Théorie de l'information et applications

TPs d'Agnès Desolneux, Eva Wesfreid, Sébastien Li Thiao Té

Jean-Michel Morel.

September 1, 2009

Introduction

Notre civilisation a rendu trois types de données numériques omniprésents dans la communication humaine: les sons digitaux, les images digitales et les données alpha-numériques (les textes). Ils représentent désormais trois des cinq vecteurs principaux de la communication humaine. (Les deux autres sont la conversation et la gestuelle, quand les communiquants sont en présence l'un de l'autre). Les recherches sur les trois modes de représentation digitaux, très complexes, ayant chacun sa structure propre, ne font encore que commencer. Toutefois, la théorie de l'information et l'analyse de Fourier ont permis de délimiter un champ scientifique suffisant pour qu'on puisse, entre autres, traiter très rigoureusement d'un des problèmes les plus cruciaux: la compression des données transmises sous l'une quelconque de ces trois formes.

Dans ce cours, nous allons présenter toutes les bases mathématiques utiles pour comprendre comment un texte, une image, ou un son digitaux sont créés, quelles sont les distorsions inhérentes à la nature de l'image ou du son digital d'une part, et celles qui sont entraînées par une "mauvaise" digitalisation, enfin comment mesurer la quantité d'information qu'elles contiennent et enfin comment les comprimer.

Le cours ira de la théorie à la pratique la plus commune. On commencera avec l'analyse de Fourier, les ondelettes, et la théorie de l'information et de la communication de Shannon.

Et on en arrivera à détailler les formats de compression les plus courants en informatique et sur le web pour textes, images et sons.

La théorie de la communication de Shannon, publiée en 1948, rebaptisée ensuite théorie de l'information est le texte fondateur, que nous analyserons en détail. Cette théorie propose une chaîne complète pour coder, comprimer et transmettre tous messages entre humains, et particulièrement les signaux (voix et images). Les théories mathématiques sous-jacentes sont d'une part la théorie des probabilités discrètes à laquelle s'ajoute une notion issue de la thermodynamique des gazs, la notion d'entropie et d'autre part l'analyse de Fourier, qui permet de formaliser le passage d'un signal ou d'une image vus comme des fonctions à leur représentation discrète par échantillonnage.

La théorie de Shannon: compression sans perte : les textes

Nous traiterons d'abord de la théorie de Shannon, qui définit la quantité d'information contenue dans un message et traite de sa compression optimale. La notion centrale de cette partie est celle d'entropie. Les codages les plus usités: Shannon, Huffman, Lempel-Ziv, seront expliqués, démontrés mathématiquement, et testés.

Des expériences sur des textes de divers types seront menées pour vérifier la validité des algorithmes de compression, et mener quelques expériences de synthèse automatique de texte et de comparaison entropique de textes variés.

Les images

Bien que notre civilisation ait multiplié la présence des signaux, images et sons, et surtout des images et sons digitaux (ou numériques), leur nature est mal connue du public, et même des spécialistes. En effet, ceux-ci sont rarement à la source de l'image : ils ne savent pas comment l'image a été enregistrée, transmise, comprimée. La vision correcte de l'image demande une connaissance approfondie de la structure des images digitales et de toutes les distorsions

entrainées par leur caractère d'images digitales. Dans ce texte, nous allons présenter toutes les bases mathématiques utiles pour comprendre comment une image ou un son digitaux sont créés, quelles sont les distorsions inhérentes à la nature de l'image ou du son digital d'une part, et celles qui sont entrainées par une "mauvaise" digitalisation. Nous décrirons ensuite les méthodes classiques et nouvelles de restauration, c'est-à-dire les méthodes qui visent, partant d'un signal abimé, à retrouver une image conforme à une acquisition correcte. Nous commencerons par un long exposé des notions d'analyse de Fourier nécessaires pour comprendre l'échantillonnage de l'image, i.e. sa réduction à un tableau fini de valeurs numériques. Nous expliquerons la théorie de Shannon, qui fixe les règles d'échantillonnage correct, et nous décrirons les manipulations élémentaires que permet l'usage correct de cette théorie : translation, rotation et zoom de l'image notamment. Toutes ces manipulations seront illustrées d'exemples réalistes. Ensuite, nous aborderons les distorsions "nécessaires", celles qu'entraine la nature même des images numériques, à savoir le phénomène de Gibbs ou "ringing", la quantification et le flou. Pour chacun de ces phénomènes, nous montrerons aussi des exemples, et nous traiterons ensuite le problème de la restauration, c'est-à-dire de l'élimination des distorsions, notamment quand elles sont plus poussées que ne le permet la théorie de Shannon (trop de bruit, trop de flou, trop de quantification, trop d'aliasing...) Enfin, utilisant les éléments d'analyse de Fourier détaillés l'an dernier et la théorie de Shannon, en arrivera à expliquer en détail comment marche l'algorithme de compression d'images du web: l'algorithme JPEG (format .jpg de toutes les images courantes).

Les sons

La dernière partie en vient au son et donne les éléments du vaste programme de décomposition d'un son en atomes temps-fréquence, ou "notes". L'outil principal est la transformée de Fourier à fenêtre, avec une ouverture sur la théorie des ondelettes.

Le cours

Principe du cours

Le but de ce cours est donc de donner les outils mathématiques, mais toujours en les reliant à des expériences visuelles ou auditives permettant au lecteur de voir l'effet des opérations sur les textes, images et signaux.

Un polycopié sera distribué.

Les travaux pratiques associés à ce cours sont essentiels et mettront en oeuvre pratiquement, sur des images, des sons et des textes, toutes les notions introduites.

La note sera basée sur un rapport de travaux pratiques et un devoir où les élèves devront résoudre les exercices mathématiques les plus illustratifs.

Déroulement du cours

Le cours dure 16 heures suivies de 16 heures de travaux dirigés. Le cours ne demande que des connaissances de licence (séries de Fourier et probabilité discrète). Ces notions sont de toutes façons rappelées dans le polycopié. Son but est de décrire une des théories mathématiques majeures du XX-ème siècle, la théorie de la communication de Shannon.

- Premier et deuxième cours de quatre heures qui introduisent aux notions d'entropie et d'entropie relative d'une variable aléatoire discrète et expliquent la théorie de la communication de Shannon. Après une explication du modèle markovien du langage, le codage optimal théorique par la méthode du décompte des messages typiques est démontré. Ensuite, cette méthode est étendue aux couples entrée-sorties typiques pour démontrer le grand théorème de Shannon, à savoir l'existence de communication sûre malgré le bruit. Enfin un algorithme de codage universel, Lempel-Ziv, sera expliqué.

- Troisième cours de quatre heures. Dans les deux premières heures on revient sur la théorie de l'information en donnant la théorie des codes préfixe, l'inégalité de Kraft et en prouvant l'optimalité du code de Huffman et la quasi-optimalité du code de Shannon. Dans la deuxième partie on revient sur la théorie de l'échantillonnage en expliquant la transformée de Fourier à fenêtre et sa variante orthogonale, les ondelettes de Malvar-Wilson.

- Quatrième cours de quatre heures. Entièrement consacré à la réduction du continu au discret, à savoir la théorie de l'échantillonnage de Shannon appliquée aux signaux et images. Le lien entre transformée de Fourier discrète et série de Fourier est mis au clair par la formule d'aliasage. L'utilisation de la FFT pour diverses manipulations d'images est ensuite décrite (zoom sans aliasage, translation, rotation). Le phénomène de Gibbs sera aussi commenté. La représentation en Fourier permet un premier abord des questions de débruitage, de déflouage, et en général de filtrage des images. Le standard de compression JPEG sera décrit en détail, car il présente un usage intégré de toutes les notions de traitement d'images et de théorie de l'information introduites précédemment.

- Les quatres travaux pratiques illustrent directement le cours : après une introduction à Matlab, génération de phrases par modèle markovien, entropie d'une phrase et compression par le code de Huffman, échantillonnage des images et diverses manipulations, enfin segmentation d'un signal de voix en intervalles par le choix d'une base de Malvar-Wilson d'entropie minimale. C'est dans ce dernier travail que les deux aspects (Fourier et entropie) sont utilisés conjointement.

- Le contrôle de travail et de connaissances consiste en la remise par les étudiants d'un devoir contenant les solutions des exercices ainsi que d'un rapport de travaux pratiques. Les deux copies sont notés sur dix.

- Le cours oral a ajouté quelques commentaires sur le filtrage et la restauration qu'il faudra ajouter au chapitre "le cas discret".

Le devoir
Le devoir consiste à rendre tous les exercices du cours, auquels vous pouvez ajouter tout commentaire de lecture, critique du cours, expérience ou correction. Ils seront très bienvenus. Les exercices dits de lecture où on vous demande de lire le texte de Shannon, de commenter et de comparer avec les preuves apportées sont optionnels et donc en bonus. Le texte de Shannon se trouve à l'adresse suivante sur le site des Bell labs, laboratoire où il fut écrit : http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf .

Contents

1 La modélisation probabiliste discrète (révision) 7
1.1 Probabilité conditionnelle 9
1.1.1 Indépendance 9
1.1.2 Modèles, exemples 9
1.2 Exercices 11

2 Distributions discrètes de probabilité (révision) 13
2.1 Espérance et variance 14
2.2 La convergence en probabilité 14
2.3 Exercices 15

3 Codes préfixes 17
3.1 Théorie des codages préfixes 18
3.1.1 Un premier exemple: le code de Shannon 22

4 Le codage de Huffman 27
4.1 Exercices et implémentations Matlab 30

5 Langage et communication selon Shannon 33
5.1 Introduction 33
5.2 Exercices d'implémentation en Matlab 35

6 Messages répétés et entropie 37
6.1 Messages typiques 37
6.2 Exercices et implémentations Matlab 41

7 La communication sûre est possible malgré le bruit 43
7.1 Transmission dans un canal bruité 43
7.2 Le théorème fondamental pour un canal discret bruité 46

8 Séries de Fourier (Révision) 49
8.1 Convolution des fonctions périodiques et séries de Fourier 52
8.1.1 Autres bases de Fourier 54
8.2 Bases de Fourier en dimension 2 55
8.3 Décroissance des coefficients de Fourier et problèmes de compression du signal 56
8.4 Phénomène de Gibbs 57

9 Le cas discret (Révision) 63
9.1 Transformée de Fourier Discrète, applications 63
9.1.1 La dimension 1 63
9.1.2 La dimension 2 66
9.1.3 Le phénomène du repliement de spectre ou aliasage 67
9.1.4 La transformée de Fourier rapide 71
9.1.5 L'utilisation de la transformée de Fourier discrète pour définir zoom, translations et rotations des images 74
9.1.6 Importances relatives de la phase et du module de la TFD pour une image 80
9.2 Lien avec la théorie de Shannon 80

10 La compression des images et la norme JPEG 85
10.1 Introduction 85
10.2 L'algorithme avec pertes 86
10.3 JPEG, codage sans pertes 91
10.4 Exercices et implémentation Matlab 91

11 Ondelettes de Malvar-Wilson et segmentation de la voix 93

Chapter 1

La modélisation probabiliste discrète (révision)

On part d'un ensemble d'évènements élémentaires ou atomiques Ω, par exemple les résultats d'un match de football, $\Omega = \{(0,0),(0,1),(1,0),\dots,\ (40,40)\}$. Plus généralement $\Omega = \mathbb{N}\times\mathbb{N}$. Un évènement en général est un sous-ensemble de Ω. Par exemple $A = \{(m,n) \mid m > n\}$ caractérise le fait que l'équipe 1 gagne, et $B = \{(m,n) \mid m = n\}$ caractérise le match nul. Si $\omega = (m,n)$ est æ en A, s'écrit que "ω est une réalisation de A."

Définition 1.1 *Algèbre d'ensembles "intéresants": C'est un ensemble $\mathcal{A}$ d'ensembles de Ω que satisfait les axiomes suivants:*

- $\emptyset, \Omega \in \mathcal{A}$
- $A \in \mathcal{A} \Rightarrow A^c \in \mathcal{A}$
- $A, B \in \mathcal{A} \Rightarrow A \cap B \in \mathcal{A}$ *(et donc aussi $A \cup B \in \mathcal{A}$).*

Définition 1.2 *Variable aléatoire discrète. C'est une application $X : \Omega \to E$ où E est un ensemble fini ou dénombrable et tel que les ensembles $\{\omega \in \Omega \mid X(\omega) = e\}$ soient tous dans $\mathcal{A}$. Dans le cas où $E \subset \mathbb{R}$, on parle de variable aléatoire real.*

Dans l'exemple du football, $M(\omega) = M((m,n)) := m$ est une variable aléatoire réelle qui peut s'appeler "nombre de buts de l'équipe 1". Regardons un exemple de Ω plus général qui est aussi un grand classique de théorie des probabilités, le jeu de pile ou face. On code face par 0 et pile par 1. Une suite infinie de tirages appartient à l'ensemble d'évènements $\Omega := \{0,1\}^{\mathbb{N}}$. Chaque élément de Ω s'écrit $\omega = (\omega(1),\dots,\omega(n),\dots)$ et l'application $X_n : \omega \to \omega(n)$ est un variable aléatoire qu s'interprète comme le "résultat du n-ième tirage". Aussi nous pouvons considérer l'ensemble des N premiers tirages, $\Omega_N := \{0,1\}^N$. D'une certaine manière Ω_N est contenu dans Ω mais pour le formaliser il faut associer à chaque élément $\omega_N = (\omega(1),\dots,\omega(N)) \in \Omega_N$ l'ensemble de toutes les suites qui commencent par ω_N, que nous appellerons $\Omega(\omega_N)$. Nous pouvons considèrer l'algèbre engendrée par les $\Omega(\omega_N)$ quand ω_N varie dans Ω_N et N varie dans $\mathbb{N}$. Il s'agit de l'algèbre la plus petite contenant tous ces évènements. Cette algèbre s'appelle "algèbre d'évènements en temps fini".

Exercice 1 Démontrer que tout élément de l'algèbre $\mathcal{A}$ des évènements en temps fini est une union finie d'évènements de type $\Omega(\omega_N)$. Pour le prouver, il suffit d'appeler $\tilde{\mathcal{A}}$ cet ensemble

d'unions finies et de prouver qu'elle a la structure d'une algèbre. S'il en est ainsi, elle est forcément l'algèbre la plus petite contenant les $\Omega(\omega_N)$.

En fait l'algèbre des évènements en temps fini ne nous dit rien sur ce qui se passe à l'infini. Par exemple l'ensemble : $A := \{\omega \in \Omega \mid \lim_n X_n(\omega) = 0\}$ n'est pas dans $\mathcal{A}$. Pour le voir, il suffit de remarquer que s'il était dans $\mathcal{A}$, on aurait $A = \cup_{n\in I}\Omega(\omega_n)$ où I est un sous-ensemble fini de $\mathbb{N}$ et $\omega_n \in \Omega_N$. Appelons k l'indice le plus grand qui apparaît dans I. Alors on vérifie immédiatement que si $\omega \in A$ et si on considère un autre élément $\omega' \in \Omega$ tel que $\omega'(i) = \omega(i)$ pour $i \leq k$, alors ω' est en A. Donc nous pouvons imposer que $\omega'(n) = 1$ pour $n \geq k$, et celà implique que ω' n'est pas dans A, une contradiction. C'est pourquoi Kolmogorov a étendu la notion d'algèbre à celle de $\sigma-$algèbre, ce qui permet de considérer un évènement comme A.

Définition 1.3 *Une σ-algèbre $\mathcal{F}$ de Ω est une algèbre telle que si A_n est dans $\mathcal{F}$, alors $\cup_n A_n$ est aussi dans $\mathcal{F}$. Etant donné un ensemble $\mathcal{A}$ de parties de Ω, on appelle σ-algèbre engendrée par $\mathcal{A}$ et on écrit $\sigma(\mathcal{A})$ l'intersection de toutes las σ-algèbres contenant $\mathcal{A}$.*

Une telle intersection existe parce qu'il y a au moins une σ-algèbre qui contienne $\mathcal{A}$: l'ensemble $\mathcal{P}(\Omega)$ de toutes les parties de Ω est en effet une σ-algèbre.

Exercice 2 Démontrer que l'ensemble $A := \{\omega \in \Omega \mid \lim_n X_n(\omega) = 0\}$ est dans $\sigma(\mathcal{A})$, où $\mathcal{A}$ désigne l'algèbre d'évènements en temps fini. Indication: prouver que

$$A = \bigcup_{n\geq 1} \bigcap_{m\geq n} \{\omega \mid X_m(\omega) = 0.\}$$

En pratique on commence par connaître la valeur de la probabilité de quelques évènements. Donc, on déduit la probabilité des évènements de l'algèbre $\mathcal{A}$ engendrée par ces évènements, et finalement on déduit la probabilité des évènements de $\sigma(\mathcal{A})$. Les règles pour déduire ces probabilités les unes des autres sont:

Définition 1.4 *Soit Ω un espace de probabilité muni d'une une σ-algèbre $\mathcal{F}$. On dit que $\mathbb{P}$ est une probabilité sur $(\Omega, \mathcal{F})$ si pour tout A dans $\mathcal{F}$ et pour toute suite disjointe A_n dans $\mathcal{F}$,*

- $0 \leq \mathbb{P}(A) \leq 1$, $\mathbb{P}(\Omega) = 1$
- $\mathbb{P}(\bigcup_n A_n) = \sum_n \mathbb{P}(A_n)$.

La dernière propriété s'appelle "σ-additivité" de la probabilité.

Exercice 3 déduire les conséquences suivantes:

- si $A \subset B$, alors $\mathbb{P}(A) \leq \mathbb{P}(B)$.
- Si $A_n \in \mathcal{F}$, $\mathbb{P}(\cup_n A_n) \leq \sum_n \mathbb{P}(A_n)$.

1.1 Probabilité conditionnelle

En réalité très souvent les probabilités auxquelles on a accès sont des probabilités conditionnelles.

Définition 1.5 *Etant donné un espace de probabilité* $(\Omega, \mathcal{F}, \mathbb{P})$ *et* $A, B \in \mathcal{F}$*, on appelle probabilité conditionnelle de* A *sachant* B

$$P(A \mid B) := \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} \text{ si } \mathbb{P}(B) \neq 0, = 0 \text{ si } \mathbb{P}(B) = 0.$$

Exercice 4 Démontrer que pour tout B dans $\mathcal{F}$, l'application $A \to \mathbb{P}(A \mid B)$ est une probabilité sur $(\Omega, \mathcal{F})$. Démontrer également la "règle des causes totales" : si les B_i, $i \in \mathbf{N}$ sont des évènements formant une partition de Ω, alors

$$\mathbb{P}(A) = \sum_{i=1}^{\infty} \mathbb{P}(A \mid B_i)\mathbb{P}(B_i).$$

1.1.1 Indépendance

Définition 1.6 *Soit* $(\Omega, \mathcal{F}, \mathbb{P})$ *un espace de probabilité.*

A et B sont indépendantes si $\mathbb{P}(A \mid B) = \mathbb{P}(A)$*, ce qui est équivalent à* $\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$.

- *Une famille* $(A_i)_{i \in I}$ *est une famille d'évènements indépendants si pour toute sous-famille finie* $A_{i_1}, \ldots, A_{i_n}$, $\mathbb{P}(A_{i_1} \cap \cdots \cap A_{i_n}) = \mathbb{P}(A_{i_1}) \ldots \mathbb{P}(A_{i_n})$.
- *Une suite* $(X_n)_{n \geq 1}$ *de variables aléatoires discrètes* $X_n : \Omega \to E$ *est indépendante si pour tout* $(e_1, \ldots, e_n, \ldots) \in E^{\mathbf{N}}$*, les évènements* $(X_n = e_n)$ *sont indépendants.*
- *Une suite* $(X_n)_{n \geq 1}$ *de variables aléatoires réelles* $X_n : \Omega \to \mathbb{R}$ *est indépendante si pour toute suite d'intervalles de* $\mathbb{R}$*,* $(I_1, \ldots, I_n, \ldots)$ *les évènements* $(X_n \in I_n)$ *sont indépendants.*

Exercice 5 Deux exemples importants:

- Soit $\Omega = [0,1]^N$, $\mathbb{P}(A) = \text{volume}(A)$ par $A \subset \Omega$. Démontrer que les coordonnées $X_i : \omega = (\omega_1, \ldots, \omega_n) \in \Omega \to \omega_i$ sont des variables aléatoires indépendantes.
- Si par contre $\Omega = B(0,1)$ est la boule de centre 0 et de rayon 1 et $\mathbb{P}$ est la mesure de Lebesgue multipliée par un facteur λ tel que $\mathbb{P}(B(0,1)) = 1$, montrer que les variables X_i ne sont pas indépendantes.

1.1.2 Modèles, exemples

Problème des trois prisonniers

Trois prisonniers A, B et C sont enfermés sans communication dans la prison d'un régime totalitaire. Ils savent que deux d'entre eux sont condamnés à mort mais ils ignorent lesquels. Bien sûr, le geôlier n'est pas autorisé à le leur faire savoir. Un des prisonniers, A, propose

alors au geôlier le raisonnement suivant : "Je sais déjà que l'un de B ou de C est condamné. Tu ne me donneras donc aucune information utile sur mon sort si tu me communiques que l'un de B ou de C est condamné. S'il te plaît, donne-moi le nom d'un des deux, B ou C, qui est condamné." Le geôlier réfléchit un moment et répond : "-tu as raison. Je t'annonce que B est condamné. Mais ne va pas croire qu'avec cette information tu va pouvoir tirer quoi que ce soit d'utile sur ton propre sort" Le prisonnier répond "- tout le contraire ; avant j'avais une probabilité de 2/3 d'être condamné ; maintenant tout se joue entre C et moi, n'est-ce pas? Donc ma probabilité de mourir est devenue 1/2 et non plus deux tiers. Je peux dormir un peu plus tranquille!!". Le geôlier hausse les épaules et s'en va.

Qui a raison, du prisonnier A ou du geôlier? Le prisonnier A a-t-il raison de penser qu'il a gagné un peu de tranquillité, ou est-il victime d'une illusion?

Solution

Pour formaliser cet type de problème, il faut chercher les évènements atomiques, c'est-à-dire énumérer chaque suite de probabilités et ensuite essayer de chercher sa probabilité. Une fois calculées les probabilités de ces évènements atomiques, toue autre probabilité devient la probabilité d'un évènement qui est une union d'évènements atomiques. Donc elle devient une somme de probabilités, facile à calculer. Selon cette description la manière de procéder est:

- énumérer et nommer tous les évènements distincts (en général des suites d'évènements);
- donner un nom aux variables aléatoires d'intérêt;
- exprimer les probabilités indiquées par le problème: très souvent nous verrons que ce sont des probabilités conditionnelles;
- prendre en compte toutes les indépendances implicites dans l'énoncé;
- formaliser les évènements dont on veut calculer les probabilités et les exprimer en fonction des évènements élémentaires;
- finalement calculer la probabilité cherchée.

Notre problème montre deux évènements aléatoires consécutifs: d'abord le choix des prisonniers condamnés: AB, BC ou AC. Ensuite le choix éventuel effectué par le geôlier de B ou de C dans le cas où il lui faut effectivement choisir, quand B et C sont tous deux condamnés. De plus, il faut prendre en compte un grand principe des probabilités que parfois on appelle *principe de probabilité subjective*, suivant lequel quand de plusieurs possibilités 1, 2, ... n on ignore tout, on attribue la même probabilité à chacune des possibilités, c'est-à-dire $1/n$. Dans notre cas la probabilité que AB, ou BC, ou AC soient condamnés doit donc être fixée à un tiers. De la même manière, si B et C sont condamnés le geôlier doit choisir entre eux pour donner un nom. Comme A ignore quel critère le geôlier adopte pour choisir entre B et C, il doit considérer que les probabilités que le geôlier nomme B ou C sont égales à 1/2. Les évènements atomiques sont ABC, ACC, BCB et BCC, où les deux premières lettres indiquent les condamnés, et la dernière est le choix du condamné nommé par le geôlier. Les variables aléatoires naturelles sont X, Y, Z, où XY est la liste ordonnée des condamnés et Z le condamné indiqué par le geôlier. Par exemple $XY(ABC) = AB$ et $Z(ABC) = C$. Maintenant, nous pouvons exprimer les probabilités subjectives qui sont nos seules données:

- $\mathbb{P}(XY = AB) = \mathbb{P}(XY = BC) = \mathbb{P}(XY = AC) = \frac{1}{3}$
- $\mathbb{P}(Z = C \mid XY = BC) = \mathbb{P}(Z = B \mid XY = BC) = \frac{1}{2}$.

