continues X et Y

$$I(X; Y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f_{XY}(x, y) \log_2 \left(\frac{f_X(x|y)}{f_X(x)} \right) dx dy.$$

Propriétés :

1. $I(X; Y) = I(Y; X)$
2. $I(X; Y) \geq 0$
3. $I(X; Y) = h(X) - h(X|Y)$
4. $I(X; Y) = h(Y) - h(Y|X)$

où

$$h(X|Y) = - \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f_{XY}(x, y) \log_2 (f_X(x|y)) dx dy$$

2.2 Courbe débit-distorsion

Compression de sources continues :

Représentation d'une variable aléatoire *continue* X

par

une variable aléatoire *discrete* \tilde{X}

\Leftrightarrow

Quantification

Mesure de *distorsion* nécessaire
pour mesurer la perte de qualité

Comment choisir les valeurs prises par la source discrète ?

Mesure de distorsion permet de faire ce choix.

Exemples de mesures de distorsion :

- mesure de distorsion quadratique

$$d(x_k, \tilde{x}_k) = (x_k - \tilde{x}_k)^2$$

- autres mesures sont de la forme

$$d(x_k, \tilde{x}_k) = |x_k - \tilde{x}_k|^p, \text{ avec } p > 0.$$

Ces mesures de distorsion s'appliquent à un symbole unique.

Pour une source \mathbf{X}_k générant un vecteur de n symboles, la mesure de distorsion par rapport à cette source quantifiée $\tilde{\mathbf{X}}_k$

$$d(\mathbf{X}_k, \tilde{\mathbf{X}}_k) = \frac{1}{n} \sum_{k=1}^n d(x_k, \tilde{x}_k).$$

La *distorsion* est l'espérance de la mesure de distorsion

$$D = E(d(\mathbf{X}_k, \tilde{\mathbf{X}}_k)) = \frac{1}{n} \sum_{k=1}^n E(d(x_k, \tilde{x}_k)).$$

Si en plus, la source est stationnaire

$$D = E(d(x, \tilde{x})).$$

Soient

- une source continue sans mémoire X de distribution de probabilité $p(x)$,
- une source discrète \tilde{X} , obtenue en quantifiant X ,
- une mesure de distorsion par symbole

$$d(x, \tilde{x})$$

où x et \tilde{x} sont des valeurs possibles pour X et \tilde{X} .

La quantité minimale de bits nécessaire pour représenter la sortie de X avec une distorsion inférieure ou égale à D est

$$R(D) = \min_{p(\tilde{x}|x) \text{ tq } E(d(X, \tilde{X})) \leq D} I(X; \tilde{X}) \quad (\text{en bits/symbole})$$

$R(D)$ décroît lorsque D croît
ou
 $D(R)$ croît lorsque R décroît.

Théorème 5 (Shannon, 1959) *Le débit minimal d'information pour représenter une source gaussienne discrète et sans mémoire de variance σ_X^2 en fonction de la distorsion D est*

$$R_g(D) = \begin{cases} \frac{1}{2} \log_2 (\sigma_X^2 / D) & \text{si } 0 \leq D \leq \sigma_X^2 \\ 0 & \text{si } D > \sigma_X^2 \end{cases}$$

D étant évaluée avec une mesure de distorsion quadratique.

Si $D \geq \sigma_X^2$, aucune information ne doit être transmise

↔ au décodeur source, reconstruction du signal par des 0.

Si $D > \sigma_X^2$, idem

↔ au décodeur source, reconstruction du signal par une variable aléatoire gaussienne de variance $D - \sigma_X^2$.

Théorème 6 (Shannon, 1959) *Il existe un schéma de codage transformant la sortie d'une source en mots de codes, tel que pour toute distorsion D , un débit de $R(D)$ bits par symbole (échantillon) soit suffisant pour reconstituer la source initiale avec une distorsion arbitrairement proche de D .*

Courbe débit-distorsion $R(D)$

\Leftrightarrow

Borne inférieure du débit R atteignable
pour une certaine distorsion D .

Inversion de la courbe débit-distorsion



Courbe *distorsion-débit*

Dans le cas d'une source gaussienne sans mémoire

$$R_g(D) = \frac{1}{2} \log_2 (\sigma_X^2 / D)$$

d'où

$$D_g(R) = 2^{-2R} \sigma_X^2.$$

En prenant le logarithme

$$10 \log_{10} D_g(R) = -6R + 10 \log_{10} \sigma_X^2$$

⇒ la distorsion (l'erreur quadratique moyenne) décroît de 6 dB/bit.

Pas de résultats explicites pour les courbes débit-distorsion
de sources autres que la source gaussienne

Mais encadrements possibles.

Théorème 7 *La courbe débit-distorsion $R(D)$ d'une source à valeurs continues sans mémoire de moyenne nulle et de variance σ_X^2 , où D est une distorsion basée sur une mesure de distorsion quadratique est bornée*

$$R(D) \leq R_g(D) = \frac{1}{2} \log_2 (\sigma_X^2 / D), \quad 0 \leq D \leq \sigma_X^2.$$

Pour représenter une source avec une distorsion donnée, la source gaussienne sera celle pour laquelle le nombre de bits par symbole sera le plus important.

Si on considère la courbe distorsion-débit,

$$D(R) \leq D_g(R) = 2^{-2R} \sigma_X^2.$$

Une borne inférieure existe également.

Pour le même type de source,

$$R(D) \geq R^*(D) = h(X) - \frac{1}{2} \log_2 2\pi e D,$$

où $h(X)$ est l'entropie différentielle de la source.

Si on considère la courbe distorsion-débit,

$$D(R) \geq D^*(R) = \frac{1}{2\pi e} 2^{-2(R-h(X))}.$$

En prenant le logarithme

$$10 \log_{10} D^*(R) = -6R + 6h(X) - 10 \log_{10} 2\pi e$$

\implies la borne inférieure distorsion décroît également de 6 dB/bit.

En résumé :

$$R^*(D) \leq R(D) \leq R_g(D).$$

et

$$D^*(R) \leq D(R) \leq D_g(R)$$

$D^*(R)$ et $D_g(R)$ ont une décroissance de 6dB/bit.

$p(x)$	$h(X)$	$R_g - R^*$ (bit/symbole)	$D_g - D^*$ (dB)
$\frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2}$	$\frac{1}{2} \log_2 (2\pi e\sigma^2)$	0	0
$\frac{1}{2\sqrt{3}\sigma}, x \leq \sqrt{3}\sigma$	$\frac{1}{2} \log_2 (12\sigma^2)$	0.255	1.53
$\frac{1}{\sqrt{2}\sigma} e^{-\sqrt{2} x /\sigma}$	$\frac{1}{2} \log_2 (2e^2\sigma^2)$	0.104	0.62
$\frac{\sqrt[4]{3}}{\sqrt{8\pi\sigma x }} e^{-\sqrt{3} x /2\sigma}$	$\frac{1}{2} \log_2 (4\pi e^{0.423}\sigma^2/3)$	0.709	4.25

Tableau comparatif des encadrements de quelques courbes débit-distorsion et distorsion-débit pour différent types de sources

2.3 Quantification scalaire

Dispositif de quantification



Compression avec perte du signal d'entrée.

Entrée :

- source X de distribution de probabilité $p(x)$,
- nombre de niveaux de sortie $L = 2^R$.

Sortie :

- seuils de décision $\{x_k\}_0^L$
- niveaux de sortie $\{\tilde{x}_k\}_1^L$.

Problème de la quantification optimale (minimisant la distorsion)

⇓

Choix de $\{x_k\}_0^L$ et de $\{\tilde{x}_k\}_1^L$.

Distorsion résultant de la quantification

$$D = \int_{-\infty}^{+\infty} d(\tilde{x}(x), x) p(x) dx,$$

où

$\tilde{x}(x)$ est la sortie quantifiée associée à x

$d(\tilde{x}(x), x) = f(\tilde{x}(x) - x)$ est une mesure de distorsion.