Deux des probabilités auxquelles nous avons accès sont des probabilités conditionnelles. Celà est tout-à-fait naturel car cela correspond à des questions du type : "quelle est la probabilité de tel évènement si tel autre se produit?" Maintenant, nous pouvons traduire la question que se pose le prisonnier : "quelle est ma probabilité d'être sauvé sachant que le geôlier m'a indiqué que B est condamné?" Nous devons donc calculer:

$$p = \mathbb{P}(XY = BC \mid Z = B)$$

Par définition de la probabilité conditionnelle,

$$p = \frac{\mathbb{P}(XY = BC \ \& \ Z = B)}{\mathbb{P}(Z = B)}.$$

Mais en utilisant ce que nous savons et de nouveau la probabilité conditionnelle.

$$\mathbb{P}((XY = BC) \ \& \ (Z = B)) = \mathbb{P}(Z = B \mid XY = BC)\mathbb{P}(XY = BC) = \frac{1}{2}\frac{1}{3} = \frac{1}{6}.$$

Pour calculer $\mathbb{P}(Z = B)$ nous pouvons utiliser la *règle des causes totales*:

$$\begin{aligned} \mathbb{P}(Z{=}B) &= \mathbb{P}(Z{=}B|XY{=}AB)\mathbb{P}(XY{=}AB) + \mathbb{P}(Z{=}B|XY{=}AC)\mathbb{P}(XY{=}AC) \\ &\quad + \mathbb{P}(Z{=}B|XY{=}BC)\mathbb{P}(XY{=}BC) \\ &= \frac{1}{3} + 0 + \frac{1}{2}.\frac{1}{3} = \frac{1}{2}. \\ &= \end{aligned}$$

Finalement nous obtenons

$$p = \frac{\frac{1}{6}}{\frac{1}{2}} = \frac{1}{3}.$$

L'estimation de sa propre probabilité de survie par A n'a pas été changée par l'information donnée par le geôlier.

Exercice 6 La conclusion antérieure dépend strictement de l'hypothèse que le prisonnier ne connait rien sur le critére de sélection par le geôlier entre B et C. Supposons que le prisonnier devine que le geôlier préfère nommer B si B et C sont condamnés. Alors $q = \mathbb{P}(Z = B \mid XY = BC) > \frac{1}{2}$. Reprendre les calculs précédents et démontrer que le prisonnier gagne alors de l'information grâce à la réponse du geôlier.

1.2 Exercices

Exercice 7 Un modèle discret pour gagner à un jeu très simple et moins sinistre. Il y a trois cartes. La première a deux faces rouges et nous l'appellerons RR. La seconde a deux faces vertes (VV) et la troisième a une face rouge et une autre verte (RV). On tire une carte au hasard et on tire aussi au hasard le coté exposé de la carte. Les joueurs observent cette face exposée et font des paris sur la couleur de l'autre face.

Supposons par exemple que la face exposée soit rouge. Quelle est la probabilité que l'autre face soit rouge? Et verte? Supposons que chaque joueur parie pour une couleur. Si l'autre joueur place 100 euros sur la table en pariant pour rouge, combien devrais-je placer sur la table, en pariant sur vert, pour que mon espérance de gain soit positive?

Exercice 8 Soit $\mathcal{A}_n$ une suite croissante ($\mathcal{A}_n \subset \mathcal{A}_{n+1}$) d'algèbres de Ω. Démontrer que $\bigcup_n \mathcal{A}_n$ est une algèbre.

Exercice 9 Soit A_i, $i \in \mathbf{N}$ une partition de Ω. Décrire l'algèbre engendrée par les A_i et la σ-algèbre engendrée par les A_i.

Exercice 10 Démontrer que $\mathbb{P}(A \mid B\&C)\mathbb{P}(B \mid C) = \mathbb{P}(A\&B \mid C)$.

Exercice 11 Si $A_1, \dots A_n$ sont indépendants alors $\tilde{A}_i$ sont indépendants, où $\tilde{A}_i$ peut être arbitrairement A_i ou A_i^c.

Exercice 12 Soit un espace Ω à quatre elements $\{\omega_1, \omega_2, \omega_3, \omega_4)$. Soit $\mathbb{P}$ la probabilité définie par $\mathbb{P}(\omega_i) = \frac{1}{4}$, $i = 1, \dots, 4$. Considérons les évènements $A = \{\omega_1, \omega_2\}$, $B = \{\omega_2, \omega_3\}$, $C = \{\omega_1, \omega_3\}$. Vérifier que A et B sont indépendants, B et C aussi, C et A aussi, mais que A, B, C ne sont pas indépendants.

Exercice 13 Trois touristes tirent en même temps sur un éléphant. L'animal meurt, touché par deux balles. La valeur de chaque chasseur est mesurée par la probabilité qu'il atteigne sa cible. Ces probabilités sont 1/4, 1/2, 3/4. Calculer pour chacun des chasseurs sa probabilité d'avoir raté l'éléphant. (Cette probabilité dépend de l'évènement observé: si l'éléphant avait reçu trois balles, nous saurions par exemple que la probabilité d'avoir raté est zéro pour chaque chasseur).

Exercice 14 Le professeur Peplluis Serra Balaguer voyage de Toronto à Paris en passant par New York et Francfort. La probabilité que la valise se perde est identique dans chacun de ces aéroports et égale à p. Quand le professeur Balaguer arrive à Paris, sa valise a disparu. Quelles sont les probabilités que la valise soit restée à Toronto, Londres et Paris?

Chapter 2

Distributions discrètes de probabilité (révision)

Définition 2.1 *Soit $(\Omega, \mathcal{F}, \mathbb{P})$ un espace de probabilité, $\mathcal{X}$ un ensemble fini ou dénombrable et $X : \Omega \to \mathcal{X}$ tel que les ensembles $\{\omega \in \Omega \mid X(\omega) = x\}$ soient tous dans la tribu $\mathcal{F}$. Alors on dit que X est une variable aléatoire sur $(\Omega, \mathcal{F}, \mathbb{P})$.*

Par exemple une suite de tirages à pile ou face, qui se formalise comme une suite de variables de Bernoulli, X_1, ..., X_n avec $X_i = 0$ (pile) où $X_i = 1$ (face) et $\mathbb{P}(X_i = 1) = p$, $\mathbb{P}(X_i = 0) = 1 - p$. Si les tirages sont indépendants, nous avons

$$\mathbb{P}(X_1 = x_1, \ldots, X_n = x_n) = \mathbb{P}(X_1 = x_1) \ldots \mathbb{P}(X_n = x_n) \text{ pour tout } (x_1, \ldots, x_n) \in \{0,1\}^n.$$

Alors en écrivant $X = (X_1, \ldots, X_n)$ et $x = (x_1, \ldots, x_n)$,

$$\mathbb{P}(X = x) = p^{h(x)}(1-p)^{n-h(x)}, \text{ où } h(x) := \sum_{i=1}^{n} x_i.$$

La fonction $h(x)$ s'appelle "poids de Hamming" de x.

Exercice 15 Vérifier que $\sum_{x \in \{0,1\}^n} p(x) = \sum_{k=0}^{n} C_n^k p^k (1-p)^{n-k} = 1$.

Définition 2.2 *Si $\mathcal{X}$ est un ensemble dénombrable et $(p(x))$, $x \in \mathcal{X}$ une fonction sur $\mathcal{X}$ satisfaisant:*

1. $0 \leq p(x) \leq 1$
2. $\sum_{x \in \mathcal{X}} p(x) = 1$,

on dit que $(p(x))$, $x \in \mathcal{X}$, est une distribution de probabilité sur $\mathcal{X}$.
Exemple fondamental: Si X est une variable aléatoire définie sur $(\Omega, \mathcal{F}, \mathbb{P})$ et à valeurs dans $\mathcal{X}$, on considère pour tout $A \subset \mathcal{X}$,

$$\mathbb{P}_X(A) := \mathbb{P}(X \in A) = \sum_{x \in A} \mathbb{P}(X = x) = \sum_{x \in A} \mathbb{P}_X(x).$$

Posant $p(x) = \mathbb{P}_X(x)$, on dit que $(p(x))$, $x \in \mathcal{X}$ est la distribution (ou loi) de la variable X.

Donnons quelques exemples de distributions classiques.

Si $\mathcal{X} := \{0,1\}^n$, alors $p(x) := p^{h(x)}(1-p)^{n-h(x)}$ s'appelle distribution de Bernoulli d'ordre n et paramètre p. C'est la loi de la variable aléatoire $X := (X_1, \ldots, X_n)$ où X_i sont des variables aléatoires de Bernoulli de paramètre p et ordre 1. Donc on peut considérer la distribution sur $\{0, 1, \ldots, n)$ de la variable aléatoire $S_n := \sum_{i=1}^n X_i$. C'est facile de vérifier que $\mathbb{P}(S_n = k) = C_n^k p^k (1-p)^{n-k} = p_k$. Cette distribution sur $\{0, 1, \ldots, n\}$ s'appelle distribution binomiale.

2.1 Espérance et variance

Soit X une variable aléatoire discrète avec valeurs dans $\mathcal{X}$, de distribution $p(x)$, $x \in \mathcal{X}$. Soit $f : \mathcal{X} \to \mathbb{R} \cup \{+\infty, -\infty\}$ une application.

Définition 2.3
- *Si $f \geq 0$, on définit $\mathbb{E}[f(X)] := \sum_{x\in\mathcal{X}} \mathbb{P}(X = x) f(x)$. (Cette quantité peut être infinie).*
- *Si $\mathbb{E}[|f(X)|] < +\infty$, se define $\mathbb{E}[f(X)] := \sum_{x\in\mathcal{X}} \mathbb{P}(X = x) f(x)$ et on dit que $f(X)$ est intégrable.*
- *$\mathbb{E}$ est linéaire et monotone ($\forall x\ f(x) \leq g(x) \Rightarrow \mathbb{E}f(X) \leq \mathbb{E}g(X)$.)*

Dans le cas où X est elle-même une variable à valeurs réelles, on définit:

- moyenne de X : $m(X) = m_X := \mathbb{E}X$;
- Si X^2 est intégrable, variance de X: $\sigma^2(X) = \sigma_X^2 := \mathbb{E}(X^2) - (\mathbb{E}X)^2$.

$\sigma(X) = \sigma_X$ s'appelle la déviation typique de X.

Exercice 16 Vérifier que $\sigma_X^2 = \mathbb{E}[(X - m_X)^2]$. Vérifier que si X est Bernoulli d'ordre 1 et paramètre p, $m_X = p$ et $\sigma_X^2 = p(1-p)$. Dans le cas de la binomiale de paramètre p et ordre n, vérifier que $m_X = np$ et $\sigma_X^2 = np(1-p)$.

2.2 La convergence en probabilité

Commençons par quelques inégalités. L'inégalité de Markov dit que si X est une variable aléatoire et f une fonction mesurable telle que $f(X)$ soit intégrable,

$$\mathbb{P}(|f(X)| \geq a) \leq \frac{\mathbb{E}|f(X)|}{a}.$$

On en tire la fameuse inégalité de Tchebychev, qui dit que pour une variable aléatoire réelle et pour tout $\varepsilon > 0$,

$$\mathbb{P}(|X - \mathbb{E}X| \geq \varepsilon) \leq \frac{\sigma^2(X)}{\varepsilon^2}.$$

La démonstration consiste à appliquer l'inégalité de Markov à $f(X) = |X - \mathbb{E}X|^2$.

Exercice 17 Démontrer ces deux inégalités!

Nous allons appliquer l'inégalité de Tchebychev pour prouver une loi fondamentale, la loi faible des grands nombres.

Théorème 2.1 *Soit une suite de variables de Bernoulli indépendantes X_i de paramètre p, $i = 1, \ldots, n, \ldots$ et $S_n := \sum_{i=1}^n X_i$. Alors $\mathbb{P}(|\frac{S_n}{n} - p| \geq \varepsilon) \to 0$ quand $n \to \infty$.*

En effet l'espérance (ou moyenne) de S_n est np et la variance de S_n est $np(1-p)$. Donc la moyenne de $\frac{S_n}{n}$ est p et sa variance est $\sigma^2 = \frac{p(1-p)}{n}$. Appliquons l'inégalité de Tchebytchev pour obtenir

$$\mathbb{P}(|\frac{S_n}{n} - p| \geq \varepsilon) \leq \frac{p(1-p)}{n\varepsilon} \to 0 \text{ quand } n \to \infty.$$

Définition 2.4 *Soit Y_n des variables aléatoires réelles définies sur le même espace de probabilité. On dit que $Y_n \to Y$ en probabilité si $\lim_{n\to\infty} \mathbb{P}(|Y_n - Y| > \varepsilon) = 0$ for all $\varepsilon > 0$.*

La loi faible des grands nombres s'étend facilement à une situation légèrement plus générale. Considérons une suite de variables aléatoires X_n indépendantes, équidistribuées et de variance bornée $\sigma^2(X)$ et d'espérance $\mathbb{E}X$. Alors avec le même raisonnement que précédemment (additivité des variances et des espérances et inégalité de Tchebychev), on obtient le même résultat pour $S_n = \sum_{k=1}^n X_k$.

2.3 Exercices

Exercice 18 Un lot de montres identiques pour touristes arrive chez un marchand de Barbès. Ce lot peut provenir de deux usines: l'une à Singapour et l'autre à Hong Kong. Le marchand sait qu'en moyenne l'usine de Singapour produit un pourcentage de montres défectueuses de 1/200 tandis que l'usine de Hong Kong a un pourcentage de 1/1000. Le marchand teste une première montre et vérifie qu'elle marche. Quelle est la probabilité que la seconde montre qu'il va tirer fonctionne?

Exercice 19 On tire au hasard deux points dans $[0, 1]$, indépendamment. Le plus petit des nombres obtenus est plus grand que 1/3. Quelle est la probabilité que l'autre soit supérieur à 3/4?

Exercice 20 Pour $\lambda > 0$ on définit la loi de Poisson sur $\mathbf{N}$ par

$$p(k) = e^{-\lambda}\frac{\lambda^k}{k!},\ k = 0, 1, \ldots, .$$

Vérifier que $\sum_k p(k) = 1$ et calculer la moyenne et la variance de cette distribution.

Exercice 21 Calculer la moyenne et la variance d'une loi binomiale d'ordre n et de paramètre p.

Exercice 22 Soient $X_1, \ldots, X_n$ n des variables aléatoires discrètes indépendantes, identiquement distribuées de variance commune σ^2. Calculer la variance et l'écart type de $\frac{X_1+\cdots+X_n}{n}$.

Exercice 23 On dit qu'une variable aléatoire X à valeurs dans $\mathbf{N}$ suit une loi géométrique de paramètre $p \in [0, 1]$ si $\mathbb{P}(T = n) = p(1-p)^{n-1}$, $n \geq 1$. Calculer la moyenne et la variance de T. Démontrer que T "n'a pas de mémoire", c'est-à-dire que $\mathbb{P}(T \geq n_0 + n \mid T > n_0) = \mathbb{P}(T \geq n)$, $n \geq 1$.

Exercice 24 Soit X une variable aléatoire avec valeurs en $\mathbf{N}$. Démontrer que $\mathbb{E}X = \sum_{n=1}^{\infty} \mathbb{P}(X \geq n)$.

Chapter 3

Codes préfixes

Considérons une distribution discrète de probabilité, $(p_1, \ldots, p_k)$. On définit l'entropie de cette distribution par

$$H(p_1, \ldots, p_k) := -\sum_{i=1}^{k} p_i \log p_i.$$

En général le logarithme est en base 2, c'est-à-dire $\log 2 = \log_2 2 = 1$. L'interprétation de cette formule par Shannon est la suivante: en théorie de la communication le récepteur ignore ce que l'émetteur va lui écrire, mais il a néanmoins une idée claire de la probabilité de chaque message possible. Par exemple $p = (p_1, \ldots, p_k)$ peut être la distribution de probabilité des mots d'un dictionnaire $\mathcal{X} := \{x_1, \ldots, x_k\}$. Le degré de communication est supérieur quand l'incertitude sur le message croît. Pour Shannon, $H(p)$ mesure cette incertitude. Par exemple si $p_1 = 1$ et si les autres probabilités sont nulles, on vérifie que $H(p) = 0$, ce qui signifie qu'il n'y a pas d'incertitude. Si tous les p_i sont égaux à $\frac{1}{k}$, l'entropie est $\log_2 k$ et doit être maximale. Nous verrons que cette intuition est juste. L'incertitude ou entropie (Shannon utilise ces deux mots comme deux équivalents) se mesure en *bits*, une abbréviation de BInary digiT, terme créé par John W. Tukey. (Tukey est aussi l'inventeur avec James Cooley de la Fast Fourier Transform et c'est lui qui inventa le néologisme *software*). Le bit, unité fondamentale de l'informatique se définit comme la quantité d'information reçue par un récepteur qui attend un message 0 ou 1 et qui attribue la même probabilité $\frac{1}{2}$ aux deux possibilités. Effectivement si $p = (\frac{1}{2}, \frac{1}{2})$, on vérifie que $H(p) = 1$.

Définition 3.1 *Etant donné un ensemble de messages $\mathcal{X} = \{x_1, \ldots, x_k\}$ nous appellerons codage de ces messages une application $h : \mathcal{X} \to \{0,1\}^{(\mathbb{N})}$, ensemble des suites finies de zéros et de uns. Nous écrirons $l_i := l(h(x_i))$, la longueur du code de x_i.*

Le code le plus élémentaire que l'on puisse faire est d'énumérer les messages de $i = 0$ à k, convertissant i en un nombre binaire. Alors $h(x_1) = 0$, $h(x_2) = 1$, $h(x_3) = 10$, $h(x_4) = 11$, etc. La longueur maximale l_k des codes vérifie

$$[\log k] \leq l_k := l(h(x_k)) \leq [\log k] + 1.$$

Shannon démontre de plusieurs manières que, étant donnée une source émettrice d'entropie p, la longueur minimale moyenne des messages codés en bits est exactement $H(p)$. En d'autres termes il est possible de transmettre ce qu'émet la source en codant ses messages avec des nombres faits de zéros et de uns de longueur moyenne $H(p)$. Pour comprendre mieux, nous

devrons spécifier ce que nous entendons par *codage*. Nous allons commencer avec une théorie légèrement plus restrictive, la théorie des codages dits *préfixes*.

3.1 Théorie des codages préfixes

Un problème technique surgit. Si nous émettons des messages répétés, nous ignorerons où termine l'un et où commence le suivant: nous observons une suite de zéros et de uns et ne savons comment la couper. Si de toute suite observée $h(x_{i_1}) \dots h(x_{i_n})$ on peut déduire de manière unique les $x_{i_1}, \dots, x_{i_n}$, nous dirons que le codage est *uniquement déchiffrable*. Une manière très simple d'obtenir des codages déchiffrables est de leur imposer la propriété de préfixe.

Définition 3.2 *Un codage est appelé préfixe si pour tout $i, 1 \leq i \leq k$, chaque code $h(x_i)$ n'est le préfixe d'aucun autre code $h(x_j)$.*

Exercice 25 Démontrer qu'un codage préfixe est uniquement déchiffrable.

La théorie des codages préfixes est merveilleusement simple. Nous allons caractériser tous les codages préfixes et donner quelques exemples optimaux.

Théorème 3.1 (Inégalité de Kraft). *Si h est un codage préfixe, notons $l_1, \dots, l_i, \dots, l_k$ les longueurs des codes pour chaque symbole x_i. Si h est préfixe, alors*

$$\sum_{i=1}^{k} 2^{-l_i} \leq 1. \tag{3.1}$$

Réciproquement, soient (l_i), $1 \leq i \leq k$ entiers positifs tels que (3.1) soit vérifiée. Alors il existe un codage h avec la propriété du préfixe, dont les codes ont pour longueurs (l_i), $1 \leq i \leq k$.

Démonstration La démonstration se fait de manière très intuitive en dessinant un arbre binaire complet de profondeur n dont les feuilles sont les nombres binaires de n chiffres, de $00\dots00$ à $11\dots11$ (Voir la figure 3.1.) La racine de l'arbre est le mot vide. Ses fils sont 0 et 1. Le premier, 0, a pour fils 00 et 01 et le second, 1, a 10 et 11, etc. Les noeux de l'arbre donnent tous les codes possibles de longueur inférieure ou égale à n. Chaque noeud de l'arbre est la racine d'un sous-arbre. Dans ce sous-arbre il y a tous les codes qui ont sa racine comme préfixe. De cette remarque on déduit qu'*un codage préfixe h est tel qu'aucun code n'appartienne au sous-arbre d'un autre code.* En d'autres termes, les sous-arbres des $h(x_i)$ sont disjoints. Il est aussi facile de voir que comme $h(x_i)$ a une longueur l_i, son sous-arbre a une profondeur $n - l_i$ et le nombre de feuilles du sous-arbre de $h(x_i)$ est 2^{n-l_i}. Comme ces ensembles de feuilles sont tous disjoints, et comme le nombre total de feuilles est 2^n on arrive à l'inégalité

$$\sum_i 2^{n-l_i} \leq 2^n$$

qui implique l'inégalité de Kraft.

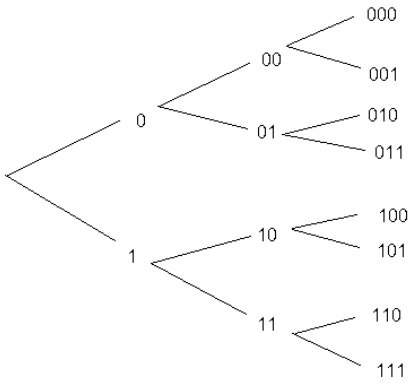


Figure 3.1: Arbre binaire complet des chiffres de moins de trois symboles. Chaque noeud de l'arbre représente un code. Si un codage est préfixe, il correspond à une sélection de noeuds tels qu'aucun noeud ne soit la racine d'un sous-arbre dont un autre noeud serait racine. Exemple: 00, 011, 10, 110 est un codage préfixe.

Maintenant voyons la réciproque. Si les nombres $(l_i)_{1\leq i\leq k}$ vérifient l'inégalité de Kraft, pourrons nous définir un codage préfixe h tel que $l_i = h(x_i)$? La construction est semblable au raisonnement précédent (voir la figure 3.2). On considère de nouveau l'arbre complet binaire de profondeur $n = \max_i l_i$. On ordonne les l_i par ordre croissant, $l_1 \leq \cdots \leq l_i \leq \ldots l_k$ et on considère les 2^{n-l_1} premières feuilles de l'arbre : ce sont les feuilles d'un sous-arbre dont la racine a pour longueur l_1: on décide que cette racine soit le code de x_1. Ensuite voyons les 2^{n-l_2} feuilles suivantes. De nouveau ce sont les feuilles d'un sous-arbre de racine de longueur l_2 et cette racine devient le code de x_2. Nous pouvons itérer tant que la quantité de feuilles utilisées n'excède pas 2^n, mais celà est exactement ce que nous garantit l'inégalité de Kraft! La figure 3.2 traite l'exemple suivant :

$$l_1 = 2,\ l_2 = l_3 = 3,\ l_4 = 5.$$

Dans cet exemple $n = 5$ et nous avons $2^{5-2} = 8$, $2^{5-3} = 4$, $2^{5-5} = 1$, ce que nous donne la taille de chaque sous-arbre. Los codes obtenus pour x_1, x_2, x_3, x_4 sont $00, 010, 011$ et 10000. ∘

Exercice 26 Démontrer de manière plus formelle que le codage obtenu dans la seconde partie de la démonstration du théorème 3.1 est préfixe. Pour comprendre mieux, tenter la même construction avec le même exemple, mais cette fois en ordonnant les sous-arbres avec des tailles qui croissent de haut en bas. Que se passe-t-il? Pourquoi ça ne marche pas?

Exercice 27 Important! vérifier que la construction du code de la seconde partie du théorème 3.1 se résume par l'algorithme suivant:

Algorithme calculant un codage préfixe $h(x_i)$
pour une suite $l_1 \leq \cdots \leq l_i \leq \cdots \leq l_k$ de longueurs fixées et vérifiant l'inégalité de Kraft.

On écrit tous les nombres binaires entre 0 et $2^m - 1$ avec exactement n chiffres (0 ou 1) en ajoutant à gauche les zéros nécessaires. Alors

1. $n = \max_i l_i$;
2. $h(x_1) =$ les l_1 premiers bits de 0 (soit $0 \dots 0$ l_1 fois);
3. $h(x_i) =$ les l_i premiers bits de $2^{n-l_1} + \cdots + 2^{n-l_{i-1}}$, $i \geq 2$.

Le théorème précédent nous donne une condition nécessaire sur les longueurs des codes pour qu'un codage préfixe permette de coder k messages. Maintenant notre problème, c'est que les longueurs l_i des codes soient les plus petites possible. Pour celà considérons de nouveau un ensemble $\mathcal{X} = (x_1, \dots, x_k)$ de k messages avec une distribution $p = (p_1, \dots, p_k)$. Si h est un codage des k messages avec $l_i = h(x_i)$, nous voulons minimiser la longueur moyenne, c'est-à-dire l'espérance de la longueur des codes. Cette longueur moyenne est, exprimée comme une espérance,

$$L(h) := \sum_{i=1}^{k} p_i l(h(x_i)) = \sum_i p_i l_i = \mathbb{E}(l(h(X))).$$

Théorème 3.2 *Appelons $L_{\inf} := \inf_h L(h)$, la longueur moyenne minimale que l'on peut obtenir avec un codage préfixe. Alors*

$$H(p) \leq L_{\inf} \leq H(p) + 1.$$

Lemme 3.1 *Soit $p = (p_1, \dots, p_k)$ une distribution de probabilité. La solution unique du problème où les inconnues sont les q_i,*

$$\begin{cases} -\sum_{i=1}^{k} p_i \log q_i = \min! \\ \sum_{i=1}^{k} q_i \leq 1,\ q_i \geq 0. \end{cases} \tag{3.2}$$

est $q_i = p_i$.

Démonstration Si $p = (p_i)$ est une distribution de probabilité et $q_i \geq 0$ satisfait $\sum_i q_i \leq 1$, en utilisant la concavité du logarithme,

$$\sum_{i=1}^{k} p_i \log \frac{q_i}{p_i} \leq \log \sum_{i=1}^{k} q_i \leq \log 1 \leq 0,$$

ce qui nous donne

$$-\sum_{i=1}^{k} p_i \log p_i \leq -\sum_{i=1}^{k} p_i \log q_i. \tag{3.3}$$

Donc $q := p$ réalise le minimum dans le problème (3.1). L'inégalité est stricte à moins que $\sum_i q_i = 1$ et $p_i = q_i$ par tout i. ∘

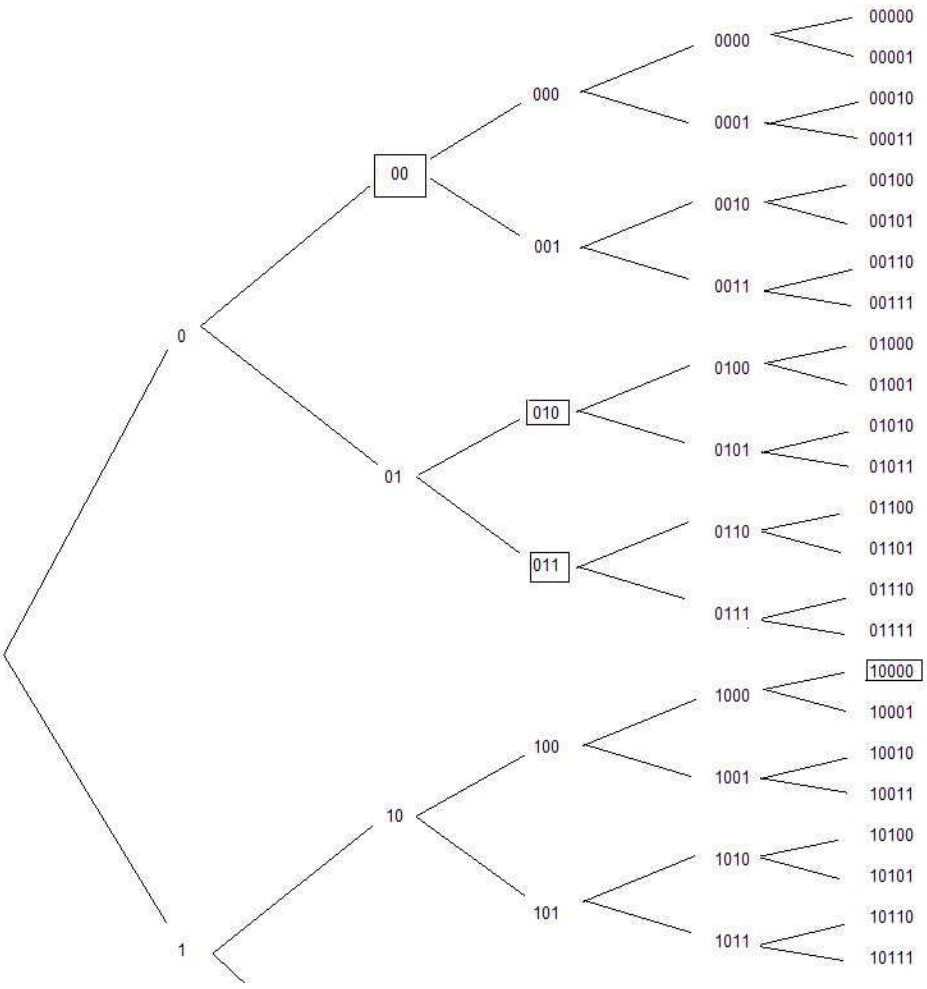


Figure 3.2: Dans un arbre binaire de profondeur 5 on construit un codage préfixe pour la suite de longueurs $l_1 = 2$, $l_2 = l_3 = 3$, $l_4 = 5$. C'est facile de voir que cette série vérifie l'inégalité de Kraft. Dans la figure on voit de haut en bas les codes obtenus, correspondant à des sous-arbres disjoints et de taille décroissante. Chacun a un nombre de feuilles 2^{5-l_j} et la racine du sous-arbre devient le code de x_j.

Preuve du théorème 3.2. Le problème que nous voulons résoudre est de minimiser la longueur moyenne d'un code sous la condition donnée par l'inégalité de Kraft, c'est-à-dire chercher $(l_i)_{i=1,\dots,k}$ telles que

$$\begin{cases} -\sum_{i=1}^k p_i l_i = \min! \\ \sum_{i=1}^k 2^{-l_i} \leq 1. \end{cases} \tag{3.4}$$

Résolvons d'abord le problème sans nous préoccuper du fait que les l_i doivent être entiers. On cherche donc une solution telle que l'on ait seulement $l_i \geq 0$. Alors en posant $y_i := 2^{-l_i}$ le problème (3.4) est équivalent au problème (3.2). Donc sa solution est $l_i^* = -\log p_i$ et nous obtenons que le minimum $\sum_{i=1}^k p_i l_k^* = H(p)$ est l'entropie. Toutefois les l_i^* ne sont généralement pas entiers et le mieux que puissions faire est de fixer $l_i := \lceil l_i^* \rceil$, l'entier le plus petit supérieur à l_i^*. Comme $l_i \geq l_i^*$ la condition de Kraft est vérifiée

$$\sum_{i=1}^k 2^{-l_i} \leq \sum_{i=1}^k 2^{-l_i^*} \leq 1.$$

Donc il existe un codage préfixe dont les codes ont pour longueur l_i et en plus

$$H(p) = \sum_{i=1}^k p_i l_i^* \leq \sum_{i=1}^k p_i l_i = \mathbb{E}l(h(X)) \leq \sum_{i=1}^k p_i (l_i^* + 1) = H(p) + 1.$$

∘

3.1.1 Un premier exemple: le code de Shannon

Soit X une source avec des symboles $\mathcal{X} = \{x_1, \dots, x_n\}$ et une distribution de probabilité $p = (p_1, \dots, p_n)$. Selon le théorème 3.2 nous pouvons construire des codes préfixes presque optimaux pour une source $p = (p_1, \dots, p_i, \dots, p_k)$ en prenant $l_i := \lceil \log p_i \rceil$ comme longueur du code de x_i.

Algorithme de codage de Shannon

1. Ordonner les p_i de façon décroissante: $p_1 \geq p_2 \geq \cdots \geq p_n$;
2. soit $P_i := \sum_{k=1}^{i-1} p_i$ (en particulier $P_1 = 0$);
3. $h(x_i)$ est composé des $l_i = \lceil -\log p_i \rceil$ premiers chiffres du développement binaire de P_i. En d'autres termes, on écrit $P_i = 0, a_1 a_2 \dots a_{l_i} \dots$ et on garde $a_1 \dots a_{l_i}$.

Nous allons vérifier directement que le code est préfixe et quasi optimal.

En effet, comme nous avons $-\log p_i \leq l_i \leq -\log p_i + 1$, on obtient

$$H(p) = -\sum_i p_i \log p_i \leq \sum_i p_i l_i \leq -\sum_i p_i (\log p_i + 1) = H(p) + 1.$$

Cela implique que la longueur moyenne (c'est-à-dire l'espérance de la longueur) des codes, $\mathbb{E}l$, vérifie

$$H \leq \mathbb{E}l \leq H + 1.$$

Cette relation signifie que le codage est quasi optimal. En effet, il vérifie la même inégalité qu'un codage optimal et on perd tout au plus un bit par symbole. Reste à démontrer que le codage ainsi défini a la propriété du préfixe. Observons que si $j \geq 1$, alors $P_{i+j} - P_i \geq p_i \geq 2^{-l_i}$. Si les l_i premiers bits de P_i coïncidaient avec ceux de P_{i+j}, celà impliquerait $P_{i+j} - P_i < 2^{-l_i}$. Donc ils ne coïncident pas et le codage est préfixe.

Exercices et implémentations Matlab

Exercice 28 Comparer le code de Shannon avec le code de l'algorithme 27.

Exercice 29 Expériences à faire. Implémenter en Matlab un algorithme qui:

1. étant donné un texte extrait les fréquences (probabilités empiriques) de tous les caractères (y compris les espaces et les chiffres). Ainsi on déduit une distribution empirique de ces caractères $p = (p_1, \ldots, p_k)$;
2. calcule l'entropie binaire de p, qui indique combien de bits il faut payer par caractère;
3. déduit quelle est la longueur théorique prévue pour le texte en bits (produit de l'entropie par le nombre de caractères);
4. calcule les codes de Shannon associés avec p;
5. génère le code binaire du texte, calcule sa longueur et la compare avec la longueur théorique prévue.

Exercice 30 D'un codage $x \in \mathcal{X} \to h(x) \in \{0,1\}^{(\mathbb{N})}$ on dit qu'il est uniquement déchiffrable si on a l'implication:

$$h(x_{i_1}) \ldots h(x_{i_n}) = h(x_{j_1}) \ldots h(x_{j_k}) \Rightarrow n = k \text{ et } x_{i_1} = x_{j_1}, \ldots, x_{i_n} = y_{j_k}.$$

Démontrer que si h est un codage avec la propriété de préfixe, alors le codage qui consiste à inverser l'ordre de chaque $h(x_i)$ est uniquement déchiffrable. En conclusion, il y a des codes uniquement déchiffrables qui n'ont pas la propriété du préfixe.

Exercice 31 Trouver un codage optimal pour la distribution

$$p := (0.01;\ 0.04;\ 0.05;\ 0.07;\ 0.09;\ 0.1;\ 0.14;\ 0.2;\ 0.3).$$

Exercice 32 Soient $p = (p_1, \ldots, p_n)$ et $q = (q_1, \ldots, q_l)$ deux distributions discrètes et $p \otimes q$ leur produit tensoriel défini par

$$p \otimes q = (p_1 q_1, \ldots, p_1 q_l,\ p_2 q_1, \ldots, p_2 q_l,\ \ldots, p_m q_1, \ldots, p_m q_l).$$

Quelle interprétation peut-on donner de cette distribution dans le langage des variables aléatoires? Vérifier que

$$H(p \otimes q) = H(p) + H(q).$$

Déduire que $H(p^{(n)}) = nH(p)$ où $p^{(n)}$ est le produit tensoriel de p par lui-même, n fois.

Exercice 33 Formulaire de Shannon (source : Shannon)

Commençons par rappeler un résultat que nous avons déjà utilisé:

Lemme 3.2 *Soient p et q deux distributions de probabilité discrètes. Alors*

$$\sum p(x) \log \frac{p(x)}{q(x)} \geq 0.$$

L'égalité se produit si et seulement si $p(x) = q(x)$ pour tout x.

Démonstration du Lemma 3.2. On utilise la concavité stricte du logarithme,

$$-\sum p(x) \log \frac{p(x)}{q(x)} = \sum p(x) \log \frac{q(x)}{p(x)} \leq \log(\sum p(x) \frac{q(x)}{p(x)}) = \log 1 = 0,$$

et cette inégalité est une égalité si et seulement si toutes les valeurs $\frac{p(x)}{q(x)}$ sont égales. Dans ce dernier cas leur valeur commune est évidemment 1. ∘

Soient X et Y deux variables aléatoires discrètes de distribution conjointe $p(x, y) = \mathbb{P}(X = x, Y = y)$. Donc on a $\mathbb{P}(X = x) = p(x) = \sum_{y \in \mathcal{X}} p(x, y)$ et $\mathbb{P}(Y = y) = p(y) = \sum_{x \in \mathcal{X}} p(x, y)$ L'entropie (ou incertitude) d'une variable discrète X à valeurs dans l'alphabet fini $\mathcal{X}$, et l'entropie d'une paire de variables aléatoires (X, Y) à valeurs dans $\mathcal{X} \times \mathcal{Y}$ ont été définies par

$$H(X) = -\sum_x p(x) \log p(x), \quad H(X, Y) = -\sum_{x,y} p(x, y) \log p(x, y).$$

1) Vérifier que l'entropie de X est l'espérance de $g(X)$, où $g(X) = \log \frac{1}{p(X)} = -\log p(x)$.

2) Démontrer que $H(X) \leq \log \mathrm{Card}(\mathcal{X})$, et que cette inégalité devient une égalité si et seulement si X a une distribution uniforme sur $\mathcal{X}$. (Utiliser le lemme 3.2 où p est la distribution de X et q la distribution uniforme sur $\mathcal{X}$).

3) Exemple important : On choisit pour X la variable de Bernouilli, $X = 1$ avec probabilité p, $X = 0$ avec probabilité $1 - p$. Alors

$$H(X) = -p \log p - (1 - p) \log(1 - p),$$

fonction de p que nous appellerons $H(p)$. Vérifier que $H(X) = 1$ bit quand $p = \frac{1}{2}$. Dessiner le graphe de $H(p)$ et démontrer que H vaut 0 en 0 et 1, et est maximal quand $p = \frac{1}{2}$. Interprétation : l'incertitude sur X est maximale quand $p = \frac{1}{2}$ et minimale quand X est déterministe. L'entropie est une mesure de l'incertitude sur la valeur de X.

4) L'entropie conjointe de deux variables aléatoires est mas petite que la somme des entropies. L'égalité se produit si et seulement si les deux variables aléatoires sont indépendantes.

$$H(X, Y) \leq H(X) + H(Y). \tag{3.5}$$

Il suffit d'appliquer le lemme 3.2 aux distributions $p(x, y)$ et $p(x)p(y)$.

5) **Entropie conditionnelle de Y sachant X :** c'est "la moyenne de l'entropie de Y pour chaque valeur de X, pondérée par la probabilité d'observer cette valeur particulière de X".

Traduire cette définition due à Shannon en une formule, et vérifier que la formule qui suit est une formule équivalente:

$$H(Y|X) = -\sum_{x,y} p(x,y) \log p(y|x).$$

"*This quantity measures how uncertain we are of Y on the average when we know X.*" Utilisons la définition de la probabilité conditionnelle :

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

$$H(Y|X) = -\sum_{x,y} p(x,y) \log p(x,y) + \sum_{x,y} p(x,y) \log \sum_y p(x,y) = H(X,Y) - H(X).$$

Donc

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

"*The uncertainty (or entropy) of the joint event X, Y is the uncertainty of X plus the uncertainty of Y when X is known*". Mais nous savons que

$$H(X) + H(Y) \geq H(X,Y) = H(X) + H(Y|X).$$

Alors

$$H(Y) \geq H(Y|X).$$

"*The uncertainty of Y is never increased by knowledge of X. It will be decreased unless X and Y are independent events, in which case it is not changed*".

Exercice 34 Entropie relative et information mutuelle

1) Considérons deux distributions de probabilité $p(x)$ et $q(x)$. Nous appellerons distance de Kullback Leibler, ou entropie relative des deux distributions $p(x)$ et $q(x)$ la quantité

$$D(p||q) := \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} = \mathbb{E}_p \log \frac{p(X)}{q(X)}.$$

Vérifier que $D(p||q) \geq 0$ et que $D(p||q) = 0 \Leftrightarrow p = q$. L'entropie relative est une mesure de la distance entre deux distributions. On appelle information mutuelle $I(X,Y)$ l'entropie relative entre la distribution conjointe $p(x,y)$ et la distribution produit $p(x)p(y)$,

$$I(X,Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{X}} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} = D(p(x,y)||p(x)p(y)) = \mathbb{E}_{p(x,y)} \log \frac{p(X,Y)}{p(X)p(Y)}.$$

2) Démontrer que

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

$$I(X,Y) = I(Y,X) = H(X) + H(Y) - H(X,Y), \quad I(X,X) = H(X).$$

On observera que $I(X,Y) = 0$ si X et Y sont indépendantes et que $I(X,Y) = 0$.

3) Soient X_1,, X_2, ..., X_n variables aléatoires discrètes de distribution conjointe $p(x_1,\ x_2, \dots x_n)$. Démontrer que

$$H(X_1,\ X_2, \dots X_n) = \sum_{i=1}^{n} H(X_i|X_{i-1}, \dots, X_1). \tag{3.6}$$

4) Démontrer, en utilisant les définitions de D et de la probabilité conditionnelle que

$$D(p(x,y)||q(x,y)) = D(p(x)||q(x)) + \sum_x p(x) D(p(y|x)||q(y|x)).$$

5) déduire finalement que

$$H(X_1,\ X_2, \dots X_n) \leq \sum_{i=1}^{n} H(X_i),$$

et que l'égalité se produit si et seulement si X_i sont indépendantes. (Utiliser la question précédente et (3.6)).

Chapter 4

Le codage de Huffman

En 1952 Huffman découvrit un codage particulièrement simple. Nous allons le décrire et démontrer su optimalité. L'algorithme s'explique de nouveau très bien avec une figure et un exemple. Soient

$$p = (p_1, \ldots, p_k) = (0.01; 0.02; 0.04; 0.13; 0.13; 0.14; 0.15; 0.15; 0.23),$$

ordonnées en croissant. A partir de cette suite on construit un arbre binaire dont les feuilles seront les probabilités. A chaque pas de la construction de l'arbre on groupe les deux probabilités qui ont la somme la plus petite et on les remplace par leur somme. On passe donc à la suite obtenue en réunissant $(p_1 + p_2, \ldots, p_k)$. Le noeud parent immédiat de p_1 et p_2 est alors $p_1 + p_2$. Itérant le procédé $n - 1$ fois, on construit un arbre comme celui de la figure 4.1. Avec la convention graphique que la probabilité la plus grande est toujours à droite et la probabilité la plus petite à gauche, l'arbre se réarrange comme indiqué dans la figure 4.2. Alors nous pouvons associer par la convention habituelle un code binaire à chaque noeud dans l'arbre, ce qui fixe en particulier un code pour chaque feuille de l'arbre, c'est-à-dire chaque p_i. Comme nous allons voir, ce procédé nous donne un codage optimal au sens adopté au chapitre précédent.

Exercice 35 Calculer la longueur moyenne du code qui a été construit!

Exercice 36 Etant donnée une distribution de probabilités discrète $p = (p_1, \ldots, p_k)$, démontrer qu'il existe bien au moins un codage optimal, à savoir un codage dont l'espérance de longueur est minimale parmi les espérances de longueur de tous les codages possibles.

Le lemme qui suit explique pourquoi un code de Huffman est optimal.

Lemme 4.1 *Soit $n \geq 3$ et considérons une distribution décroissante de probabilités $p = (p_1, p_2, \ldots, p_k)$, avec $p_1 \geq p_2 \geq \cdots \geq p_k > 0$. Alors il existe un code optimal h pour p tel que*

$$h(k) = w0, \quad h(k-1) = w1,$$

pour au moins une suite w faite de zéros et de uns, et tel que le code h' défini par

$$h'(i) = h(i), \quad 1 \leq i \leq k-1, \quad h'(k-1) = w$$

soit optimal pour la distribution $p' = (p_1, p_2, \ldots, p_{k-2}, p_{k-1} + p_k)$.

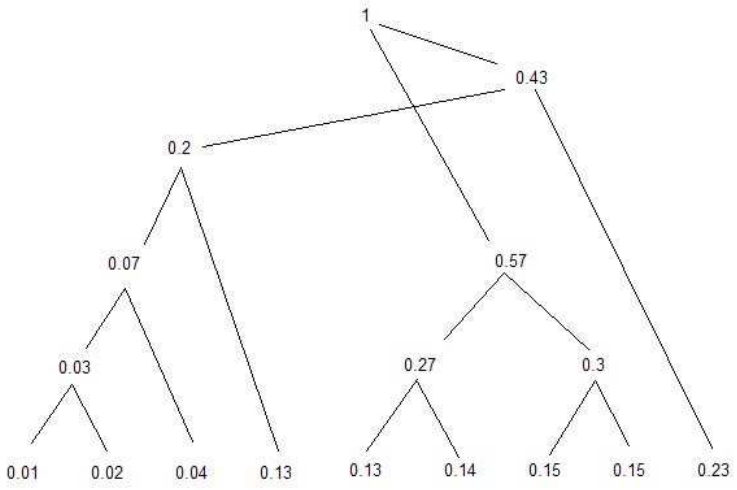


Figure 4.1: Soit $p = (p_1, \ldots, p_n) = (0.01; 0.02; 0.04; 0.13; 0.13; 0.14; 0.15; 0.15; 0.23)$ une distribution ordonnée. A partir de cette suite on construit un arbre binaire. A chaque pas de la construction de l'arbre on groupe les deux probabilités qui ont la somme la plus petite et leur somme est placée au noeud qui est créé comme parent immédiat de ces deux probabilités. En itérant le procédé $n-1$ fois, on construit un arbre binaire dont la racine est 1, la somme de toutes les probabilités, et dont les feuilles sont les probabilités de départ.

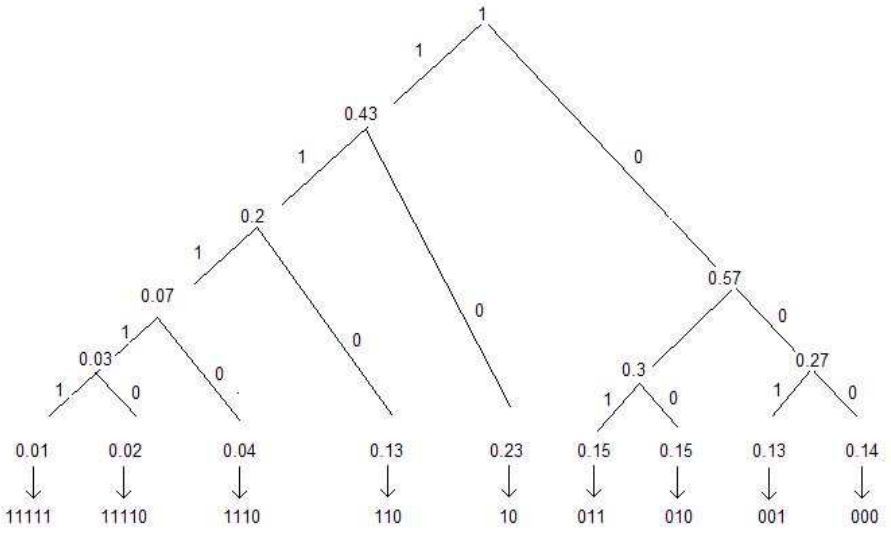


Figure 4.2: Avec la convention graphique que la probabilité la plus grande est toujours à droite et la probabilité la plus petite à gauche, l'arbre de la figure 4.1 se réarrange. Alors nous pouvons associer par la convention habituelle un code binaire à chaque noeud dans l'arbre, ce qui fixe en particulier un code pour chaque feuille de l'arbre, c'est-à-dire chaque p_i.