Problème traité par Lloyd, 1982 et Max, 1960

\implies quantificateur de Lloyd-Max

2.3.1 Quantification scalaire uniforme

Niveaux de sortie du quantificateur sont uniformément répartis.

Le niveau de sortie

$$\tilde{x}_k = \frac{1}{2} (2k - 1) \Delta$$

est associé à une entrée

$$(k - 1) \Delta \leq x < k\Delta,$$

où Δ est le pas d'échantillonnage.

On a alors

$$\begin{aligned}
 D &= \int_{-\infty}^{+\infty} d(\tilde{x}(x), x) p(x) dx \\
 &= 2 \sum_{k=1}^{L/2-1} \int_{(k-1)\Delta}^{k\Delta} f\left(\frac{1}{2}(2k-1)\Delta - x\right) p(x) dx \\
 &\quad + 2 \int_{(L/2-1)\Delta}^{+\infty} f\left(\frac{1}{2}(2k-1)\Delta - x\right) p(x) dx.
 \end{aligned}$$

On cherche à minimiser D . Ici, le seul paramètre de réglage est Δ .

On calcule la dérivée de D par rapport à Δ :

$$\begin{aligned}
 \frac{\partial D}{\partial \Delta} &= \sum_{k=1}^{L/2-1} (2k-1) \int_{(k-1)\Delta}^{k\Delta} f'\left(\frac{1}{2}(2k-1)\Delta - x\right) p(x) dx \\
 &\quad + (L-1) \int_{(L/2-1)\Delta}^{+\infty} f'\left(\frac{1}{2}(2k-1)\Delta - x\right) p(x) dx
 \end{aligned}$$

Choix de $f(x)$, par exemple $f(x) = x^2$

⇓

Calcul numérique de Δ_{opt} en résolvant $\frac{\partial D}{\partial \Delta}(\Delta_{\text{opt}}) = 0$.

Nombre de niveaux de sortie	Pas de quantification optimal Δ_{opt}	Erreur quadratique minimale D_{min}	$10 \log D_{\text{min}}$ (dB)
2	1.5960	0.3634	-4.4
4	0.9957	0.1188	-9.25
8	0.5860	0.03744	-14.27
16	0.3352	0.01154	-19.38
32	0.1881	0.00349	-24.57

Δ_{opt} pour une source gaussienne
de moyenne nulle et de variance $\sigma_X^2 = 1$.

La décroissance de $10 \log D_{\min}$ est d'environ 5.2 dB/bit



sous-optimal par rapport à la décroissance de $10 \log D_g(R)$

La contrainte d'un pas de quantification fixe est trop forte



Quantification scalaire à pas variable.

2.3.2 Quantification scalaire non uniforme

Niveaux de sortie du quantificateur ne sont plus uniformément répartis.

Le niveau de sortie

$$\tilde{x}_k, k = 1, \dots, L$$

est associé à une entrée

$$x_{k-1} \leq x < x_k,$$

avec en plus

$$x_0 = -\infty \text{ et } x_L = +\infty$$

Dans ces conditions, la distorsion est

$$D = \sum_{k=1}^L \int_{x_{k-1}}^{x_k} f(\tilde{x}_k - x) p(x) dx.$$

Minimisation de D faite en ajustant $\{x_k\}_0^L$ et $\{\tilde{x}_k\}_1^L$.

Conditions *nécessaires* pour que D soit minimale

$$\frac{\partial D}{\partial x_k} = 0 \implies f(\tilde{x}_k - x_k) = f(\tilde{x}_{k+1} - x_k), \quad k = 1, \dots, L-1$$

et

$$\frac{\partial D}{\partial \tilde{x}_k} = 0 \implies \int_{x_{k-1}}^{x_k} f'(\tilde{x}_k - x) p(x) dx = 0, \quad k = 1, \dots, L.$$

Dans le cas particulier où $f(x) = x^2$:

– la première condition

$$f(\tilde{x}_k - x_k) = f(\tilde{x}_{k+1} - x_k), \quad k = 1, \dots, L - 1$$

devient

$$x_k = \frac{1}{2} (\tilde{x}_k + \tilde{x}_{k+1}), \quad 1, \dots, L - 1$$

Les seuils de décision sont au milieu des niveaux de sortie.