Il s'agit dans tous les cas de codage préfixe. Voyons d'abord pourquoi le lemme justifie la construction du code de Huffman. Le lemme nous garantit que pour construire un codage optimal d'une distribution de k éléments, il suffit de trouver un code optimal pour la distribution de $n-1$ éléments obtenue en groupant les deux probabilités les plus petites. Et que le code de ces probabilités p_{k-1} et p_k s'obtient en ajoutant un zéro et un 1 au code de $p_k + p_{k-1}$. Au dernier pas, quand n est réduit à 2, le codage optimal est forcé, 0 et 1. Donc on déduit que tous les codes successifs sont optimaux.

Preuve du lemme 4.1. Soit h un code optimal pour p. Nous pouvons imposer, comme une propriété générale des codes préfixes, que les longueurs vérifient $l_1 \leq l_2 \leq \cdots \leq l_k$. En effet si par exemple $l_1 > l_2$, il y a deux cas : si $p_1 = p_2$ nous pouvons échanger p_1 et p_2. Par contre $p_1 > p_2$ est impossible, puisque celà impliquerait que nous pouvons obtenir un code de longueur moyenne strictement inférieure en interchangeant les codes de p_1 et p_2. En effet cet échange donnerait $p_2 l_1 + p_1 l_2 < p_1 l_1 + p_2 l_2$.

Observons aussi comme une propriété générale d'un codage préfixe optimal que $l_{k-1} = l_k$. Sinon, nous pourrions maintenir la propriété du préfixe et faire diminuer sa longueur moyenne en supprimant les $l_k - l_{k-1}$ derniers bits de $h(k)$.

Le code $h(k-1)$ peut s'écrire $w0$ ou bien $w1$. Supposons par exemple que ce soit $w0$. Alors nous pouvons choisir $h(k) = w1$. En effet, si ce code est déjà utilisé par un p_i, on peut échanger le code $h(i)$ de p_i et celui de p_{k-1}. Si le code $w1$ n'est pas utilisé, il est clair que nous pouvons l'utiliser pour $h(k)$ tout en maintenant la propriété du préfixe. En effet si un autre $h(j)$ était préfixe de $w1$, comme il serait de longueur strictement inférieure puisqu'il est différent de $w1$, $h(j)$ serait préfixe de $w0 = h(k-1)$. Après ces échanges éventuels, la longueur moyenne n'a pas changé et reste optimale.

Considérons maintenant le code $\tilde{h}$ induit par h sur $p' := (p_1, \ldots, p_{k-2}, p_{k-1} + p_k)$:

$$\tilde{h}(i) = h(i), \quad (1 \leq i \leq k-2) \quad \text{ et } \tilde{h}(k-1) = w.$$

Pour conclure la preuve du lemme, il reste à démontrer que $\tilde{h}$ est de longueur moyenne optimale. La longueur moyenne de ce codage, $\tilde{L}$, est reliée à celle de h, L, par la relation

$$L = \tilde{L} + p_{k-1} + p_k.$$

Soit L' la longueur moyenne d'un code optimal h' sur p'. Partant de h' on peut définir un code $\hat{h}$ sur p par

$$\hat{h}(i) = h'(i), \quad (1 \leq i \leq k-2), \quad \hat{h}(k-1) = h'(k-1)0, \quad \hat{h}(k) = h'(k-1)1.$$

Donc la longueur moyenne de $\hat{h}$ est

$$\hat{L} = L' + p_{k-1} + p_k.$$

Mais nous savons que L' est la longueur optimale de p' et que L est la longueur optimale de p. Donc $\hat{L} \geq L'$, $\tilde{L} \geq L'$ et nous obtenons:

$$\tilde{L} + p_{k-1} + p_k = L \leq \hat{L} = L' + p_{k-1} + p_k \leq \tilde{L} + p_{k-1} + p_k,$$

ce qui nous donne $L' = \tilde{L}$. Donc $\tilde{h}$ est optimal. ∘

4.1 Exercices et implémentations Matlab

Exercice 37 Expériences. Implémenter en Matlab un algorithme qui:

1. étant donné un texte de N caractères extrait les fréquences (probabilités empiriques) de tous les caractères, y compris les espaces et chiffres. Ainsi, on déduit une distribution empirique de ces caractères $p = (p_1, \ldots, p_k)$;

2. de la même manière, un entier n étant fixé (en pratique $n = 2, 3, 4$ ou 5), calcule les fréquences de chaque suite de n symboles (bien sûr on ne stocke les fréquences que pour les suites qui apparaissent au moins une fois.) Cette distribution de probabilité s'appelle p^n;

3. calcule l'entropie binaire de $H^n := H(p^n)$, qui indique combien de bits il faut dépenser par caractère;

4. déduit quelle est la longueur théorique prévue pour le texte en bits (produit de l'entropie $H(p^n)$ par le nombre de caractères divisé par n);

5. calcule les codes de Shannon associés à p^n;

6. génère le code binaire du texte, calcule sa longueur L^n et la compare avec la longueur théorique prévue;

7. vérifie que $\frac{N}{n}H^n \leq L^n \leq \frac{N}{n}(H^n + 1)$;

8. vérifie que $n \to \frac{H^n}{n}$ est une fonction décroissante et donne le facteur optimal de compression obtenu.

Exercice 38 La définition de l'entropie, axiomatique de Shannon, opus cit. p. 49 Considérons un ensemble fini d'évènements dont les probabilités sont $p_1, \ldots, p_n$. L'entropie $H(p_1, \ldots, p_n)$ va se définir comme une mesure de l'incertitude sur celui des évènements $i = 1, \ldots, n$ qui se produira. Pour comprendre les axiomes qui nous conduisent à la définition de l'entropie, il faut prendre en compte qu'une partition d'évènements disjoints peut être l'objet de regroupements partiels. Par exemple nous pouvons grouper les k premiers évènements en un évènement unique de probabilité p'_1. On obtient ainsi une distribution de probabilité, $(p'_1 = \sum_{i=1}^{k} p_i,\ p_{k+1}, \ldots,\ p_n)$ telle qu'à son tour le premier évènement se décompose en $(\pi_1, \ldots,\ \pi_k)$ avec $\pi_i = \frac{p_i}{p_1 + \cdots + p_k}$. Dans la distribution initiale nous avions un tirage entre n évènements. Dans le second cas nous avons d'abord un tirage entre $n - k + 1$ évènements disjoints, suivi d'un second tirage entre k évènements quand le premier tirage a donné 1. En résumé, nous avons des présentations du même tirage final sous deux formes:
$(p_1, \ldots\ p_n)$, ou bien
$((p'_1, \pi_1), \ldots\ (p'_1, \pi_k), p_{k+1}, \ldots\ p_n)$
Nous formalisons alors l'entropie ou incertitude par les axiomes intuitifs suivants

1. H est continue

2. Supposons que les p_i soient égaux, $p_i = \frac{1}{n}$. Alors H doit être une fonction croissante de n

3. Si on recompose les n évènements par regroupement suivi de tirage conditionnel, comme expliqué plus haut, l'incertitude finale doit être la même. Cela nous conduit à exiger

$$H(p_1, \ldots, p_n) = H(p'_1, p_{k+1}, \ldots p_n) + p'_1 H(\pi_1, \ldots \pi_k).$$

Notre but est de démontrer qu'avec ces axiomes 1, 2 et 3 il existe une constante positive K telle que

$$H(p_1, \ldots, p_n) = -K \sum_{i=1}^{n} p_i \log p_i.$$

1) Ecrivons $H(\frac{1}{n}, \ldots \frac{1}{n}) = A(n)$. Utilisant l'axiome 3, démontrer que $A(s^m) = mA(s)$. Si t est un autre entier, on peut trouver, pour n arbitrairement grand, un m tel que $s^n \leq t^n < s^{m+1}$. Utilisant la monotonie (axiome 2) déduire que $\frac{m}{n} \leq \frac{A(t)}{A(s)} \leq \frac{m}{n} + \frac{1}{n}$ et finalement que $A(t) = K \log t$.

2) Supposons que $p_i = \frac{n_i}{\sum_1^n n_i}$ soient des probabilités rationnelles. Démontrer grâce à l'axiome 3 que

$$K \log \sum n_i = H(p_1, \ldots, p_n) + K \sum p_i \log n_i.$$

3) Traiter le cas général en approchant les p_i par des rationnels et en utilisant l'axiome 2 de continuité.

Exercice 39 Messages typiques Considérons une suite X_n de v.a.i.i.d. avec valeurs dans un ensemble fini de symboles $\mathcal{X} = \{x_1, x_2, ..., x_k\}$ et telles que $P(X_n = x_i) = p_i$, $i = 1, ..., k$. Soit $\mathcal{X}^n$ l'ensemble des suites de longueur n, que nous appelons messages. Il y a k^n tels messages. Considérons aussi l'entropie de la répartition $(p_i)_{1 \leq i \leq k}$,

$$H_2(p_1, ..., p_k) = -\Sigma_{i=1}^{i=k} p_i \log_2 p_i.$$

Cette quantité va être reliée à la probabilité d'un message long. La remarque cruciale est que les messages longs (n grand) ont tous plus ou moins la même probabilité d'être émis. Pour s'en rendre compte, il suffit d'appliquer la loi faible des grands nombres à la variable aléatoire définie comme le logarithme moyen de la probabilité d'une suite longue,

$$\frac{1}{n} \log P(X_1, \ldots, X_n) := \frac{1}{n} \sum_{i=1}^{n} \log P(X) \to E(\log P(X_i)) = \sum_i p_i \log p_i = H_2(p_1, \ldots, p_k).$$

On déduit la formule fondamentale

$$P(X_1, \ldots, X_n) = 2^{n(H_D(p_1, ..., p_k) + \epsilon(n))},$$

avec $\epsilon(n) \to 0$ quand $n \to +\infty$.

Cette observation nous conduit à définir ce que nous appellerons l'ensemble des "messages typiques",

$$C_n = \{(x_1, ..., x_n) \in \mathcal{X}^n, \ 2^{-n(H_2 + \varepsilon)} \leq p(x_1)...p(x_n) \leq 2^{-n(H_2 - \varepsilon)}\}.$$

1) Démontrer que $E(-\log_2 p_{X_n}) = H_2$.

2) déduire que

$$P((X_1,\ ...,\ X_n) \in C_n^c) \leq P(\{|\frac{1}{n}\Sigma_{l=1}^n(-\log_2 p_{X_l}) - H_2| \geq \varepsilon\}) \leq \frac{\mathrm{Var}(-\log_2 p_{X_1})}{n\varepsilon^2}.$$

3) Démontrer que

$$P((X_1,\ ...,\ X_n) \in C_n)) \geq 2^{-n(H_2+\varepsilon)}\mathrm{Card}(C_n).$$

déduire que $\mathrm{Card}(C_n) \leq 2^{(H_2+\varepsilon)n}$.

4) Réciproque. Supposons que nous ayons trouvé $\tilde{C}_n \subset \mathcal{X}^n$ tel que $\lim_{n\to+\infty} P((X_1,\ ...,\ X_n) \in \tilde{C}_n) = 1$ et $\mathrm{Card}(\tilde{C}_n) \leq 2^{Kn}$. On va montrer que $K \geq H_2$.

4a) Vérifier que $\lim_{n\to\infty} P((X_1,\ ...,\ X_n) \in C_n \cap \tilde{C}_n) = 1$.

4b) Démontrer que $P((X_1,\ ...,\ X_n) \in C_n \cap \tilde{C}_n) \leq 2^{-n(H_2-\varepsilon)}2^{Kn}$ et conclure.

5) Soient $\mathcal{X} = \{0,1\}$, $p_1 = p$, $p_2 = (1-p)$. Si $p_1 = p_2 = \frac{1}{2}$, vérifier que $H_2 = 1$ et que le nombre des suites typiques est 2^n. (En d'autres termes la compression est impossible).
Cas général : étudier la forme de $H_2(p) = -p\log p - (1-p)\log(1-p)$ et en déduire le comportement du nombre de suites typiques.

6) Application au codage. Nous allons interpréter H_2 comme la longueur moyenne de messages codés dans l'alphabet $\{0,1\}$ de manière optimale. Commençons par la description d'un codage qui réalise une telle longueur moyenne. Pour celà, fixons $\varepsilon > 0$ et choisissons n suffisamment grand, de sorte que $P(C_n^c) \leq \varepsilon$. (Expliquer pourquoi c'est possible). Donc, on attribue un code binaire à chacun des éléments de C_n. Celà implique que le nombre de codes binaires est inférieur ou égal à $2^{n(H_2+\varepsilon)}$. Considérons alors les codes non typiques, qui sont beaucoup plus nombreux! Leur nombre est de l'ordre de $k^n = 2^{n\log k}$. Donc nous pouvons les énumérer avec tout au plus k^n codes binaires distincts des précédents c'est-à-dire les nombres inférieurs à $2^{n\log k} + 2^{n(H_2+\varepsilon)} \leq 2^{n\log k+1}$. De cette manière un code est attribué à tous les éléments de $\mathcal{X}^n$. Démontrer que la longueur moyenne d'un code binaire avec ce codage est inférieure ou égale à $n(H_2 + \varepsilon(1+\log k))$. $H(p)$ peut en conséquence s'interpréter comme la longueur moyenne du code utilisée pour chaque symbole quand n est grand.

7) Finalement on se demande si on pourrait trouver un codage encore plus efficace. Si c'était possible, on aurait un sous-ensemble de messages $\tilde{C}_n$ de cardinal plus petit que 2^{nK} avec $K < H_2$ et tel que $\mathbb{P}(\tilde{C}_n) \to 1$ quand $n \to \infty$. Démontrer que ce n'est pas possible. Conclure que nous venons de démontrer que:

La longueur minimale par symbole d'un codage transmettant des messages de longueur n dans l'alphabet $\mathcal{X}$ avec la distribution $p_1, \ldots,\ p_k$ est $H_2(p_1, \ldots,\ p_k)$.

Chapter 5

Langage et communication selon Shannon

5.1 Introduction

Notre but est, suivant Shannon, d'expliquer comment nous pouvons réduire le problème de mesurer la quantité d'information transmise dans un dispositif de communication à une analyse aussi élémentaire que celle d'une distribution discrète de probabilité. Shannon réduit la communication à la transmission d'une série de symboles émis par une source aléatoire. La source émet les symboles. Chacun d'entre eux représente, par exemple, une phrase en français. L'incertitude du récepteur est grande, mais pas non plus totale, puisqu'il y a des phrases plus probables que d'autres. Par exemple dans une conversation la probabilité qu'une réponse soit "oui" ou "non" est loin d'être négligeable. L'hypothèse fondamentale est que le récepteur tout comme l'émetteur connaissent la probabilité de chaque phrase. Cette hypothèse pourrait paraître fantastique si Shannon ne nous donnait pas les moyens de calculer effectivement une bonne estimation de chaque unité significative du langage, partant des lettres pour arriver aux phrases et même aux textes. Le modèle sous-jacent est un modèle Markovien. On suppose par exemple que, étant donnée une suite de symboles $m_1 m_2 \dots m_n$, la probabilité qu'apparaisse ensuite un symbole m_{n+1} dépend seulement de m_n et pas des précédents. Evidemment, cette hypothèse markovienne est fausse, mais devient de plus en plus vraie quand la taille des symboles croît.

Grâce à l'hypothèse markovienne, la probabilité d'une suite de symboles peut se calculer par la formule

$$\mathbb{P}(m_1 m_2 \dots m_n) = \mathbb{P}(m_1)\mathbb{P}(m_2|m_1)\mathbb{P}(m_3 \mid m_2) \dots \mathbb{P}(m_n \mid m_{n-1}),$$

où $\mathbb{P}(m_2 \mid m_1) = \frac{\mathbb{P}(m_1 m_2)}{\mathbb{P}(m_1)}$ est la probabilité que m_2 suive m_1 dans un texte et $\mathbb{P}(m_1)$ est la probabilité que m_1 apparaisse dans un texte. Toutes ces probabilités peuvent s'estimer empiriquement en utilisant un texte ou un ensemble de textes, dans lesquels on calcule la fréquence de chaque mot et ensuite la fréquence de chaque paire de mots successifs. Calculer ces fréquences et les garder sous forme de dictionnaire est informatiquement possible et même facile.

Shannon, avec son génie, et dans un temps où l'informatique balbutiait, démontre avec un procédé très simple que la "chaîne de Markov du langage" peut être simulée avec la plus grande facilité si on dispose d'un livre. L'intérêt de cette simulation est de démontrer que

si on respecte les probabilités et les probabilités de transition apprises d'un texte en anglais, alors les phrases synthétisées ressemblent à l'anglais!

Le procédé de Shannon consiste à :

- choisir au hasard un mot dans un livre (ouvrir au hasard, placer le doigt au hasard)
- ouvrir de nouveau le livre au hasard et chercher le même mot ; quand on l'a trouvé, choisir le mot qui le suit
- itérer

Voici les exemples historiques simulés par Shannon avec ce procédé:

1. Approximation d'ordre zéro (symboles tirés indépendants, équiprobables):
   XFOML RXKHRRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACI-UBZLHJQD.

2. Approximation d'ordre 1 (symboles indépendants, mais avec les fréquences d'un texte anglais).
   OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL.

3. Approximation d'ordre 2 (fréquences de paires de lettres correctes pour l'anglais).
   ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEATSONARE FUSO TIZIN ANDY TOBE SEACE CTISBE.

4. Approximation d'ordre 3 (fréquences des triplets de lettres correctes).
   IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE.

5. Approximation d'ordre 1 avec des mots : la fréquence des mots est correcte
   REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFER-ENT NATURAL HERE HE THE A CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE;

6. Approximation d'ordre 2 avec des mots : les probabilités de transition entre mots sont comme en anglais.
   THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED;

Shannon observe : "The resemblance to ordinary English text increases quite noticeably at each of the above steps. Note that these samples have reasonably good structure out to about twice the range that is taken into account in their construction. Thus in 3. the statistical process insures reasonable text for two-letter sequences, but four-letter sequences from the sample can usually be fitted into good sentences. In 6. sequences of four or more words can easily be placed in sentences without unusual or strained constructions."

5.2 Exercices d'implémentation en Matlab

Exercice 40 Implémenter en Matlab un algorithme de **synthèse markovienne de texte probabilistiquement correct**. Ce programme, pour chaque k:

- calcule la probabilité empirique $\mathbb{P}(x_i \mid y^k) := \frac{\mathbb{P}(y^k x_i)}{\mathbb{P}(y^k)}$ qu'apparaisse un caractère x_i sachant qu'il est précédé par un k-gramme y^k. ($\mathbb{P}(x_i)$ est la fréquence de x_i et $\mathbb{P}(y^k)$ est la fréquence de y^k calculées dans l'exercice précédent);
- tire au sort le k-gramme initial $x_{i_1} \dots x_{i_k}$ du texte synthétisé suivant la distribution de probabilité p^k;
- tire au sort le caractère suivant selon la distribution conditionnelle $x \to \mathbb{P}(x \mid x_{i_1} \dots x_{i_k})$;
- itère: étant donné un texte déjà synthétisé de n caractères, synthétise le $n+1$-ème caractère suivant la distribution conditionnelle $x \to \mathbb{P}(x \mid x_{i_{n-k+1}} \dots x_{i_n})$;
- édite le texte synthétisé.

Exercice 41 Le but est d'apprendre un code avec un texte très long, et de coder un autre texte avec le premier.

1) Algorithme qui calcule le code de Shannon associé à une distribution de probabilité

2) Algorithme qui calcule le code de Huffman associé à une distribution de probabilité

3) Appliquer ces algorithmes à des textes pour vérifier que la longueur du binaire obtenu correspond bien à la longueur théorique optimale $nH(p)$, où n est le nombre de symboles (nombre de caractères, ou de digrammes, ou trigrammes, ou mots, ou paires de mots).

4) Pour placer le codage dans un cadre aplicatif réaliste, nous avons réalisé qu'il fallait définir le code avec un premier texte d'apprentissage qui nous donnerait une distribution de probabilité de référence. Ce texte doit être grand ou très grand, pour garantir que la grande majorité des symboles codés apparaisse plusieurs fois et permette d'estimer sa probabilité.

5) Une fois établi un code de référence (par exemple un dictionnaire de mots, et un dictionnaire de paires de mots pour le français, obtenu d'un livre), l'utiliser pour coder un AUTRE texte, et voir quel degré de compression on obtient.

Détails techniques d'expérimentation pour les textes

- Il faut établir pour chaque code un dictionnaire symbole → code et le dictionnaire inverse code → symbole
- les symboles peuvent être des monogrammes (lettres), digrammes, trigrammes, mots, ou paires de mots. Il ne convient pas d'aller plus loin parce que les probabilités deviennent trop petites et ne sont plus observables.)
- Quand on code un texte nouveau, est possible qu'apparaissent des symboles qui ne sont pas dans le dictionnaire. Dans un tel cas, le symbole reste tel quel dans le code. Ainsi le code est une suite de zéros, uns, et symboles non codés. La longueur de ce qui n'est pas codé se compte comme (nombre de lettres)×8, vu que chaque lettre se code "bêtement" avec huit bits.

– Quand on code par mots ou par paires de mots: les séparateurs (ponctuation, parenthèses, etc.) seront comptés comme mots. On néglige les majuscules puisqu'après un point il est facile de reconnaître qu'il y a une majuscule. Chaque mot commence par un espace.

Chapter 6

Messages répétés et entropie

Nous allons interpréter l'entropie $H(p)$ comme la moyenne du logarithme du nombre de messages "typiques" quand une source envoie une suite de n symboles successifs indépendants qui suivent tous la distribution p. Ensuite, on va en déduire que l'entropie est la longueur moyenne minimale, mesurée en bits/symbole, requise pour coder une source d'entropie $H(p)$. Toutes les suites $x_1x_2\dots x_n$ sont possibles, mais elles ne sont pas équiprobables. Cependant par la loi des grands nombres, la fréquence d'apparition de chaque symbole x dans $x_1\dots x_n$ tend vers $p(x)$ quand n tend vers l'infini. C'est pourquoi il y a beaucoup moins de messages typiques que de messages possibles. Nous allons voir que le nombre de messages typiques a pour ordre de grandeur $2^{nH(p)}$ tandis que le nombre de messages possibles est $(\mathrm{Card}X)^n = 2^{n\log \mathrm{Card}(X)}$.

6.1 Messages typiques

Considérons une suite X_n de v.a.i.i.d. avec valeurs dans un ensemble fini de symboles $\mathcal{X}$ et telles que $\mathbb{P}(X_n = x) = p(x)$. Soit $\mathcal{X}^n$ l'ensemble des suites de longueur n, que nous appelerons *messages de longueur n,* ou simplement *messages.* Il y a k^n messages possibles. Considérons aussi l'entropie de la distribution $p = (p(x))_{x\in\mathcal{X}}$,

$$H(p) = -\Sigma_{x\in\mathcal{X}} p(x)\log p(x).$$

Cette quantité va être interprétée en relation avec la probabilité d'un message long. La remarque cruciale est que les messages longs (n grand) ont tous plus ou moins la même probabilité d'être émis. Pour s'en rendre compte, il suffit d'appliquer la loi forte des grands nombres à la variable aléatoire définie comme le logarithme moyen de la probabilité d'une suite longue,

$$\frac{1}{n}\log p(X_1,\dots,\ X_n) := \frac{1}{n}\sum_{i=1}^{n}\log p(X_i) \to E(\log p(X)) = \sum_{x\in\mathcal{X}} p(x)\log p(x) = H(p).$$

On en déduit la formule fondamentale

$$p(X_1,\dots,\ X_n) = 2^{n(H(p)+\epsilon(n))},$$

avec $\epsilon(n) \to 0$ quand $n \to +\infty$. (Ici les convergences sont des convergences presque sûres).

Cette observation, que nous n'allons pas utiliser (nous utiliserons seulement la loi faible des grands nombres), nous conduit néanmoins à la définition suivante.

Définition 6.1 *Pour chaque $\varepsilon > 0$ nous appellerons ensemble des "messages typiques",*

$$C_n = \{(x_1,\ ...,\ x_n) \in \mathcal{X}^n,\ 2^{-n(H(p)+\varepsilon)} \leq p(x_1 \dots x_n) = p(x_1)...p(x_n) \leq 2^{-n(H(p)-\varepsilon)}\}.$$

Lemme 6.1 *L'ensemble C_n des messages typiques associé avec p, n et ε vérifie*

1. $\mathbb{P}(C_n) \geq 1 - \varepsilon$ *pour n suffisamment grand ;*
2. *Card*$(C_n) \leq 2^{(H(p)+\varepsilon)n}$

Démonstration Considérons la variable aléatoire $S_n = \sum_{i=1}^n \log p(X_i)$ et passons au logarithme dans les inégalités définissant C_n. On voit que

$$C_n = \{(x_1,\ ...,x_n) \mid -n(H(p)+\varepsilon) \leq \sum_{i=1}^n \log p(x_i) \leq -n(H(p)-\varepsilon)\}.$$

Donc

$$C_n = \{|\frac{1}{n}\Sigma_{i=1}^n(-\log p(X_i)) - H(p)| \leq \varepsilon\}.$$

Observons que $E(-\log p(X)) = H(p)$. Cela provient directement de la définition de l'espérance d'une variable aléatoire $f(X)$ quand X est une autre variable aléatoire à valeurs dans $\mathcal{X}$, $\mathbb{E}f(X) = \sum_{x\in\mathcal{X}} f(x)\mathbb{P}(X = x)$. Ici, on l'applique à $f(x) = -\log p(x)$.

Nous pouvons appliquer l'inégalité de Tchebychev

$$\mathbb{P}(|\frac{S_n}{n} - \mathbb{E}(-\log p(X))| \geq \varepsilon) \leq \frac{\sigma^2(-\log p(X))}{n\varepsilon^2} \to 0 \text{ quand } n \to \infty,$$

ce qui nous donne

$$\mathbb{P}((X_1,\ ...,\ X_n) \in C_n^c) = \mathbb{P}(\{|\frac{1}{n}\Sigma_{i=1}^n(-\log p(X_i)) - H(p)| > \varepsilon\}) \leq \frac{\sigma^2(-\log p(X))}{n\varepsilon^2}.$$

Fixant ε et choisissant n suffisamment grand nous obtenons $\mathbb{P}(C_n) \geq 1 - \varepsilon$.

Dans C_n les suites ont toutes plus ou moins la probabilité $2^{-nH(p)}$ et, plus précisément:

$$\mathbb{P}(C_n)) \geq \sum_{(x_1\dots x_n)\in C_n} p(x_1 \dots x_n) \geq 2^{-n(H(p)+\varepsilon)}\text{Card}(C_n).$$

Comme $\mathbb{P}(C_n) \leq 1$ on déduit que $\text{Card}(C_n) \leq 2^{(H(p)+\varepsilon)n}$.

∘

Nous allons interpréter $nH(p)$ comme la longueur moyenne de messages de n symboles codés dans l'alphabet $\{0, 1\}$ de manière optimale.

Lemme 6.2 *Soit $\mathcal{X}$ un ensemble de messages élémentaires ou symboles et X une source émettant des messages répétés avec la distribution $p = (p(x))_{x\in\mathcal{X}}$. Pour tout $\varepsilon > 0$ nous pouvons, si n est suffisamment grand, coder les messages répétés de telle manière que la longueur moyenne par symbole soit inférieure ou égale à $H(p) + \varepsilon$.*

Démonstration Pour réaliser le codage annoncé, fixons $1 > \varepsilon > 0$ et choisissons n suffisamment grand pour assurer que $\mathbb{P}(C_n^c) \leq \varepsilon$ et $\text{Card}(C_n) < 2^{n(H(p)+\varepsilon)}$. Donc nous pouvons coder tous les éléments de C_n en utilisant les nombres binaires supérieurs ou égaux à $2^{[n(H(p)+\varepsilon)+1]}$ et strictement inférieurs à $2^{[n(H(p)+\varepsilon)+2]}$. Ce sont des nombres binaires qui ont tous la même longueur $[n(H(p)+\varepsilon)+2]$. Donc ils forment un code préfixe. De plus au moins un de ces codes, m, n'est pas utilisé.

Considérons alors les codes non typiques, qui sont beaucoup plus nombreux! Leur nombre est strictement inférieur à $k^n = 2^{n \log k}$. Comme pour les éléments de C_n, nous pouvons les énumérer avec des nombres binaires ayant tous la même longueur $[n \log k + 2]$. Pour obtenir un codage préfixe, il convient d'ajouter à tous ces nombres le préfixe m. Ainsi, nous obtenons des codes de longueur $[n \log k + 2] + [nH(p) + 2]$.

Nous avons attribué un code à tous les éléments de $\mathcal{X}^n$. La longueur de chaque code binaire de C_n est inférieure ou égale à $n(H(p)+\varepsilon)+2$. La longueur des autres codes est inférieure ou égale à $n(\log k + H(p)) + 4$. Comme $\mathbb{P}_n(C_n) \leq 1$ et $\mathbb{P}_n(C_n^c) \leq \varepsilon)$, la longueur moyenne $\mathbb{E}l_n$ d'un message de n symboles avec ce codage vérifie

$$\mathbb{E}l_n \leq \varepsilon(n(\log k + H(p)) + 4) + (1-\varepsilon)(n(H(p)+\varepsilon)+2), \text{ ce qui nous donne:}$$

$$\limsup_{n\to\infty} \frac{\mathbb{E}l_n}{n} \leq (1-\varepsilon)(H(p)+\varepsilon) + \varepsilon(\log k + H(p)).$$

Comme p et k sont fixes et ε arbitrairement petit, nous obtenons le résultat annoncé. ∘

Maintenant il faut démontrer l'inégalité inverse, c'est-à-dire que nous ne *pouvons pas* coder les messages de n symboles avec strictement moins que $H(p)$ bits par symbole.

Lemme 6.3 *Pour tout $\epsilon > 0$, si n est suffisamment grand, l'espérance $\mathbb{E}l_n$ de la longueur de tout codage binaire h de C_n vérifie*

$$\mathbb{E}l_n \geq n(H(p) - \varepsilon).$$

Démonstration Fixons $\varepsilon > 0$ et considérons l'ensemble des messages typiques C_n. La probabilité π_i de chacun de ces messages, énumérés de $i = 1$ à N, vérifie (par définition de l'ensemble des messages typiques)

$$\pi(i) \geq 2^{-n(H(p)+\varepsilon)}. \tag{6.1}$$

Le cardinal N de l'ensemble C_n vérifie $2^{n(H(p)-\varepsilon)} \leq N \leq 2^{n(H(p)+\varepsilon)}$. De plus, nous savons que

$$\sum_{i=1}^{N} \pi_i = \mathbb{P}(C_n) \geq 1 - \varepsilon \tag{6.2}$$

pour n suffisamment grand. Soit $h(i)$ un codage binaire de C_n. Pour éviter que certains des codes commencent par des zéros, nous pouvons, pour que tous ces codes binaires distincts deviennent des nombres binaires distincts, juxtaposer à tous les $h(i)$ un 1 à gauche, ce qui augmente leur longueur de 1. Supposons sans perte de généralité que les nouveaux codes $1h(i)$ sont ordonnés en ordre croissant. Donc $1 \leq i \leq N$ est le rang de $1h(i)$. Finalement

remarquons que $[\log i]$ est la longueur de i écrit comme un nombre binaire. Comme $1h(i) \geq i$, nous avons $l(1h(i)) \geq [\log i]$, ce qui donne

$$\sum_{i=1}^{N} \pi_i l(h(i)) \geq \sum_{i=1}^{N} \pi_i [\log i] - 1 \geq \sum_{i=1}^{N} \pi_i \log i - 2, \tag{6.3}$$

puisque $[r] \geq r - 1$. On a

$$\sum_{i=1}^{N} \pi_i \log(i) \geq (\log N - k) \sum_{i=N2^{-k}}^{N} \pi_i. \tag{6.4}$$

Mais des estimations sur π_i et N, il vient

$$\sum_{1}^{2^{-k}N} \pi_i \leq 2^{-n(H(p)-\varepsilon)} 2^{-k} 2^{n(H(p)+\varepsilon)} = 2^{2n\varepsilon - k}. \tag{6.5}$$

De (6.4), (6.5) et (6.2), on tire

$$\sum_{i=1}^{N} \pi_i \log(i) \geq (\log N - k)(1 - 2^{2n\varepsilon - k} - \varepsilon) \geq (n(H(p) - \varepsilon) - k)(1 - 2^{2n\varepsilon - k} - \varepsilon).$$

Choisissant $k =: 4n\varepsilon$, on obtient de cette dernière minoration et de (6.3)

$$\liminf_{n \to \infty} \frac{1}{n} \sum_{i=1}^{N} \pi_i l(h(i)) \geq (H(p) - 5\varepsilon)(1 - \varepsilon).$$

Cette inégalité étant vraie pour ε arbitrairement petit, on conclut.

∘

En combinant les deux lemmes précédents, nous pouvons démontrer le résultat fondamental de Shannon :

Théorème 6.1 *L'espérance minimale $\mathbb{E}l$ de la longueur par symbole d'un message émis n fois par une source d'entropie $H(p)$ est égal à $(H(p) + \varepsilon(n))$ avec $\varepsilon(n) \to 0$ quand $n \to \infty$.*

Démonstration Considérons par chaque n un codage h de longueur minimale de $\mathcal{X}^n$. Comme $\mathcal{X}^n \supset C_n$, ce codage code C_n et nous pouvons appliquer le lemme 6.3. Donc sa longueur par symbole est supérieure ou égale à $H(p) - \varepsilon$, pour n grand. Comme h est un codage optimal, l'espérance de sa longueur est aussi pour n grand inférieure ou égale à $H(p) + \varepsilon$ par le lemme 6.2. Combinant les deux résultats nous voyons que pour tout $\varepsilon > 0$ et pour n suffisamment grand, $H(p) - \varepsilon \leq \mathbb{E}l \leq H(p) + \varepsilon$.

∘

6.2 Exercices et implémentations Matlab

Exercice 42 On va montrer que la longueur moyenne $\lambda(N)$ des nombres binaires inférieurs à N vérifie $\lambda(N) \sim \log N$, où $\varepsilon(N) \to 0$ quand $N \to \infty$.

1) Soit m tel que $2^m \leq N < 2^{m+1}$. Vérifier que les nombres binaires plus grands que 2^{j-1} et strictement plus petits que 2^j ont une longueur égale à j et que leur nombre est égal à 2^{j-1}. Déduire que

$$m+1 \geq \log N \geq \lambda(N) = \frac{1}{N}(\sum_{j=1}^{m} j2^{j-1} + (m+1)(N-2^m+1)). \tag{6.6}$$

2) Soit $p \in I\!N$ tel que $2^{-p} < \varepsilon$. Ecrire

$$\lambda(N) \geq \frac{1}{N}(\sum_{j=m-p+1}^{m} j2^{j-1} + (m+1)(N-2^m+1))$$

et déduire que $\lambda(N) \geq (m-p+1)(1-2^{-p})$.

3) Déduire que

$$1 \geq \lim\sup_{N\to\infty} \frac{\lambda(N)}{\log N} \geq (1-2^{-p})$$

et conclure.

Exercice 43 Le codage de Lempel Ziv
Ziv et Lempel ont inventé un algorithme de codage universel qui est optimal, au sens très général que pour n'importe quelle suite ergodique, son taux de compression tend vers l'entropie de la source. La preuve qu'il en est ainsi est donnée dans Cover-Thomas, pp. 319 à 326. Cette preuve n'est pas difficile. De plus, il est immédiat de vérifier que cette preuve s'applique directement à l'hypothèse plus simple qu'une source est markovienne d'ordre k. Dans ce cas, la preuve mentionnée montre que si on considère le vecteur aléatoire $(X_1, \ldots, X_n)$ fait des n valeurs produites par la source, et si on note $l(X_1, X_2, \ldots, X_n)$ la longueur du code de Lempel-Ziv de ce vecteur, alors

$$\limsup \frac{1}{n} l(X_1, \ldots, X_n) \leq H(\mathcal{X}),$$

où l'entropie de la source markovienne $\mathcal{X}$ d'ordre k est définie par $H(\mathcal{X}) = H(X_k \mid X_{k-1}, \ldots, X_0)$. Cette définition de l'entropie d'une source markovienne est parfaitement transparente: elle mesure l'incertitude laissée sur X_n quand on connaît les k précédents, $X_{n-1}, \ldots, X_{n-k}$.

L'algorithme de Lempel-Ziv est facile à décrire. Etant donnée une suite de longueur n, elle se décompose en chaînes de 0 et 1 de longueur minimale et telles que chacune d'entre elles apparaît pour la première fois dans la suite. Ainsi la suite 1011001010001111010111000110101011101 se décompose en

$$(1)(0)(11)(00)(10)(100)(01)(111)(010)(1110)(001)(101)(010)$$

1) Démontrer que chacune des chaînes de la décomposition s'écrit w0 ou w1 où w est une chaîne apparue antérieurement. On appelle $c(n)$ le nombre de chaînes de la suite.

2) Alors on code la suite en donnant comme code à chaque chaîne s'écrivant w0 ou w1 le numéro d'ordre dans la suite de la chaîne antérieure w, suivi de son dernier bit (0 ou 1). Le code de la suite que nous avons donnée comme exemple est donc

$(0000, 1)(0000, 0)(0001, 1)(0010, 0)(0001, 0)(0101, 0)(0010, 1)(0011, 1)(0111, 0)(1000, 0)$
$(0010, 1)(0011, 1)(0101, .)$

La dernière chaîne, incomplète, est laissée telle quelle.

3) Démontrer que la longueur totale de la suite comprimée est $c(n)(\log c(n) + 1)$ bits.

4) Ecrire l'algoritme en Matlab, l'appliquer à un texte très grand et comparer la longueur en bits résultante avec celle obtenue avec un code de Huffman.

Chapter 7

La communication sûre est possible malgré le bruit

7.1 Transmission dans un canal bruité

Le problème principal que Shannon aborde et résoud est la transmission dans un canal avec bruit, c'est-à-dire subissant des erreurs aléatoires durant la transmission. Le problème principal est de décider si la transmission est possible et, surtout, à quel prix en termes de redondance. Shannon part de ses expériences de jeunesse quand, enfant dans une ferme americaine immense, il comuniquait avec ses copains par télégraphe grâce aux fils eléctriques des haies de pâturages. Il réalisa qu'un message en anglais très incomplet, où presque la moitié des lettres est incorrecte ou manquante, peut être reconstruit correctement par le récepteur. Celà est dû à la redondance du langage. Le destin de Shannon était de faire très jeune cette observation empirique, et de la formaliser mathématiquement bien des années plus tard, quand il réussit à calculer l'entropie de l'anglais de tous les jours. On vérifie en effet que le degré de compression d'un texte anglais courant atteint cinquante pour cent. Une autre intuition acquise par l'expérience est celle que Shannon traduira dans son fameux théorème: plus une suite est longue, et plus elle est facile à reconstruire (et donc aussi à comprimer).

Le théorème de Shannon, bien qu'obtenu par des arguments d'existence mathématique non constructifs, provoqua l'enthousiasme des ingénieurs par sa simplicité et par le défi technologique ainsi lancé. Shannon considère une source émettrice X, ou source d'entrée. Mais s'il y a du bruit dans le canal la réception est représentée par une variable aléatoire Y distincte de X, que l'on appelle la sortie du canal. Pour mesurer l'incertitude sur X laissée quand on observe la sortie Y, Shannon introduit la notion *d'entropie relative*, $H(X|Y)$. Elle peut aussi être utilisée sous la forme $H(Y|X)$, qui mesure l'incertitude qui sera laissée sur la sortie Y connaissant l'entrée X. $H(Y|X)$ mesure donc l'incertitude causée par le bruit.

Dans son théorème fondamental, Shannon démontre qu'il est possible de coder les messages émis par X de telle manière que la proportion d'erreurs dans la transmission soit arbitrairement basse. En d'autres termes la communication sûre malgré le bruit est possible! Shannon propose de mesurer la capacité d'un canal de transmission bruité comme $\max_X H(X) - H(X|Y)$, le maximum étant pris parmi toutes les sources possibles prises comme entrées. Cette capacité est donc obtenue en soustrayant de la quantité d'information émise $H(X)$ l'incertitude $H(X|Y)$ laissée sur X après l'observation de Y. Le grand théorème de Shannon est quantitatif: toute source d'entropie inférieure à la capacité peut être transmise

intégralement dans le canal.

Messages et sorties typiques pour un canal avec bruit
Considérons une source X de loi $p(x)$, $x \in \mathcal{X}$. Un canal de transmission bruité transmet X avec des erreurs. Le résultat observé est Y, de loi $p(y)$, $y \in \mathcal{Y}$. La loi conjointe de X et Y est $p(x,y) = \mathbb{P}(X = x, Y = y)$, définie sur $\mathcal{X} \times \mathcal{Y}$. Dans le cas d'un canal sans bruit nous aurions $p(x,x) = p(x)$ par tout x et $p(x,y) = 0$ si $x \neq y$. Las entropies de deux variables et de leur couple s'écrivent $H(X)$, $H(Y)$ et $H(X,Y)$. Dans la communication répétée, si la source émet des messages composés de n symboles équidistribués et indépendants, le message résultant vu comme une variable aléatoire s'écrit X^n et la sortie Y^n. La séquence Y^n est également composée de symboles indépendants. Plus précisément, les couples (X_1, Y_1), (X_2, Y_2), ... (X_n, Y_n) sont supposés indépendants. X^n et Y^n sont à valeurs dans $\mathcal{X}^n$ et $\mathcal{Y}^n$ respectivement. Les valeurs possibles de X^n s'écriront $x^n \in \mathcal{X}^n$ et celles de Y^n, $y^n \in \mathcal{Y}^n$.

Définition 7.1 *Nous appellerons ensemble de* paires entrée-sortie typiques relativement à la distribution $p(x,y)$ *l'ensemble* A_ε^n *des suites* $\{(x^n, y^n)\}$ *dont les probabilités sont typiques au sens suivant:* $A_\varepsilon^n = B_\varepsilon^n \cap C_\varepsilon^n \cap D_\varepsilon^n$ *avec*

$$B_\varepsilon^n = \{(x^n, y^n) \in \mathcal{X}^n \times \mathcal{Y}^n : | -\frac{1}{n} \log p(x^n, y^n) - H(X,Y)| < \varepsilon\}; \tag{7.1}$$

$$C_\varepsilon^n = \{(x^n, y^n) \in \mathcal{X}^n \times \mathcal{Y}^n : | -\frac{1}{n} \log p(x^n) - H(X)| < \varepsilon\}; \tag{7.2}$$

$$D_\varepsilon^n = \{(x^n, y^n) \in \mathcal{X}^n \times \mathcal{Y}^n : | -\frac{1}{n} \log p(y^n) - H(Y)| < \varepsilon\}; \tag{7.3}$$

où $p(x^n) =: \mathbb{P}_1(\{x^n\}) = \Pi_{i=1}^n p(x_i)$, $p(y^n) =: \mathbb{P}_2(\{y^n\}) = \Pi_{i=1}^n p(y_i)$, $p(x^n, y^n) = \mathbb{P}(\{x^n, y^n\}) = \Pi_{i=1}^n p(x_i, y_i)$ *en utilisant l'indépendance des messages successifs, et* $\mathbb{P}_1$ *et* $\mathbb{P}_2$ *sont les marginales de* $\mathbb{P}$.

Lemme 7.1 *Soit* (X^n, Y^n) *une suite de longueur* n *de v.a.i.i.d. suivant la loi* $\mathbb{P}(\{x,^n, y^n\}) = p(x^n, y^n) = \Pi_{i=1}^n p(x_i, y_i)$. *Alors*

1. $\mathbb{P}((X^n, Y^n) \in A_\varepsilon^n) \to 1$ *quand* $n \to \infty$.
2. $Card(\{y^n,\ (x^n, y^n) \in A_\varepsilon^n\}) \leq 2^{n(H(Y|X)+2\varepsilon)}$
3. $Card(\{x^n,\ (x^n, y^n) \in A_\varepsilon^n\}) \leq 2^{n(H(X|Y)+2\varepsilon)}$

La relation 2. indique une borne cruciale sur le nombre de messages typiques y^n que peut causer un message typique x^n. Dans la démonstration qui suit et dans le reste de ce chapitre nous écrirons $2^{a \pm \varepsilon}$ pour indiquer un nombre quelconque b tel que $2^{a-\varepsilon} \leq b \leq 2^{a+\varepsilon}$. Avec cette notation, on a la relation $2^{a \pm \varepsilon} \times 2^{b \pm \varepsilon} = 2^{a+b \pm 2\varepsilon}$.

Démonstration On va montrer la première relation. Considérons $\pi_1 : (x^n,\ y^n) \to x^n$ et $\pi_2 : (x^n,\ y^n) \to y^n$ les aplications de projection de $\mathcal{X}^n \times \mathcal{Y}^n$ sur $\mathcal{X}^n$ et $\mathcal{Y}^n$ respectivement. Le résultat du lemme 6.1 permet d'affirmer que quand $n \to \infty$,

$$\mathbb{P}(B_\varepsilon^n) \to 1 \tag{7.4}$$

$$\mathbb{P}(C_\varepsilon^n) = \mathbb{P}(\pi_1(C_\varepsilon^n) \times Y^n) = \mathbb{P}_1(\pi_1(C_\varepsilon^n)) \to 1 \tag{7.5}$$

$$\mathbb{P}(D_\varepsilon^n) = \mathbb{P}(X^n \times \pi_2(D_\varepsilon^n)) = \mathbb{P}_2(\pi_2(D_\varepsilon^n)) \to 1. \tag{7.6}$$

En effet, $\pi_1(C_\varepsilon^n)$ est tout bonnement l'ensemble des messages typiques pour $\mathbb{P}_1$ et de même $\pi_2(D_\varepsilon^n)$ est l'ensemble des messages typiques pour $\mathbb{P}_2$. La relation 1. se déduit alors de la remarque générale que si des suites d'ensembles B_i^n, $i = 1, \ldots, k$ vérifient $\mathbb{P}(B_i^n) \to 1$ quand $n \to \infty$, alors $\mathbb{P}(\cap_{i=1}^k B_i^n) \to 1$ aussi. On applique cette remarque à l'intersection des trois ensembles précédents,

$$A_n^\varepsilon = B_\varepsilon^n \cap (\pi_1(C_\varepsilon) \times Y^n) \cap (X^n \times \pi_2(D_\varepsilon^n)).$$

Maintenant passons à la relation 2. Par les hypothèses de typicité, $\mathbb{P}(X^n = x^n) = p(x^n) = 2^{-n(H(X)\pm\varepsilon)}$ et $\mathbb{P}((X^n, Y^n) = (x^n, y^n)) = p(x^n, y^n) = 2^{-n(H(X,Y)\pm\varepsilon)}$. Donc

$$p(x^n) = \sum_{y^n} p(x^n, y^n) \geq \sum_{y^n,\ (x^n,y^n)\in A_\varepsilon^n} p(x^n, y^n) \geq \mathrm{Card}(\{y^n,\ (x^n, y^n) \in A_\varepsilon^n\}) \inf_{(x^n,y^n)\in A_\varepsilon^n} p(x^n, y^n).$$

Mais $p(x^n) = 2^{n(H(X)\pm\varepsilon)}$ et $p(x^n, y^n) = 2^{n(H(X,Y)\pm\varepsilon)}$. Donc

$$\mathrm{Card}(\{y^n,\ (x^n, y^n) \in A_\varepsilon^n\}) \leq 2^{-n(-H(X,Y)+H(X)+2\varepsilon)} = 2^{n(H(Y|X)+2\varepsilon)}.$$

La démonstration de 3. est strictement analogue de celle de 2. ∘

Capacité d'un canal bruité

Considérons un canal discret, c'est-à-dire un système disposant d'un alphabet d'entrée $\mathcal{X}$ et d'un alphabet de sortie $\mathcal{Y}$ ainsi que d'une matrice de probabilités de transition $p(y|x)$ donnant la probabilité d'observer un symbole y comme sortie quand le symbole x a été émis. Nous dirons qu'un tel canal est "sans mémoire", car la distribution de probabilité de la sortie dépend uniquement de l'entrée et est indépendante des entrées et sorties précédentes. Nous allons alors définir le *transmission rate* du canal, ou taux de transmission étant données la source X et sa sortie correspondante Y par

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = H(X) + H(Y) - H(X,Y).$$

The first defining expression has already been defined as the amount of information sent less the uncertainty of what was sent. The second measures the amount received less the part of this which is due to noise. The third is the sum of the two amounts less the joint entropy and therefore in a sense is the number of bits per second common to the two. Thus, all three expressions have a certain intuitive significance. $H(Y \mid X)$ est donc interprétée comme une mesure de la quantité de bruit, sans que l'on ait pour cela besoin de modéliser le bruit par lui-même. Rappelons que $I(X;Y)$ est une formule symétrique en X et Y, que l'on appelle aussi information mutuelle, et qui est nulle si et seulement si X et Y sont indépendantes.