– la seconde condition

$$\int_{x_{k-1}}^{x_k} f'(\tilde{x}_k - x) p(x) dx = 0, \quad k = 1, \dots, L$$

devient

$$\int_{x_{k-1}}^{x_k} (\tilde{x}_k - x) p(x) dx = 0$$

\tilde{x}_k est alors le centroïde de la région de $p(x)$ entre x_{k-1} et x_k
 \implies résolution numérique possible quel que soit $p(x)$.

Procédure doit être répétée itérativement à partir, par exemple, de x_k répartis uniformément.

Pour une source de distribution de probabilité gaussienne de variance unité, on obtient

– avec $L = 4$

k	x_k	\tilde{x}_k
1	-0.9816	-1.510
2	0.0	-0.4528
3	0.9816	0.4528
4	$+\infty$	1.510

$$L = 4 : D_{\min} = 0.1175, 10 \log D_{\min} = -9.3\text{dB}$$

– avec $L = 5$

k	x_k	\tilde{x}_k
1	-1.748	-2.152
2	-1.050	-1.344
3	-0.5006	-0.7560
4	0	-0.2451
5	0.5006	0.2451
6	1.050	0.7560
7	1.748	1.344
8	$+\infty$	2.152

$$L = 8 : D_{\min} = 0.03454, 10 \log D_{\min} = -14.62\text{dB}$$

Enfin, si l'on compare quantificateur uniforme et non-uniforme

R (bits/symbole)	$10 \log_{10} D_{\min}$	
	Uniforme (dB)	Non uniforme (dB)
1	-4.4	-4.4
2	-9.25	-9.3
3	-14.27	-14.62
4	-19.38	-20.22
5	-24.57	-26.02
6	-29.83	-31.89
7	-35.13	-37.81

Pour des hauts débits, la décroissance de $10 \log_{10} D_{\min}$ est proche de 6dB/bit

Comparaison par rapport à la courbe débit-distorsion

2.3.3 Codage entropique

Sortie d'un quantificateur non uniforme est une source discrete \tilde{X} .

Symboles émis par cette source

$$\{\tilde{x}_1, \dots, \tilde{x}_L\},$$

probabilités associées

$$\{p_1, \dots, p_L\}.$$

Entropie de la source discrete

$$H(\tilde{X}) = \sum_{k=1}^L p_k \log_2 p_k.$$

\implies codage de Huffman permet de diminuer R pour un même D_{\min} .

Exemple 8 *Pour la quantification non-uniforme avec $L = 4$ vue précédemment*

$$p_1 = p_4 = 0.1635$$

$$p_2 = p_3 = 0.3365$$

d'où

$$H(\tilde{X}) = 1.911 \text{ bits/symbole}$$

Par un codage de Huffman efficace (sur des blocs de symboles), il est possible d'obtenir

$$10 \log_{10} D_{\min} = -9.3 \text{ dB avec } R = 1.911 \text{ bits/symbole.}$$

R (bits/symbole)	Entropie (bits/symbole)	Distorsion $10 \log_{10} D_{\min}$
1	1.0	-4.4
2	1.911	-9.3
3	2.825	-14.62
4	3.765	-20.22
5	4.730	-26.02

Codage entropique

=

Quantification scalaire non uniforme + Huffman

Comparaison par rapport à la courbe débit-distorsion

Les performances sont meilleures, mais un gain peut encore être obtenu.