Définition 7.2 *Nous appellerons capacité d'un canal discret sans mémoire la quantité*

$$C = \max_{p(x)} I(X;Y),$$

où le maximum se calcule sur toutes les distributions de probabilité possibles $p(x)$, $x \in \mathcal{X}$ en entrée.

La première chose qu'il faut remarquer est que ce problème est un problème d'optimisation dans $\mathbb{R}^{\mathrm{Card}(\mathcal{X})}$, puisque nous connaissons les valeurs de $p(y|x)$.

Exemple 1: transmission sans bruit
Si le canal transmet intégralement une entrée binaire sans erreur, la matrice de transition est l'identité. Alors $Y = X$ et donc $I(X;X) = H(X) - H(X|X) = H(X)$. Alors la capacité est maximale quand l'entropie de la source émettrice est maximale, ce qui implique ce que nous attendons, à savoir $p(0) = p(1) = \frac{1}{2}$ et $C = H(p) = 1$ bit.

Exemple 2 : canal binaire symétrique
Prenons $\mathcal{X} = \mathcal{Y} = \{0,\ 1\}$ et

$$p(y = 1|x = 1) = p(y = 0|x = 0) = 1 - p, \quad p(y = 1|x = 0) = p(y = 0|x = 1) = p.$$

Comme l'entropie d'une variable de Bernoulli $\mathcal{B}(p, 1-p)$ est $H(p) = -p\log p-(1-p)\log(1-p)$, nous obtenons

$$H(Y)-H(Y|X) = H(Y)-\sum p(x)H(Y|X = x) = H(Y)-\sum p(x)H(p) = H(Y)-H(p) \leq 1-H(p).$$

Il y aura égalité dans cette inégalité si et seulement si X est uniforme, puisqu'alors Y est aussi uniforme et $H(Y) = 1$. Donc $C = 1 - H(p)$. Quand $p = \frac{1}{2}$, la capacité est nulle et le canal ne transmet rien. Dans tous les autres cas, il y a transmission d'information.

7.2 Le théorème fondamental pour un canal discret bruité

Théorème 7.1 *(Shannon pages 71, 72, 73) Considérons un canal discret de capacité C et une source discrète d'entropie E.*
(a) Si $E \leq C$ il existe un système de codage tel que la sortie de la source se transmette par le canal avec une fréquence arbitrairement petite d'erreur.
(b) Si $E > C$ il est encore possible de coder la source de telle sorte que l'incertitude sur les messages $H(X|Y)$ soit inférieure à $E - C + \varepsilon$, où ε est arbitrairement petit.
(c) Il n'y a pas de méthode de codage permettant d'atteindre une incertitude sur les messages, $H(X|Y)$, inférieure à $E - C$.

Démonstration (a) Considérons une source X_0 de taux de transmission très proche de la capacité maximale C et soit Y_0 sa sortie. Nous allons utiliser X_0 comme entrée dans le canal. Considérons toutes les suites possibles transmises et reçues, de longueur n. Dans tout ce qui suit, les ε, η, θ seront des réels positifs qui tendent vers zéro quand le taux de la source X_0 se rapproche de la capacité maximale du canal ou quand la durée de transmission $n \to \infty$. Chaque fois que nous aurons $C\varepsilon$ avec une constante C indépendante de n, nous écrirons ε pour $C\varepsilon$ afin d'alléger la notation, s.p.d.g..

Soit A_ε^n l'ensemble des paires typiques (x^n, y^n). (Definition 7.1).

1. Les suites transmises peuvent être dans deux groupes : les messages typiques dont le nombre est $2^{n(H(X_0)\pm\varepsilon)}$ et les autres, dont la probabilité totale est inférieur à η.

2. De la même manière, les suites reçues forment un ensemble de suites typiques de nombre $2^{n(H(Y_0)\pm\varepsilon)}$ et de probabilité totale supérieur à $1 - \eta$. Nous allons appeler $\mathcal{M}_0$ cet ensemble de messages typiques.

3. Par le lemme 7.1, propriété 3, chaque sortie typique a pu être produite par tout au plus $2^{n(H(X_0|Y_0)+\varepsilon)}$ entrées.

4. De la même manière, chaque entrée typique peut produire tout au plus $2^{n(H(Y_0|X_0)\pm\varepsilon)}$ sorties (mais nous n'allons pas utiliser cette dernière propriété.)

Considérons alors une source X d'entropie $E < C$. Ecrivons $E = C - 2\theta$ et choisissons X_0 tel que son taux de transmission vérifie $H(X_0) - H(X_0|Y_0) > C - \theta$. Donc

$$E - (H(X_0) - H(X_0|Y_0)) < -\theta. \tag{7.7}$$

En un temps de transmission n, la source X peut produire $2^{n(E\pm\varepsilon)}$ messages typiques. Nous allons appeler $\mathcal{M}$ cet ensemble de messages typiques et nous allons les coder en les associant à des messages typiques de longueur n de la source X_0, qui seront utilisés comme codes. Chaque codage est une application $\mathbb{C} : \mathcal{M} \to \mathcal{M}_0$ obtenue en tirant au sort (avec une distribution uniforme) pour chaque message dans $\mathcal{M}$ un élément de $\mathcal{M}_0$ qui est donc choisi aléatoirement comme son code. Les autres messages, non typiques, ne sont tout bonnement pas codés, ce qui nous donne de toutes façons une probabilité d'erreur inférieure à η.

Nous allons évaluer la probabilité d'erreur, c'est-à-dire la probabilité qu'un message donné y_1 ait été associé à un message de $\mathcal{M}$. Pour tout message effectivement observé y_1, cette probabilité s'interprète comme une probabilité d'erreur, à savoir comme la probabilité que y_1 ait été associé à deux messages de $\mathcal{M}$. Par la conclusion 3. du lemme 7.1, nous savons que y_1 n'a pu être produit par plus de $2^{n(H(X_0|Y_0)+\varepsilon)}$ messages x_0 dans $\mathcal{M}_0$. Mais la probabilité que chaque $x_0 \in \mathcal{M}_0$ soit un code est $2^{n(E-H(X_0)\pm\varepsilon)}$. En effet, nous avons distribué $2^{n(E\pm\varepsilon)}$ messages uniformément sur $2^{n(H(X_0)\pm\varepsilon)}$ codes. Cela implique que la probabilité que y_1 soit le code d'un autre message de X (en plus du message dont il est déjà le code) est inférieure à

$$\mathbb{P}(\text{erreur sur } y_1) \leq 2^{n(E-H(X_0)\pm\varepsilon)} 2^{n(H(X_0|Y_0)+2\varepsilon)}.$$

Donc par (7.7),

$$\mathbb{P}(\text{erreur sur } y_1) \leq 2^{n(E-H(X_0)+\varepsilon+H(X_0|Y_0)+2\varepsilon)} = 2^{-n(\theta-3\varepsilon)}$$

Comme η a été fixé (arbitrairement petit) et, η une fois fixé, comme nous pouvons choisir ε aussi petit que désiré pour n assez grand, nous déduisons que la probabilité d'erreur pour chaque message est arbitrairement petite, ce qui démontre (a) pour $E < C$.

(b) Si $E \geq C$, on peut toujours appliquer la construction précédente mais ne pouvons coder plus de $2^{n(C-\varepsilon)}$ messages des $2^{n(E\pm\varepsilon)}$ typiques. Celà enlève pas mal d'intérêt au codage, puisque la majorité des messages typiques ne sont pas transmis!

(c) Supposons que l'on puisse transmettre par un canal de capacité C les messages d'une source X_0 d'entropie $E = C + a$, avec $a > 0$ et que l'incertitude sur le message vérifie $H(Y_0|X_0) = a - \varepsilon$ avec $\varepsilon > 0$. Alors $H(X_0) - H(X_0|Y_0) = C + \varepsilon$ et cela contredit la définition de C comme le maximum de $H(X_0) - H(X_0|Y_0)$ pour toutes les sources en entrée.

Exercice 44 La question de la fin

1) Expliquer le raisonnement par lequel Shannon conclut sa démonstration:

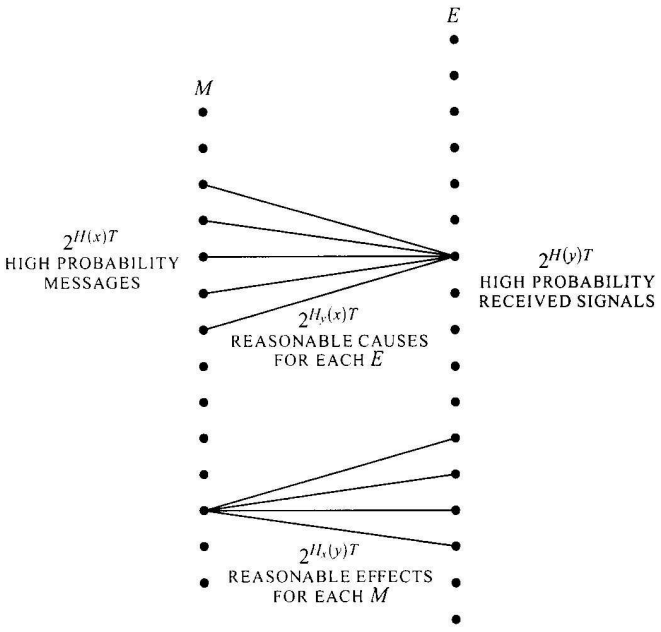


Figure 7.1: Le schéma original de Shannon. T, le temps de transmission est ce que nous avons appelé n. $H_y(x)$ est dans nos notations $H(X \mid Y)$.

Actually more has been proved than was stated in the theorem. If the average of a set of positive numbers is within ε of zero, a fraction of at most $\sqrt{\varepsilon}$ can have values greater than $\sqrt{\varepsilon}$. Since ε is arbitrarily small we can say that almost all the systems are arbitrarily close to the ideal.

2) En d'autres termes presque tous les codes considérés, choisis au hasard, sont des codes qui corrigent spontanément les erreurs! Alors, pourquoi est-il difficile en pratique de concevoir un code correcteur d'erreurs?

Chapter 8

Séries de Fourier (Révision)

On considère l'espace de Hilbert hermitien $L^2([-\pi,\pi])$ que l'on notera aussi $L^2(-\pi,\pi)$. Ces fonctions sont à valeurs réelles ou complexes. On va montrer que le système orthonormé

$$\frac{1}{(2\pi)^{\frac{1}{2}}}(e^{int})_{n\in\mathcal{Z}}$$

est une base hilbertienne de $L^2(-\pi,\pi)$. Cette base s'appelle la base de Fourier. On notera

$$c_n(f)=\frac{1}{2\pi}\int_{-\pi}^{\pi}f(x)e^{-inx}dx,$$

en sorte que pour toute f dans $L^2([-\pi,\pi])$ on puisse écrire

$$f(x)=\sum_{n\in\mathbf{Z}}c_n(f)e^{inx},$$

la série précédente convergeant au sens L^2. Les $c_n(f)$ s'appellent les coefficients de Fourier de f et sont proportionnels aux coordonnées de f dans la base de Fourier.

Pour montrer ce résultat, on va commencer par analyser le comportement des coefficients de Fourier selon la régularité de f.

Lemme 8.1 *Riemann-Lebesgue (Lemme de) (Lemme de Riemann-Lebesgue)*
i) On pose pour $f\in L^1(\mathbb{R})$,

$$\hat{f}(\xi)=\int_{\mathbb{R}}f(x)e^{-i\xi x}dx.$$

Si $f\in\mathcal{C}_c(\mathbb{R})$ est k fois continûment différentiable et telle que $f^{(k)}\in L^1(\mathbb{R})$, alors

$$|\hat{f}(\xi)|\leq\frac{||f^{(k)}||_{L^1}}{|\xi|^k}.$$

ii) Si $f\in L^1(\mathbb{R})$ alors $\int_{\mathbb{R}}f(x)e^{iax}dx\to 0$ quand $|a|\to\infty$.
iii) Application aux coefficients de Fourier : si $f\in L^1(-\pi,\pi)$,

$$\lim_{|n|\to\infty}c_n(f)=0.$$

Remarque 8.1 Si $f\in L^2$, on sait immédiatement que $c_n(f)\to 0$ car $c_n(f)$ s'interprètent comme les coordonnées de f sur un système orthonormé.

Démonstration i) En intégrant par parties k fois l'intégrale définissant $\hat{f}$, on obtient pour $\xi \neq 0$,

$$|\hat{f}(\xi)| = |\frac{1}{(i\xi)^k} \int f^{(k)}(x) e^{-ix\xi} dx| \leq \frac{||f^{(k)}||_{L^1}}{|\xi|^k}.$$

ii) Soit f_n une suite de fonctions $\mathcal{C}^\infty$ et à support compact qui tendent vers f dans L^1 (proposition **??**). On a, pour n fixé assez grand : $||f_n - f||_1 \leq \varepsilon$, ce qui implique $|\hat{f}_n(\xi) - \hat{f}(\xi)| \leq \varepsilon$ pour tout ξ. En utilisant (i), on voit que $|\hat{f}_n(\xi)| \to 0$ quand n est fixé et $|\xi| \to \infty$. Donc $|\hat{f}_n(\xi)| \leq \varepsilon$ pour ξ assez grand. Finalement,

$$|\hat{f}(\xi)| \leq |\hat{f}(\xi) - \hat{f}_n(\xi)| + |\hat{f}_n(\xi)| \leq 2\varepsilon$$

pour ξ assez grand. ∘

La proposition suivante nous dit que la série de Fourier de f converge vers $f(x)$ en tout point x où f est suffisamment régulière.

Proposition 8.1 *principe de localisation (Principe de localisation)*
Si $f \in L^1(-\pi, \pi)$ et si la fonction $y \to \frac{f(y)-f(x)}{y-x}$ est intégrable sur un voisinage de x, alors $\lim_{N\to\infty} s_N f(x) = f(x)$, où on a noté : $s_N f(x) =: \sum_{|n|\leq N} c_n(f) e^{inx}$.

Expliquons pourquoi le résultat précédent s'appelle principe de localisation. Alors que $s_N(f)$ est le résultat d'un calcul intégral sur tout l'intervalle $[-\pi, \pi]$, et donc d'un calcul global, le comportement de $s_N f(x)$ dépend du comportement local de f au voisinage de x. Il y a donc "localisation".

Démonstration **Etape 1** On se ramène au cas $f(x) = 0, x = 0$.
Supposons la proposition démontrée pour $x = 0$, $f(x) = 0$. Soit maintenant $g \in L^1(-\pi, \pi)$ telle que $\frac{g(y)-g(x)}{y-x}$ soit intégrable au voisinage de x. Alors on pose $f(y) = g(x+y) - g(x)$. On a bien $f(0) = 0$ et $\frac{f(y)}{y} = \frac{g(x+y)-g(x)}{y}$ est intégrable au voisinage de 0. Donc, par hypothèse, $s_N f(0) \to f(0) = 0$. Mais

$$s_N f(0) = \sum_{|n|\leq N} c_n(g(x+y) - g(x)) = \sum_{|n|\leq N} \frac{1}{2\pi} \int_{-\pi}^{\pi} (g(x+y) - g(x)) e^{-iny} dy$$

$$= (\sum_{|n|\leq N} \frac{1}{2\pi} \int_{-\pi}^{\pi} g(z) e^{-in(z-x)} dz) - g(x) = (\sum_{|n|\leq N} \frac{1}{2\pi} e^{inx} \int_{-\pi}^{\pi} g(z) e^{-inz} dz) - g(x)$$

$$= s_N g(x) - g(x).$$

Donc $s_N g(x) \to g(x)$. En fait, l'argument précédent montre que s_N commute avec les translations :

$$s_N[g(. + x)] = (s_N g)(. + x).$$

Etape 2 On a

$$s_N f(0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(y) \frac{\sin(N + \frac{1}{2})y}{\sin\frac{y}{2}} dy. \tag{8.1}$$

En effet, $\sum_{-N}^{N} e^{iky} = \frac{\sin(N+\frac{1}{2})y}{\sin\frac{y}{2}}$, ce qui se prouve aisément en sommant la suite géométrique.

Etape 3 Par l'étape 1 il suffit de montrer que si $f \in L^1(-\pi,\pi)$ et si $\frac{f(y)}{y}$ est intégrable autour de 0, alors $s_N f(0) \to 0$. Comme sur $[-\pi,\pi]$, $|\sin\frac{y}{2}| \geq \frac{|y|}{\pi}$, on a

$$|\frac{f(y)}{\sin\frac{y}{2}}| \leq \frac{\pi|f(y)|}{|y|} \in L^1(-\pi,\pi).$$

Donc on peut appliquer le lemme de Riemann-Lebesgue à la fonction $\frac{f(y)}{\sin\frac{y}{2}}$. On conclut que l'intégrale de (8.1) définissant $s_N f(0)$ tend vers 0 quand N tend vers l'infini. ∘

Exercice 45 Une preuve rapide et une généralisation du principe de localisation. Soit $f \in L^1(0,2\pi)$, 2π-périodique. On note $s_{N,M} f$ la série partielle de Fourier de f, définie par

$$s_{N,M} f(x) = \Sigma_{k=-N}^{k=M} c_k(f) e^{ikx},$$

où $c_k(f) = \frac{1}{2\pi}\int_0^{2\pi} f(x)e^{-ikx}dx$. On rappelle que par le Lemme de Riemann-Lebesgue, $c_k(f) \to 0$ quand $k \to \pm\infty$. Nous allons montrer le théorème suivant, qui est une version du "principe de localisation".

Théorème 8.1 *principe de localisation!généralisation (i) Soit $f(x)$ une fonction 2π-périodique telle que*

$$\frac{f(x)}{e^{ix}-1} = g(x) \in L^1(0,2\pi).$$

Alors $s_{N,M} f(0) \to 0$ quand $N, M \to +\infty$.
(ii) Plus généralement, si $x \to \frac{f(x)-c}{x-y} \in L^1(0,2\pi)$, alors $s_{N,M} f(y) \to c$.

Remarque : si f est continue en 0, la première hypothèse entraîne $f(0) = 0$. Si f est continue en y, la deuxième hypothèse entraîne $f(y) = c$.

On appelle l'énoncé précédent le principe de localisation car il dit, en termes informels, que "si f est régulière en x, alors la série de Fourier de f tend vers $f(x)$ au point x". Bien que $s_{N,M} f$ soit définie par une formule globale (une intégrale sur l'intervalle $[0,2\pi]$), la série de Fourier reconnaît les points réguliers et son comportement dépend du comportement local de f. La démonstration qui suit est un exercice vraiment élémentaire grâce à l'astucieuse démonstration due à Ronald Coifman, de l'Université de Yale (démonstration communiquée par Yves Meyer).

1) Déduire (ii) de (i).

2) Sous l'hypothèse de (i), on appelle γ_k les coefficients de Fourier de g. Montrer que $c_k = \gamma_{k-1} - \gamma_k$. En déduire que $\sum_N^M c_k \to 0$ et conclure en appliquant le Lemme de Riemann-Lebesgue.

Corollary 1 *base!de Fourier Le système*

$$\frac{1}{(2\pi)^{\frac{1}{2}}}(e^{ikt})_{k\in\mathcal{Z}}$$

est une base hilbertienne de $L^2(-\pi,\pi)$. *Notant* $c_n(f)=\frac{1}{2\pi}\int_{-\pi}^{\pi}e^{-inx}f(x)dx$, *on a donc pour toute* f *dans* $L^2([-\pi,\pi])$,

$$f(x)=\sum_{n\in\mathbf{Z}}c_n(f)e^{inx},$$

la série précédente convergeant au sens L^2.

Démonstration On appelle polynôme trigonométrique toute expression de la forme $P(t)=\sum_{k=-N}^{N}a_ke^{ikt}$, où les a_k sont des nombres complexes. Pour montrer que le système de Fourier est une base hilbertienne, il nous suffit de montrer que c'est un système total, c'est-à-dire que les polynômes trigonométriques forment un sous-espace vectoriel dense de $L^2(-\pi,\pi)$. Mais le lemme 8.1 (Principe de localisation) nous assure que si f est (e.g.) $\mathcal{C}^2$ et 2π-périodique sur $\mathbb{R}$, alors $s_N(f)(x)\to f(x)$ en tout point (On peut aussi utiliser directement le théorème de Stone-Weierstrass). Comme de plus les coefficients de la série de Fourier de f vérifient $|c_k(f)|\leq\frac{C}{k^2}$, la série de Fourier est en fait uniformément convergente et donc converge aussi dans $L^2([-\pi,\pi])$ vers f. Or, les fonctions $\mathcal{C}^2$ et 2π-périodiques forment un sous-espace dense de $L^2([-\pi,\pi])$. En effet, par la proposition **??**, les fonctions $\mathcal{C}^\infty$ à support compact dans $[-\pi,\pi]$ sont denses dans $L^2(-\pi,\pi)$. On conclut que le système de Fourier est total, et donc une base hilbertienne. ∘

Corollary 2 *principe de localisation!(pour une fonction Hölderienne) si* $f\in L^1([-\pi,\pi])$ *est Höldérienne d'exposant* $0<\alpha\leq 1$ *en* x *(c'est-à-dire* $|f(x)-f(y)|\leq C|x-y|^\alpha$ *), alors* $s_Nf(x)\to f(x)$. *Cette conclusion s'applique si* f *est une primitive sur* $[-\pi,\pi]$ *d'une fonction de* $L^2(-\pi,\pi)$.

Démonstration L'application du principe de localisation est immédiate :
$|\frac{f(y)-f(x)}{y-x}|\leq|x-y|^{\alpha-1}$ qui est bien intégrable au voisinage de x. Soit maintenant f une fonction qui est la primitive sur $[-\pi,\pi]$ d'une fonction de $L^2(-\pi,\pi)$. En appliquant l'inégalité de Cauchy-Schwarz,

$$|f(x)-f(y)|=|\int_y^x f'(t)dt|\leq|y-x|^{\frac{1}{2}}(\int_{-\pi}^{\pi}|f'(t)|^2dt)^{\frac{1}{2}}.$$

La fonction f est donc Hölderienne d'exposant $\frac{1}{2}$ et le principe de localisation s'applique. ∘

8.1 Convolution des fonctions périodiques et séries de Fourier

La décomposition en série de Fourier d'une fonction $f\in L^2([-\pi,\pi])$ implique qu'on la considère comme une fonction $2\pi-$périodique, puisque la série de Fourier l'est. On note $L^2_{per}(\mathbb{R})$

l'ensemble des fonctions $f \in L^2_{loc}(\mathbb{R})$ qui sont 2π-périodiques. Toute fonction $f \in L^2([-\pi,\ \pi])$ définit un élément unique de $L^2([-\pi,\ \pi])$.

Définition 8.1 *convolution!périodique* **et proposition** *Si $f \in L^1([-\pi,\pi])$ et $g \in L^1([-\pi,\pi])$, on prolonge f et g en des fonctions 2π-périodiques sur $\mathbb{R}$ et on pose $f*g(x) = \int_{-\pi}^{\pi} f(y)g(x-y)dy$. La fonction $f*g$ ainsi définie appartient à $L^1(-\pi,\pi)$ et est 2π-périodique.*

Exercice 46 En reprenant l'argument du théorème **??**, montrer que si $T : L^2_{per}([-\pi,\pi]) \to C^0_{per}([-\pi,\pi])$ est linéaire, continu et commute avec les translations, alors il existe une fonction $g \in L^2([-\pi,\pi])$ telle que $Tf = g*f$, où "$*$" désigne la convolution périodique.‘

Théorème 8.2 *continuité!de la convolée de deux fonctions L^2 Si $f,g \in L^2(-\pi,\pi)$, alors $f*g$ est continue et $c_n(f*g) = 2\pi c_n(f)c_n(g)$. De plus, la série de Fourier de $f*g$ converge uniformément vers $f*g$.*

Remarquons que la relation précédente montre l'effet régularisant de la convolution : les hautes fréquences de $f*g$ sont plus faibles que celles de f, puisque $c_n(g)$ tend vers zéro.