2.4 Quantification vectorielle

Limitations de la quantification scalaire

- écart de 2dB par rapport à la courbe débit-distorsion,
- pour des bas débits ($< 1\text{bit/symbole}$), plus rien n'est transmis,
- nécessité de transmettre le dictionnaire

\implies quantification vectorielle

=

quantification *simultanée* de vecteurs

2.4.1 Formulation du problème

Vecteur aléatoire à valeurs continues

$$\mathbf{X} = \{X_1, X_2, \dots, X_n\}$$

dont la distribution de probabilité jointe est

$$p(x_1, x_2, \dots, x_n).$$

Quantification de \mathbf{X} en un vecteur

$$\tilde{\mathbf{X}} = \{\tilde{X}_1, \tilde{X}_2, \dots, \tilde{X}_n\}$$

$$\tilde{\mathbf{X}} = Q(\mathbf{X})$$

Quantification vectorielle \Leftrightarrow Classification

Partition de l'espace en cellules $\{C_k\}$ Tout vecteur $\mathbf{X} \in C_k$ quantifié en $\tilde{\mathbf{x}}_k$

Recherche de la cellule $\tilde{\mathbf{X}}$
dans laquelle se trouve \mathbf{X}

Exemple dans le cas 2D

$$\mathbf{X} = \{X_1, X_2\}$$

$L = 37$ vecteurs de sortie possibles

$$\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_L\}$$

2.4.2 Distorsion

Mesure de distorsion indispensable pour mesurer la qualité de la quantification vectorielle.

Distorsion quadratique :

$$\begin{aligned}d_2(\mathbf{X}, \tilde{\mathbf{X}}) &= (\mathbf{X} - \tilde{\mathbf{X}})^T (\mathbf{X} - \tilde{\mathbf{X}}) \\ &= \frac{1}{n} \sum_{k=1}^n (x_k - \tilde{x}_k)^2\end{aligned}$$

Autres mesures de distorsion :

$$\begin{aligned}d_p(\mathbf{X}, \tilde{\mathbf{X}}) &= (\mathbf{X} - \tilde{\mathbf{X}})^T (\mathbf{X} - \tilde{\mathbf{X}}) \\ &= \frac{1}{n} \sum_{k=1}^n |x_k - \tilde{x}_k|^p\end{aligned}$$

avec $p > 0$. Le cas $p = 1$ est le plus souvent utilisé après $p = 2$.

Dans ces conditions, en supposant qu'une partition $\{C_k\}$ de l'espace est faite

$$\begin{aligned} D &= \sum_{k=1}^L P(\mathbf{X} \in C_k) E \left[d(\mathbf{X}, \tilde{\mathbf{X}}) \mid \mathbf{X} \in C_k \right] \\ &= \sum_{k=1}^L P(\mathbf{X} \in C_k) \int_{\mathbf{X} \in C_k} d(\mathbf{X}, \tilde{\mathbf{X}}) p(\mathbf{X}) d\mathbf{X} \end{aligned}$$

Distorsion minimisée en construisant correctement

- les cellules C_k
- les points $\tilde{\mathbf{X}}_k$

2.4.3 Conditions d'optimalité

On considère un quantificateur vectoriel

- à L vecteurs de sortie,
- réalisant une partition C_k , $1 \leq k \leq L$.

Première condition nécessaire d'optimalité :

$$Q(\mathbf{X}) = \tilde{\mathbf{X}}_k$$

si et seulement si

$$d(\mathbf{X}, \tilde{\mathbf{X}}_k) \leq d(\mathbf{X}, \tilde{\mathbf{X}}_\ell), \text{ pour tout } \ell \neq k, 1 \leq \ell \leq L$$

\implies sélection du plus proche voisin.

Seconde condition nécessaire d'optimalité :

$\tilde{\mathbf{X}}_k$ doit minimiser la distorsion dans la cellule C_k

↓

$\tilde{\mathbf{X}}_k$ doit minimiser

$$\begin{aligned} D_k(\tilde{\mathbf{X}}) &= E \left[d(\mathbf{X}, \tilde{\mathbf{X}}) \mid \mathbf{X} \in C_k \right] \\ &= \int_{\mathbf{X} \in C_k} d(\mathbf{X}, \tilde{\mathbf{X}}) p(\mathbf{X}) d\mathbf{X} \end{aligned}$$

$\implies \tilde{\mathbf{X}}_k$ est le centroïde (*barycentre*) de C_k

Lorsque $p(\mathbf{X})$ est connue, construction possible de $\{C_k\}$ et $\tilde{\mathbf{X}}_k$.

Si on pose

$$D_n(R) = \min_{Q(\mathbf{X})} E \left(d \left(\mathbf{X}, \tilde{\mathbf{X}} \right) \right),$$

où

$$R \geq \frac{H(\tilde{\mathbf{X}})}{n}$$

alors

$$\lim_{n \rightarrow \infty} D_n(R) = D(R).$$

Il est possible avec une quantification vectorielle de haute dimension d'approcher arbitrairement près de la courbe débit-distorsion.

2.4.4 La quantification vectorielle en pratique

Distribution de probabilité pas toujours disponible



algorithmes sous-optimaux

K-moyennes (ou LBG : Linde, Buzo, Gray) :

Ensemble de vecteurs d'apprentissage $\{\mathbf{X}(m)\}_{m=1}^M$ supposé disponible, avec $M \gg L$

\Rightarrow construction itérative de $\{C_k\}$ et $\tilde{\mathbf{X}}_k$.

Algorithme des K -moyennes :

1. Initialiser $i = 0$. Choisir L vecteurs $\left\{ \tilde{\mathbf{X}}_k(0) \right\}_{k=1}^L$ dans $\{\mathbf{X}(m)\}_{m=1}^M$.
2. Ranger les $\{\mathbf{X}(m)\}_{m=1}^M$ dans des cellules C_k en appliquant la règle du plus proche voisin

$$\mathbf{X} \in C_k(i) \text{ ssi } d(\mathbf{X}, \tilde{\mathbf{X}}_k(i)) \leq d(\mathbf{X}, \tilde{\mathbf{X}}_j(i)), \text{ pour tout } k \neq j$$

3. Faire $i = i + 1$. Recalculer pour chaque cellule définie au pas

précédent, les centroïdes

$$\tilde{\mathbf{X}}_k(i) = \frac{1}{M_k} \sum_{m, X(m) \in C_k} X(m), 1 \leq k \leq L.$$

4. Calculer la distorsion moyenne $D(i)$.
5. Aller en 2 tant que $D(i-1) - D(i) > \varepsilon$.

Les vecteurs $\{\tilde{\mathbf{X}}_k\}$ ainsi calculés forment le *dictionnaire*.

Pour la quantification, \mathbf{X} devra être comparé à tous les éléments du dictionnaire

\implies recherche exhaustive.

Complexité calculatoire

$$C = nL$$

par vecteur d'entrée.

Si L est une puissance de 2, alors $\log_2 L$ est le nombre de bits nécessaire à la représentation de chaque vecteur.

Si R est le nombre de bit par composante, $\log_2 L = nR$, d'où

$$C = n2^{nR}.$$

\implies complexité augmente exponentiellement avec n et R .

Par ailleurs, le dictionnaire doit être transmis pour pouvoir réaliser la quantification inverse.

2.4.5 Alternatives

- Quantification vectorielle en arbre
 - arbre uniforme
 - arbre non-uniforme
- ⇒ résout le problème de complexité, sous-optimal
- Quantification à base de *lattice*
- ⇒ résout le problème de complexité, plus de dictionnaire à transmettre, sous-optimal
- ...

Bibliographie

J. B. Anderson, *Digital Transmission Engineering*, Prentice Hall, Upper Saddle River, 1998.

R. E. Blahut, *Digital Transmission of Information*, Addison-Wesley, Reading, 1990.

R.G. Gallager, *Information Theory and Reliable Communication*, McGraw-Hill, New-York, 1968.

S. Lin and D. J. Costello, Jr., *Error Control Coding*, Englewood Cliffs, NJ, 1983.

J. B. Anderson and S. Mohan, *Source and Channel Coding*, Kluwer, Boston, 1993.

J. G. Proakis, *Digital Communication, 4th edition*, Mc Graw Hill, Boston, 2000.