Démonstration i) On a par l'inégalité de Cauchy-Schwarz

$$|(f*g)(x)| \leq \int |f(x-y)||g(y)|dy \leq ||f||_{L^2}||g||_{L^2}.$$

Donc $f*g$ est majorée et appartient aussi à $L^2(-\pi,\pi)$. On a, en appliquant plusieurs fois le théorème de Fubini (les intégrales se font sur $[-\pi,\pi]$ ou, indifféremment, sur n'importe quel intervalle de longueur 2π) :

$$c_n(f*g) = \frac{1}{2\pi}\int\int f(x-y)g(y)e^{-int}dydx = \frac{1}{2\pi}\int\int f(x-y)e^{-in(x-y)}g(y)e^{-iny}dydx$$

$$= \frac{1}{2\pi}(\int g(y)e^{-iny}dy)(\int f(u)e^{-inu}du) = 2\pi c_n(f)c_n(g).$$

Le terme général de la série de Fourier de $f*g$ vérifie

$$|c_n(f*g)e^{inx}| = |c_n(f)||c_n(g)| \leq |c_n(f)|^2 + |c_n(f)|^2.$$

Cette dernière série est convergente. La série de Fourier de $f*g$, $F_N(x) = \sum_{n=1}^{N} c_n(f*g)e^{inx}$, est donc uniformément convergente. Sa F limite est donc continue. Donc d'une part F_N tend vers $f*g$ dans L^2 et donc par la réciproque du théorème de Lebesgue une sous-suite tend vers cette fonction presque partout. De l'autre F_N tend vers F. On en déduit que $f*g = F$ presque partout et on en déduit que $f*g$ est égale presque partout à une fonction continue (et donc peut être appelée continue). ∘

Exercice 47 Transformée de Fourier discrète et transformée inverse.
La transformée de Fourier discrète est l'application de $L^2([-\pi,\ \pi]) \to l^2(\mathbf{Z})$ qui associe à une

fonction u la suite de ses coefficients de Fourier $c(f) = (c_k(u))_{k\in\mathbf{Z}}$. la transformée inverse est la série de Fourier associée à $c \in l^2(\mathbf{Z})$, notée

$$S(c)(x) = \sum_{k\in\mathbf{Z}} c_k e^{ikx}.$$

On a donc $S(c(f)) = f$, ce qui constitue une *formule d'inversion de Fourier.* Si $a,\ b \in l^2(\mathbf{Z})$, on note ab le produit terme à terme, défini par $(ab)_k = a_k b_k$.

1) Avec le formalisme précédent, vérifier que $S(ab) = \frac{1}{2\pi} S(a) * S(b)$.

2) Cette formule nous permet de mieux comprendre. La démonstration que nous avons donnée pour le principe de localisation. Considérons le "filtre passe-bas" $b^N \in l^2(\mathbf{Z})$ défini par $b^N_k = 1$ si $|k| \leq N$, $b^N_k = 0$ sinon. Calculer $S(b^N)$.

3) En déduire que la série de Fourier tronquée de f, $s_N f$, est obtenue par convolution 2π-périodique de f avec ce qu'on appelle le noyau de Féjer, $s_N f = h_N * f$, où

$$h_N(x) = \frac{\sin(N + \frac{1}{2})y}{\sin\frac{y}{2}}.$$

8.1.1 Autres bases de Fourier

Corollary 3 *base!de Fourier en sinus et cosinus* **Bases en sinus et en cosinus**
i) On pose pour $T > 0$ $\omega = \frac{2\pi}{T}$, c'est la fréquence de base associée à la période T. Les fonctions

$$\frac{1}{\sqrt{T}} e^{ik\omega t},\ \ k \in \mathbf{Z}$$

forment une base hilbertienne de $L^2(0,T)$. Les fonctions

$$\frac{1}{\sqrt{T}}, \sqrt{\frac{2}{T}} cos(\frac{2k\pi t}{T}), \sqrt{\frac{2}{T}} sin(\frac{2k\pi t}{T}), k = 1, 2, ...$$

forment également une base hilbertienne de $L^2(0,T)$.
ii) Il en est de même pour les fonctions base!en cosinus

$$\frac{1}{\sqrt{T}}, \sqrt{\frac{2}{T}} cos(\frac{k\pi t}{T}), k = 1, 2, ...$$

La transformée associée à la base en cosinus s'appelle la "transformée en cosinus."
Il y a également une "base en sinus",base!en sinus $\sqrt{\frac{2}{T}} sin(\frac{k\pi t}{T}), k = 1, 2, ...$

Démonstration i) La deuxième base résulte de l'application à la base de Fourier de la remarque générale suivante. Si $(e_k)_{k\in\mathbf{Z}}$ est une base hilbertienne, alors le système $f_0 = e_0$,..., $f_{2k} = \frac{e_k + e_{-k}}{\sqrt{2}}$, $f_{2k+1} = \frac{e_k - e_{-k}}{\sqrt{2}}$, ... aussi.
ii) Si $f \in L^2(0,T)$, on lui associe la fonction paire $\tilde{f}$ sur $[-T,T]$ qui coïncide avec f sur $[0,T]$. On décompose $\tilde{f}$ sur la base de Fourier de $[-T,T]$. La base de Fourier sur $[-T,T]$ est formée des fonctions $\frac{1}{\sqrt{2T}} e^{\frac{i\pi kt}{T}}$. Donc on a

$\tilde{f}(x) =_{L^2} \sum_{n\in\mathbf{Z}} \frac{1}{2T}(\int_{-T}^{T} \tilde{f}(t)e^{\frac{-i\pi kt}{T}}dt)e^{\frac{i\pi kx}{T}}$. Comme $\tilde{f}$ est paire, on voit en faisant le changement de variables $t \to -t$ dans les intégrales que les coefficients de $e^{\frac{i\pi kt}{T}}$ et $e^{\frac{-i\pi kt}{T}}$ sont égaux. On remarque aussi que $\int_{-T}^{T} \tilde{f}(t)e^{\frac{i\pi kt}{T}}dt = 2\int_0^T f(t)\cos(\frac{\pi kt}{T})dt$. Aussi,
$\tilde{f}(x) =_{L^2} \frac{1}{2T}\int_{-T}^{T} \tilde{f}(t)dt + \sum_{n\in\mathbf{N}^*} \frac{1}{2T}(\int_{-T}^{T} \tilde{f}(t)e^{\frac{-i\pi kt}{T}})(e^{\frac{i\pi kx}{T}} + e^{\frac{i\pi kx}{T}})$, et donc
$f(x) =_{L^2} \frac{1}{T}\int_0^T f(t)dt + \sum_{n\in\mathbf{N}} \frac{2}{T}(\int_0^T f(t)\cos(\frac{\pi kt}{T}))\cos(\frac{\pi kx}{T})$. Comme les fonctions $\frac{1}{\sqrt{T}}$, $\sqrt{\frac{2}{T}}\cos(\frac{\pi kx}{T})$ forment un système orthonormé de $L^2(0,T)$, l'égalité précédente exprime qu'elles forment en fait une base hilbertienne.

(iii) Si on prolonge la fonction f en une fonction impaire sur $[-T,T]$ et que l'on reprend le raisonnement précédent, on trouve la base en sinus. Cette base a la propriété, utile pour modéliser les cordes vibrantes, que ses éléments valent 0 aux extrémités de l'intervalle.

∘

Exercice 48 Détailler la preuve de (iii) en vous inspirant de la preuve de (ii).

Remarque: Le résultat ii), relatif à la transformée en cosinus, s'obtient en considérant la série de Fourier du signal pair $\tilde{f}$ obtenu par symmétrie par rapport à l'axe des y. Ceci est très important en pratique, car l'introduction de cette symmétrie, qui se généralise sans mal au cas des images, permet d'éviter la présence de discontinuités aux frontières du domaine du signal ou de l'image (supposés périodique dans le cadre de la décomposition en séries de Fourier), qui sont à l'origine d'effets de Gibbs (voir le paragraphe 8.4). Ce type de transformée en cosinus est souvent utilisé en compression des images (comme dans le standard JPEG). Un autre avantage de cette décomposition, pour la compression, est présenté ci-dessous.

8.2 Bases de Fourier en dimension 2

Les énoncés qui suivent se généralisent sans changement de démonstration à la dimension N. Nous traitons le cas $N = 2$ pour éviter des indices de sommation inutiles. On pose $x = (x_1, x_2) \in \mathbb{R}^2$, $k = (k_1, k_2) \in \mathbb{R}^2$ et on note $k.x = k_1x_1 + k_2x_2$ leur produit scalaire.

Lemme 8.2 *fonction!à variables séparées Les fonctions à variables séparées, c'est-à-dire de la forme $w(x) = u(x_1)v(x_2)$ avec $u, v \in L^2(0, 2\pi)$ forment un système total de $L^2([0, 2\pi]^2)$.*

Démonstration Les fonctions caractéristiques de rectangles sont à variables séparées et elles forment un système total de $L^2([0, 2\pi]^2)$. ∘

Lemme 8.3 *Si $u_k(x) \to u(x)$ et $v_l(x) \to v(x)$ dans $L^2(0,\ 2\pi)$, alors $u_k(x_1)v_l(x_2) \to u(x_1)v(x_2)$ dans $L^2([0,\ 2\pi]^2)$ quand $k,\ l \to +\infty$.*

Démonstration On remarque que par le théorème de Fubini,

$$||u(x_1)v(x_2)||_{L^2([0,\,2\pi]^2)} = ||u(x_1)||_{L^2([0,\,2\pi])}||v(x_2)||_{L^2([0,\,2\pi])}.$$

Donc, par l'inégalité triangulaire,

$$||u^k(x_1)v^l(x_2) - u(x_1)v(x_2)||_{L^2([0,\,2\pi]^2)} \leq ||(u^k - u)v^l||_{L^2([0,\,2\pi]^2)} + ||u(v^l - v)||_{L^2([0,\,2\pi]^2)} =$$

$$||u^k - u||_{L^2([0,\,2\pi])}||v^l||_{L^2([0,\,2\pi])} + ||u||_{L^2([0,\,2\pi])}||v^l - v||_{L^2([0,\,2\pi])}.$$

Les deux termes de droite tendent vers zéro quand $k,\ l \to +\infty$. ∘

Théorème 8.3 *série de Fourier!sur le carré Les fonctions* $e_k(x) = \frac{1}{2\pi}e^{ik.x}$, $k \in \mathbf{Z}^2$, *forment une base hilbertienne de* $L^2([0, 2\pi])$ *et on a donc pour toute fonction* $u \in L^2([0, 2\pi]^2)$,

$$u = \sum_{k\in\mathbf{Z}^2} c_k(u)e^{ik.x},\ \textit{avec}\ c_k(u) = \frac{1}{(2\pi)^2}\int_{[0,2\pi]^2} u(x)e^{-ik.x}dx, \tag{8.2}$$

la convergence de la série se vérifiant au sens de L^2.

Démonstration On vérifie facilement que e_k est un système orthonormé. Pour montrer qu'il est total, il suffit de montrer, par le lemme 8.2, que les e_k engendrent les fonctions séparables. Mais si $w(x) = u(x_1)v(x_2) \in L^2([0, 2\pi])^2$ est une telle fonction, par une application directe du théorème de Fubini, $u(x_1)$ et $v(x_2)$ sont dans $L^2(0, 2\pi)$. Les fonctions u et v sont donc sommes au sens L^2 de leurs séries de Fourier :

$$u(x_1) = \sum_{k_1\in\mathbf{Z}} c_{k_1}e^{ik_1x_1},\ \ c_{k_1} = \frac{1}{2\pi}\int_{[0,2\pi]} u(x_1)e^{-ik_1x_1};$$

$$v(x_2) = \sum_{k_2\in\mathbf{Z}} c_{k_2}e^{ik_2x_2},\ \ c_{k_2} = \frac{1}{2\pi}\int_{[0,2\pi]} v(x_1)e^{-ik_2x_2}.$$

En appliquant le lemme 8.3, on obtient une série double convergente dans $L^2([0, 2\pi]^2)$, ce qui donne (8.2) dans le cas d'une fonction séparable $w(x) = u(x_1)v(x_2)$ avec $c_k(w) = c_{k_1}(u)c_{k_2}(v)$. Il en résulte que le système $(e_k)_{k\in\mathbf{Z}^2}$ est une base hilbertienne de $L^2([0, 2\pi]^2)$ et (8.2) est donc valide. ∘

8.3 Décroissance des coefficients de Fourier et problèmes de compression du signal

On s'intéresse au comportement des coefficients de Fourier quand la 2π-périodisée de f est C^1, C^2, etc... Si f est C^p et 2π-périodique, en intégrant par parties p fois sur $[0, 2\pi]$,

$$c_n(f) = \int e^{-inx}f(x)dx = \frac{1}{(in)^p}\int e^{-inx}f^{(p)}(x)dx.$$

Donc, les coefficients décroissent d'autant plus vite que f est plus régulière.

Si maintenant f présente un saut en 0, on montre que si f est C^1 sur $[0, 2\pi]$ mais pas 2π-périodique, alors $c_n(f) = O(\frac{1}{n})$. Plus précisément, si nous notons $f(0^+)$ la valeur en 0 par la droite et $f(2\pi^-)$ la valeur en 2π par la gauche

$$c_n(f) = \frac{1}{in}\int_0^{2\pi} e^{-inx} f'(x)dx + \frac{f(0^+) - f(2\pi^-)}{in}.$$

Or on montre (par le lemme de Riemann-Lebesgue) que le premier terme est $o(\frac{1}{n})$. On sait que $\sum_{n\geq N} \frac{1}{n^2} = O(\frac{1}{n})$, et la décroissance des coefficients de Fourier de la fonction est donc très lente (1000 termes pour une précision de 10^{-3}), dès que la fonction présente une discontinuité.

En ce qui concerne les coefficients de Fourier $c_{k,l}$ d'une "image", c'est-à-dire une fonction $f(x, y)$ définie sur un carré $[0, 2\pi] \times [0, 2\pi]$, C^1, mais pas $2\pi \times 2\pi$-périodique, le résultat est identique. On montre que $c_{n,m} = O(\frac{1}{nm})$ et le reste (pour la norme L^2) de la série double est donc en $O(\frac{1}{nm})$. Donc, pour une précision de 10^{-3}, il faut encore 1000 termes.

Une bonne alternative lorsque la fonction présente une discontinuité du type précédent consiste à utiliser la tranformée en cosinus: $c_n(f) = \frac{1}{\pi}\int_0^{2\pi} \cos nx f(x)dx$. On a, en intégrant par parties et en remarquant que $\sin nx$ s'annule en 0 et 2π,

$$c_n(f) = \frac{1}{i\pi n}\int_0^{2\pi} \sin nx f'(x)dx.$$

Puis on montre que $c_n(f) = o(\frac{1}{n})$ par le lemme de Riemann-Lebesgue. Les coefficients de Fourier "en cosinus" décroissent donc plus vite qu'avec la transformée de Fourier classique et on peut donc en transmettre moins pour une qualité d'image égale. Pour transmettre une image, on la découpe en petits carrés et on transmet une partie des coefficients de Fourier de chaque imagette (principe utilisée par le standard JPEG). On augmente ainsi la probabilité qu'une imagette présente une couleur homogène et soit donc régulière. L'utilisation de la tranformée en cosinus permet donc de comprimer l'information dans les sous-carrés de l'image où celle-ci est régulière. Par contre, les calculs précédents prouvent qu'on ne gagne rien quand un "bord" est présent dans l'imagette. En effet, (on pourra expliciter le calcul pour une image blanche au dessus de la diagonale et noire en dessous), un calcul du même type que ci-dessus implique que les coefficients décroissent en $O(\frac{1}{nm})$. C'est ce qui explique les phénomènes de "halo" autour des objets sur un fond contrasté : le petit nombre de coefficients transmis ne suffit pas à approcher bien l'imagette. Nous verrons au chapitre 8.4 qu'il y a une autre raison à ceci: le phénomène de Gibbs (voir la figure 8.2). Le long des discontinuités de l'image, apparaissent toujours des oscillations résiduelles, quel que soit le nombre de coefficients transmis.

En conclusion, la transformée en cosinus, s'affranchissant des discontinuités aux frontières du domaine de l'image, présente un double avantage sur la transformée de Fourier. En termes d'économie de la représentation, elle tire mieux partie de l'eventuelle régularité de la fonction à l'intérieur de son domaine (régularité souvent élevée dans le cas d'imagettes). De plus, elle évite l'apparition d'oscillations résiduelles le long de ces frontières.

8.4 Phénomène de Gibbs

phénomène de Gibbs La représentation d'un signal par sa série de Fourier conduit à l'apparition d'oscillations résiduelles, dont l'amplitude ne dépend pas du nombre de coefficients utilisés

pour représenter la fonction. Ce résultat mathématique sur l'approximation d'un signal par les sommes partielles de sa série de Fourier porte le nom de phénomène de Gibbs. Ce phénomène est observé à la sortie de tout système physique ou numérique mesurant ou calculant une fonction f. Si la fonction $f(t)$ (t désignant par exemple le temps) "saute" brusquement d'une valeur à une autre, alors l'expérimentateur observe une série d'oscillations avant et après le saut. Il se gardera bien de les interpréter comme faisant partie du signal. En effet, le phénomène est dû au fait que les appareils de mesure (et les programmes numériques sur ordinateur) "tronquent" nécessairement les hautes fréquences. Cela veut aussi dire que l'on n'observe jamais les fonctions elles mêmes, mais des sommes partielles de leur série de Fourier. Et on observe donc aussi les "parasites" dûs à cette troncature en fréquence ; en particulier, le phénomène de Gibbs. Du point de vue mathématique, on peut énoncer le phénomène comme suit :
" Si une fonction f, par ailleurs régu'lière, présente un saut en un point, alors les sommes partielles $s_N f$ de sa série de Fourier accentuent ce saut en le multipliant par un facteur qui ne dépend pas de N."

On commence par donner le résultat précis dans un cas simple: on considère la fonction "en dents de scie" $s(x)$, 2π-périodique et telle que $s(x) = \frac{\pi - x}{2}$ sur $[0, 2\pi[$. Le calcul des coefficients de Fourier de s et le corollaire 1 montrent que $s(x)$"="$\sum_{k=1}^{\infty} \frac{\sin(kx)}{k}$ au sens de la convergence L^2, ainsi qu'en tout point de l'intervalle ouvert $]0, 2\pi[$, d'après la proposition 8.1. On considère les sommes partielles de cette série de Fourier, $s_n(x) =: \sum_{k=1}^{n} \frac{\sin(kx)}{k}$.

Proposition 8.2 (Phénomène de Gibbs)*: phénomène de Gibbs*

$$\limsup_{n\to\infty, x\to 0^+} s_n(x) = (1+c)s(0^+); \qquad \liminf_{n\to\infty, x\to 0^+} s_n(x) = (1-c')s(0^+). \tag{8.3}$$

Démonstration On va étudier la suite $s_n(\frac{\pi}{n})$ quand $n \to \infty$. On commence par étudier les variations de $G(a) =: \int_0^a \frac{\sin(t)}{t} dt$ pour en déduire que $G(\pi) > G(+\infty)$. La fonction $G(a)$ est croissante sur les intervalles pairs $[2k\pi, (2k+1)\pi]$ et décroissante sur les intervalles impairs. On voit aisément que $|G((n+1)\pi) - G(n\pi)|$ est une suite décroissante. Il en résulte que la suite $G(2n\pi)$ est une suite croissante strictement, la suite $G((2n+1)\pi)$ une suite strictement décroissante, et les deux convergent vers une valeur commune notée $G(+\infty)$. On a donc $G(\pi) > G(+\infty)$. On sait par ailleurs que $G(+\infty) = \frac{\pi}{2}$. Revenons à la suite $s_n(\frac{\pi}{n})$. On a

$$s_n(\frac{\pi}{n}) = \sum_{k=1}^{n} \frac{\sin(\frac{k\pi}{n})}{k} = \frac{\pi}{n} \sum_{k=1}^{n} \frac{\sin(\frac{k\pi}{n})}{\frac{k\pi}{n}} \to_{n\to+\infty} \int_0^{\pi} \frac{\sin u}{u} du.$$

La dernière limite vient du fait que l'on reconnaît la somme de Riemann associée à l'intégrale. Mais

$$s_n(\frac{\pi}{n}) \to G(\pi) > G(+\infty) = \frac{\pi}{2} = s(0^+),$$

car $s(0^+) = \frac{\pi}{2} = \int_0^{+\infty} \frac{\sin u}{u} du$. Donc pour tout n, il y a une valeur très proche de 0, en l'occurrence $\frac{\pi}{n}$, telle que la somme partielle de la série de Fourier dépasse d'un facteur constant $\frac{G(\pi)}{G(+\infty)}$ la valeur de la limite $s(0^+)$. Pour raisons de symétrie, la même chose se produit en 0^- avec la suite $s_n(-\frac{\pi}{n})$. Nous avons donc montré l'existence des lim sup et lim inf de l'équation (8.3). ∘

Exercice 49 On peut préciser un peu plus le résultat précédent en donnant le comportement asymptotique de $s_n(x)$ au voisinage de 0, ce qui permet de tracer les oscillations de s_n au voisinage de la discontinuité. Montrer que pour $|x| \leq 1$ et uniformément en x,

$$s_n(x) = \int_0^x \frac{\sin(nt)}{t} dt - \frac{x}{2} + O(x, \frac{1}{n}).$$

Numériquement, les constantes positives c et c' sont de l'ordre de $0,18$. Plus précisément, la somme partielle s_n de la série de Fourier de f présente des oscillations, maximales aux points $\frac{k\pi}{n}$. Les oscillations de cette approximation ont donc une fréquence de plus en plus élevée avec l'ordre d'approximation n, mais l'erreur reste proportionnelle au saut de la fonction f. Ce résultat se généralise au cas d'une fonction C^1 sur $[0, 2\pi]$, mais pas 2π périodique. Pour ce faire, on soustrait à la fonction f une fonction en "dents de scie" $\lambda s + \mu = \tilde{s}$, où λ et μ ont été choisis de manière à la rendre Lipschitzienne et on applique à la différence $f - \tilde{s}$ le principe de localisation. Il y a donc convergence uniforme de la série de Fourier de $f - \tilde{s}$ vers $f - \tilde{s}$, alors que la série de Fourier de $\tilde{s}$ présente le phénomène de Gibbs. Le développement de Fourier de f présente donc aussi le phénomène de Gibbs.

Nous illustrons, à la figure 8.1, le phénomène dans le cas de la fonction 2π-périodique, impaire, et valant 1 sur l'intervalle $]0, \pi]$. Nous montrons les sommes partielles de sa série de Fourier. Remarquons en particulier le fait que l'erreur maximum ne varie pas avec le nombre de coefficients de l'approximation. En revanche, la fréquence de ces oscillations augmente avec l'ordre d'approximation. Nous présentons ensuite une illustration du phénomène de Gibbs dans le cas des images numériques: partant d'une image, nous calculons sa série de Fourier (en fait une approximation finie de cette série présentée au paragraphe suivant: la transformée de Fourier discrète), mettons les hautes fréquences à zéro, puis calculons l'image dont la série de Fourier est celle ainsi obtenue (anticipant sur les définitions et notations du paragraphe suivant sur la transformée de Fourier discrète, nous multiplions l'image $\tilde{u}_{mn}$ par la fonction indicatrice d'un carré centré sur $\tilde{u}_{0,0}$, puis appliquons la TFD inverse). Nous montrons le résultat figure 8.2, où l'image originale est placée à gauche. Le résultat, image obtenue après troncature des hautes fréquences, à droite, présente de très nombreuses oscillations.

Ce phénomène apparaît également lorsque le spectre est utilisé à des fins de manipulation d'image, comme nous le verrons au chapitre suivant.

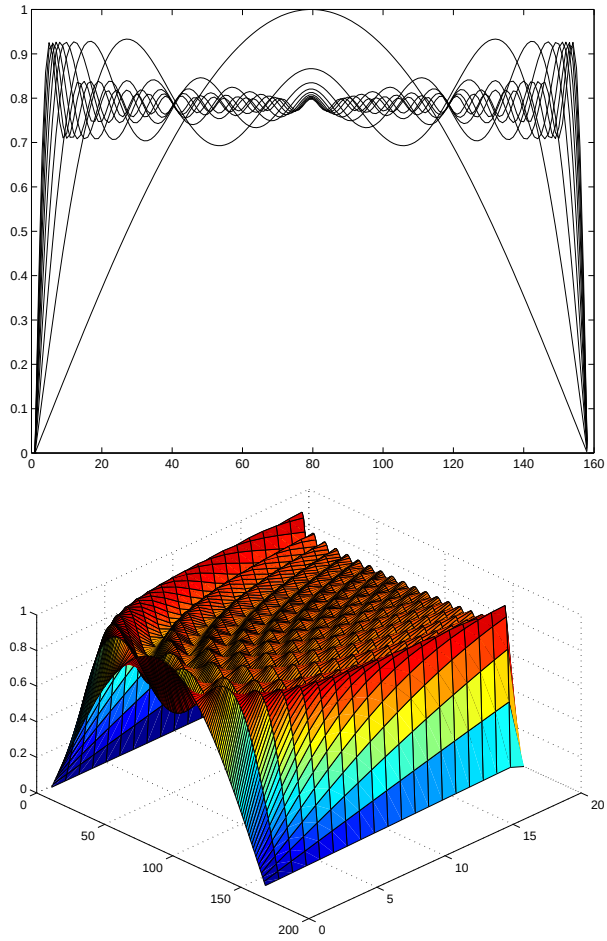


Figure 8.1: Sommes partielles de la série de Fourier de la fonction 2π-périodique, impaire, valant 1 sur $]0, \pi]$. Haut: les approximations sont représentées sur le même graphe, sur l'intervalle $]0, \pi]$. Bas: les différentes approximations sont traçées selon un troisième axe (nombre de termes entre 1 et 20). On remarque que l'erreur maximale d'approximation ne varie pas avec le nombre de termes, tandis que la fréquence des oscillations augmente.

Figure 8.2: Illustration de l'effet de Gibbs. Gauche: l'image originale; droite: l'image après que l'on ait tronqué ses hautes fréquences, et sur laquelle sont visibles de nombreuses oscillations. L'image de droite est obtenue en ne conservant que les fréquences dont le module est inférieur au quart de la fréquence maximale. Le phénomène est particulièrement visible le long des frontières du domaine de l'image (voir en particulier le côté droit) et le long des discontinuités de l'image. Remarquons que l'image est également devenue floue par suppression des hautes fréquences.

Chapter 9

Le cas discret (Révision)

9.1 Transformée de Fourier Discrète, applications

9.1.1 La dimension 1

La transformée de Fourier discrète est un moyen de calculer les coefficients de Fourier d'un fonction a-périodique u, directement à partir de ses N échantillons $u(\frac{ka}{N})$, $k = 0, ...N-1$. Cela n'est possible exactement que si la fonction présente un nombre de fréquences inférieur à N.

Pour des raisons de simplicité des notations, nous supposerons dans ce paragraphe que N est pair. Tous les résultats énoncés (sauf ceux du paragraphe 9.1.4 relatifs à la transformée de Fourier rapide) s'adaptent sans difficulté au cas N impair. En pratique, N est en fait toujours une puissance de 2.

Soit $u(x)$ une fonction réelle ou complexe de période a, et N un entier pair. On cherche un polynôme trigonométrique de la forme

$$P(x) = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{u}_n \exp\left(\frac{2i\pi nx}{a}\right), \tag{9.1}$$

qui soit égal à u aux points $\frac{ka}{N}$ pour $k = 0, ..., N-1$. On dira dans la suite que P est de degré $\frac{N}{2}$. Le but est donc d'interpoler les échantillons $u(\frac{ka}{N}) = u_k$.

Pourquoi choisir un polynôme trigonométrique ? La raison est physique : tous les dispositifs d'acquisition de signaux (sons) ou images ont une *bande passante*, c'est-à-dire un intervalle de fréquences captées par le dispositif d'enregistrement ; les autres fréquences sont perdues ou tellement atténuées qu'on les néglige : on suppose donc que la "bande passante" est $[-\frac{N}{2}, \frac{N}{2} - 1]$. Il n'y a par contre aucune raison de supposer que le signal ou image soit périodique et d'une période qui coïncide avec la fenêtre d'observation $[0, a]$ comme c'est le cas pour P. Cette hypothèse est donc imposée à la donnée et provoque une distorsion qu'on a évaluée : le phénomène de Gibbs. Si en fait la fonction u dont on possède les N échantillons n'a pas une bande de fréquence dans $-\frac{N}{2}, \frac{N}{2} - 1$, son interpolation par un polynôme trigonométrique de degré $\frac{N}{2}$ provoque une autre distorsion que nous allons évaluer précisément : l'aliasage.
On va commencer par calculer les coefficients de P.

Exercice 50 On pose $\omega_N = \exp\left(\frac{2i\pi}{N}\right)$, racine N-ième de l'unité. Montrer que $\sum_{k=0}^{N-1} \omega_N^k = 0$,

puis que $\sum_{k=0}^{N-1} \omega_N^{kl} = 0$ pour $l \neq 0$ et finalement que pour tout k_0, $\sum_{k=k_0}^{k_0+N-1} \omega_N^{kl} = 0$ pour tout $l \neq 0$.

Définition 9.1 *transformée de Fourier!discrète On pose $u_k = u(\frac{ka}{N})$ et, pour $n = -\frac{N}{2}, ...\frac{N}{2} - 1$,*

$$\tilde{u}_n = \frac{1}{N} \sum_{l=0}^{N-1} u_l \omega_N^{-nl}. \tag{9.2}$$

Les N coefficients $\tilde{u}_n$ sont appelés transformée de Fourier discrète *(TFD) des N échantillons u_k. On appelle transformée de Fourier discrète inverse l'application de C^{N} dans lui même définie par*

$$u_k = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{u}_n \omega_N^{kn},\ k = 0, ..., N-1. \tag{9.3}$$

Proposition 9.1 *transformée de Fourier!discrète Les coefficients $(\tilde{u}_n)$ définis par (9.2) sont les uniques coefficients tels que le polynôme trigonométrique (9.1) vérifie $P\left(\frac{ka}{N}\right) = u_k$, pour tout $k = 0, ..., N-1$. En d'autres termes, la transformée de Fourier discrète composée avec son inverse donne bien l'identité.*

Démonstration Pour $k = 0, ..., N-1$,

$$\begin{aligned} P(\frac{ka}{N}) &= \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{u}_n \omega_N^{nk} \\ &= \frac{1}{N} \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} (\sum_{l=0}^{N-1} u_l \omega_N^{-nl}) \omega_N^{nk} \\ &= \frac{1}{N} \sum_{l=0}^{N-1} u_l (\sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} \omega_N^{nk-nl}) \\ &= \frac{1}{N} \sum_{l=0}^{N-1} N\delta(k-l) u_l = u_k, \end{aligned}$$

où on a noté δ la fonction définie sur les entiers, valant 1 en 0, et 0 ailleurs. L'unicité provient du fait que toute application linéaire surjective de $\mathbf{C}^N$ dans lui-même est aussi injective. □

Corollary 4 *polynôme trigonométrique Si u est un polynôme trigonométrique $u(x) = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{u}_n \exp\left(\frac{2i\pi nx}{a}\right)$, les coefficients $\tilde{u}_n$ sont obtenus par la formule (9.2). Ce sont les coefficients de Fourier de u.*

Exercice 51 On note u un vecteur de $\mathbb{C}^{\mathrm{N}}$ et $TFD(u) = \tilde{u}$ sa transformée de Fourier discrète. Vérifier que $\sqrt{N}TFD$ est unitaire et que l'on a $TFD^{-1} = N.\overline{TFD}$.

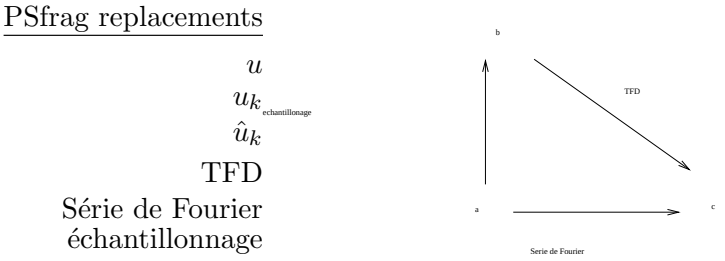


Figure 9.1: La TFD après échantillonage calcule bien les coefficients de Fourier **si la fonction** u **est un polynôme trigonométrique** (corollaire 4)

On rappelle d'autre part que si $u \in L^2(0,a)$, les coefficients de la série de Fourier de u sont définis, pour $n \in \mathbb{Z}$, par

$$c_n(u) = \frac{1}{a}\int_0^a u(x)\exp\left(\frac{-2i\pi nx}{a}\right). \tag{9.4}$$

Les coefficients $\tilde{u}_n$ de la transformée de Fourier discrète sont approchés par les termes de la TFD de (u_k) au sens suivant:

Proposition 9.2 *Soit u continue et a-périodique. Alors les $\tilde{u}_n$ sont des approximations des $c_n(u)$ par la formule des trapèzes, pour $n = \frac{-N}{2}, ..., \frac{N}{2} - 1$.*

Démonstration Il suffit d'écrire l'approximation de l'intégrale (9.4) par la méthode des trapèzes en tenant compte du fait que $u(a) = u(0)$ pour une fonction a-périodique. □

Proposition 9.3 *On suppose que les échantillons u_k sont réels. Alors $\tilde{u}_0$ et $\tilde{u}_{-\frac{N}{2}}$ sont réels, et pour $k = 1...\frac{N}{2} - 1$, $\tilde{u}_k = \overline{\tilde{u}_{-k}}$.*

Démonstration $\tilde{u}_0 = \frac{1}{N}\sum_k u_k$, et $\tilde{u}_{-\frac{N}{2}} = \frac{1}{N}\sum(-1)^k u_k$; ces deux coefficients sont donc réels. D'autre part

$$\tilde{u}_{-n} = \frac{1}{N}\sum_{k=0}^{N-1} u_k \omega_N^{kn} = \frac{1}{N}\sum_{k=0}^{N-1} u_k \overline{\omega_N^{-nk}} = \overline{\tilde{u}_n}.$$

□

Remarquons le rôle particulier joué par le terme $\tilde{u}_{-\frac{N}{2}}$, qui n'a pas de terme conjugué lui correspondant.

Proposition 9.4 *si u est un polynôme trigonométrique réel dont les fréquences sont parmi $-\frac{N}{2}, ..., \frac{N}{2} - 1$, le terme $\tilde{u}_{-\frac{N}{2}}$ est nul.*

Démonstration En effet, en regroupant les termes conjugués, on a, pour le polynôme trigonométrique P dont les coefficients sont les $\tilde{u}_n$:

$$P(x) = \tilde{u}_0 + \sum_{n=1}^{\frac{N}{2}-1}(\tilde{u}_n e^{\frac{2in\pi x}{a}} + \tilde{u}_{-n}e^{\frac{-2in\pi x}{a}}) + \tilde{u}_{-\frac{N}{2}}e^{\frac{-iN\pi x}{a}}.$$

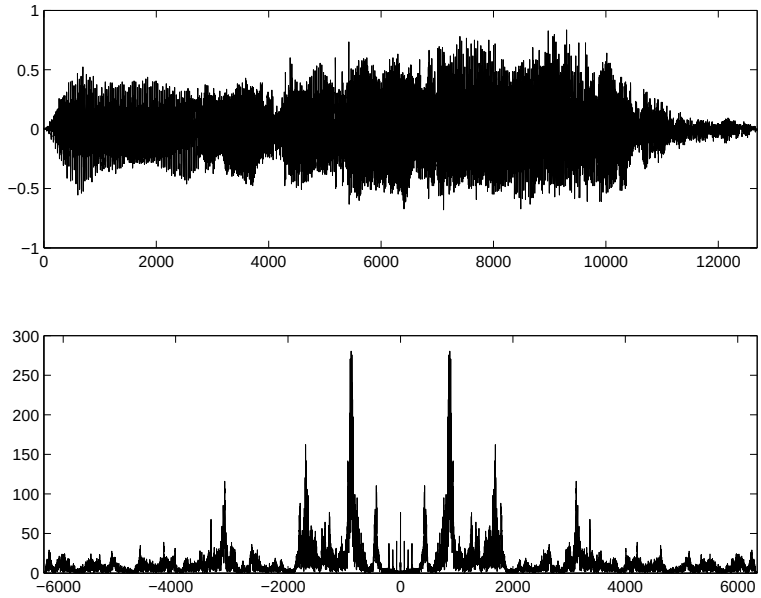


Figure 9.2: Haut: un signal correspondant à la voyelle "Ah" (le signal représente la pression de l'air en fonction du temps); bas: module de la TFD (coefficients $|\tilde{u}|$, voir le texte). On remarque que le module du spectre est symétrique, et qu'il existe trois pics importants correspondant aux fréquences dominantes.

Tous les termes de la somme sont réels sauf le dernier, qui ne l'est que si $\tilde{u}_{-\frac{N}{2}} = 0$. ∘

La Figure 9.2 montre un exemple de signal (représentant le son A) et le module de sa TFD.

9.1.2 La dimension 2

On considère un réel a, une fonction u de $\mathbb{R}^2$ dans $\mathbb{R}$, telle que $u(x+a, y+a) = u(x, y)$. On fixe à nouveau un entier N, et l'on pose $u_{k,l} = u\left(\frac{ka}{N}, \frac{la}{N}\right)$. On définit la TFD des $u_{k,l}$ comme la suite des coefficients, pour $m, n \in \{-\frac{N}{2}, ..., \frac{N}{2} - 1\}$,

$$\tilde{u}_{m,n} = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} u_{k,l} \omega_N^{-mk} \omega_N^{-nl}. \tag{9.5}$$

Exercice 52 Montrer que la transformation ainsi définie est séparable, et que le passage des $u_{k,l}$ aux $\tilde{u}_{m,n}$ s'effectue par deux TFDs à une dimension successives.

De même qu'en dimension 1, nous avons la propriété d'interpolation suivante:

Proposition 9.5 *Soient les coefficients $\tilde{u}_{m,n}$ définis, pour $m, n = -\frac{N}{2}$, ..., $\frac{N}{2} - 1$, par (9.5). Considérons le polynôme trigonométrique*

$$P(x, y) = \sum_{m,n=-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{u}_{m,n} \exp(\frac{2i\pi mx}{a}) \exp(\frac{2i\pi ny}{a}).$$

Les coefficients $\tilde{u}_{m,n}$ sont les seuls nombres complexes tels que, pour tout $k,l \in \{0,...,N-1\}$, $P\left(\frac{ka}{N},\frac{la}{N}\right) = u\left(\frac{ka}{N},\frac{la}{N}\right)$. Par conséquent, la transformée discrète inverse de $u_{k,l} \to \tilde{u}_{m,n}$ est donnée par le calcul du polynôme aux échantillons $(\frac{ka}{N},\frac{la}{N})$, $0 \le k,l \le N-1$:

$$u(k,l) = P(\frac{ka}{N},\frac{la}{N}) = \sum_{-\frac{N}{2}}^{\frac{N}{2}-1}\sum_{-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{u}_{m,n}\omega_N^{km+ln}.$$

Exercice 53 Montrer la proposition précédente. Le calcul est exactement le même qu'en dimension 1. De même qu'en dimension 1, nous pouvons identifier un certain nombre de symétries des $\tilde{u}_{m,n}$ si l'image est à valeurs réelles. On suppose à nouveau que N est pair. Montrer également la proposition suivante.

Proposition 9.6 *Supposons que les échantillons $u_{k,l}$ soient réels. Alors les coefficients $\tilde{u}_{0,0}$, $\tilde{u}_{0,-\frac{N}{2}}$, $\tilde{u}_{-\frac{N}{2},0}$, et $\tilde{u}_{-\frac{N}{2},-\frac{N}{2}}$ sont réels; de plus*

$$\forall m,n \in \{-\frac{N}{2}+1,...\frac{N}{2}-1\} \qquad \tilde{u}_{m,n} = \overline{\tilde{u}_{-m,-n}}$$

Exercice 54 A nouveau, comme en dimension 1, les coefficients $(\tilde{u}_{m,n})$ correspondent aux fréquences de l'image u, ordonnées des négatives aux positives. Plus précisément, si $u \in L^1$ et que l'on définit les coefficients de la série de Fourier de u par

$$c_{m,n} = \frac{1}{4\pi^2}\int_{[0,\,2\pi]^2} u(x,y)\exp\left(\frac{-2i\pi mx}{a}\right)\exp\left(\frac{-2i\pi ny}{a}\right),$$

alors, pour $m,n = -\frac{N}{2},...,\ \frac{N}{2}-1$ les $\tilde{u}_{m,n}$ sont des approximations des $c_{m,n}$ par la méthode des trapèzes.

La figure 9.3 présente une image et le logarithme du module de sa transformée de Fourier discrète (le logarithme est utilisé car le module des TFD des images usuelles décroit très vite lorsque l'on s'éloigne des basses fréquences).

9.1.3 Le phénomène du repliement de spectre ou aliasage

repliement de spectre

Le but de ce paragraphe est de calculer les perturbations auxquelles est exposée la transformée de Fourier discrète d'un signal lorsque celui-ci est sous-échantillonné. On vient de voir que la transformée de Fourier discrète calculait exactement les coefficients de Fourier d'un polynôme trigonométrique de degré $\frac{N}{2}$, $P(x) = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1}\tilde{u}_n\exp\left(\frac{2i\pi nx}{a}\right)$, dont on connaissait N échantillons $u(\frac{ka}{N})$, $N = 0,...,N-1$. Dans cette section, on considère une fonction $u \in L^2(0,a)$ et sa série de Fourier

$$u(x) = \sum_{n\in\mathbf{Z}} c_n(u)e^{\frac{2in\pi x}{a}}.$$

Dans toute la suite, on supposera que $\sum_{n\in\mathbf{Z}}|c_n(u)| < +\infty$, ce qui implique que u est continue et a-périodique. Cette hypothèse n'est pas irréaliste. En effet, étant donné un signal v

Figure 9.3: Gauche: une image numérique de taille 256×256; droite: le logarithme du module de sa TFD. Le spectre décroît rapidement aux hautes fréquences (rappelons que l'image étant bornée, son spectre est dans L^2). En pratique, la grande vitesse de décroissance du spectre rend nécessaire l'utilisation du logarithme pour la visualisation. La symétrie centrale du module de la TFD est visible. Les lignes horizontales et verticales correspondent aux bords verticaux et horizontaux respectivement. Remarquer également les lignes obliques qui correspondent aux bords obliques de l'image (voir en particulier les dalles sur le sol). Les droites horizontales et verticales sont également dues aux fortes discontinuités présentes aux frontières du domaine de l'image (rappelons que celle-ci est périodisée pour le calcul de son spectre).

régulier (C^2 par exemple) sur $[0, a/2]$, on peut le rendre pair en posant $u(-x) = \tilde{v}(x)$ pour $x \in [-a/2, 0]$, $u(x) = v(x)$ sur $[0, a]$. On voit que la a-périodisée de cette extension u reste Lipschitz et C^2 par morceaux et on peut en déduire (exercice !) que la série des coefficients de Fourier de u est convergente. On suppose également, ce qui est réaliste, qu'un signal u n'est en fin de compte connu que par ses échantillons sur $[0, a]$, $u(0), ..., u(\frac{N-1}{N}a)$.

Théorème 9.1 *Soit u définie sur $[0, a]$, vérifiant $\sum_n |c_n(u)| < +\infty$. Shannon !théorème discret de Alors la transformée de Fourier discrète de u est la N-périodisée de la suite des coefficients de Fourier de u :*

$$\tilde{u}_n = \sum_{q=-\infty}^{+\infty} c_{n+qN}(u), \;\; n = -\frac{N}{2}, ..., \frac{N}{2} - 1. \tag{9.6}$$

Démonstration On rappelle la notation $\omega_N = e^{\frac{2i\pi}{N}}$ et $(\omega_N)^N = 1$. Comme

$$u(x) = \sum_{m \in \mathbf{Z}} c_m(u) e^{\frac{2im\pi x}{a}},$$

on a

$$u(\frac{ka}{N}) = \sum_{m \in \mathbf{Z}} c_m(u) \omega_N^{mk}.$$

On pose pour $m \in \mathbf{Z}$, $m = qN + n$, $-\frac{N}{2} \leq n \leq \frac{N}{2} - 1$. En regroupant les termes de la série de Fourier on obtient

$$u(\frac{ka}{N}) = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} \left(\sum_{q=-\infty}^{+\infty} c_{n+qN}(u) \right) \omega_N^{nk}, \; k = 0, ..., N-1.$$

Mais on a aussi (formule d'inversion de la transformée de Fourier discrète):

$$u(\frac{ka}{N}) = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{u}_n \omega_N^{nk}, \; k = 0, ..., N-1.$$

Ces deux dernières formules définissent toutes deux la transformée de Fourier discrète et par identification on obtient la formule de "repliement de spectre" (9.6).

∘

Ce théorème va nous permettre d'interpréter les effets de moiré visibles dans beaucoup d'images digitales ou de films digitalisés (DVD). Ces effets de moiré sont dûs à un "repliement de spectre", ou "aliasage". Le repliement de spectre provient d'un sous-échantillonnage abusif. Le terme aliasage se réfère à la présence des coefficients parasites c_{n+qN}, pour $q \neq 0$ dans le calcul du coefficient de la fréquence n, $\tilde{u}_n$. Quand la transformée de Fourier discrète fait correctement son travail, qui est de retrouver le coefficient c_n de la fréquence n de u, on doit avoir $\tilde{u}_n = c_n$. Les coefficients c_{n+qN} qui s'y ajoutent dans (9.6) sont des répliques, ou "alias" de coefficients correspondant aux fréquences plus grandes $n + qN$, $q \neq 0$. D'où le terme d'aliasage.

Définition 9.2 *sous-échantillonnage d'un signal Soit un signal échantillonné* (u_k), $k = 0, ..., N-1$, *et soit* p *un entier divisant* N. *On définit l'opérateur "sous-échantillonnage d'ordre* p*" comme suit:*

$$S_p : \mathbb{R}^N \longrightarrow \mathbb{R}^{N/p}$$

$$(u_k)_{k=0,\ ...,\ N-1} \longrightarrow (v_k) = (u_{kp})_{k=0,...,N/p}.$$

Le signal (v_k) *est dit sous-échantillonné d'un facteur* p.

Nous commençons par le cas, technologiquement classique, où $p = 2$.

Corollary 5 *Soit* $(v_k) = S_2((u_k))$ *(on suppose que* $\frac{N}{2}$ *est pair). Alors* $(\tilde{v}_n)$, *la transformée de Fourier Discrète de* (v_k), *s'écrit, pour* $n = -\frac{N}{4}, ..., \frac{N}{4} - 1$,

$$\tilde{v}_n = \tilde{u}_n + \tilde{u}_{n-\frac{N}{2}} + \tilde{u}_{n+\frac{N}{2}}, \tag{9.7}$$

le deuxième terme étant par ailleurs nul si $n < 0$ *et le troisième étant nul si* $n \geq 0$.

Démonstration Appliquons le théorème 9.1 à l'unique polynôme trigonométrique P à N coefficients qui a pour échantillons les u_k. Alors par définition de la transformée de Fourier discrète, $\tilde{u}_n = c_n(P)$. On a donc pour $\frac{N}{4} \leq n \leq \frac{N}{4} - 1$,

$$\tilde{v}_n = \sum_{q \in \mathbf{Z}} c_{n+q\frac{N}{2}}(P) = \tilde{u}_n + \tilde{u}_{n-\frac{N}{2}} + \tilde{u}_{n+\frac{N}{2}}.$$

Remarquons que si $n \geq 0$ cela donne $\tilde{v}_n = \tilde{u}_n + \tilde{u}_{n-\frac{N}{2}}$, l'autre coefficient étant nul. De même, si $n < 0$, on obtient $\tilde{v}_n = \tilde{u}_n + \tilde{u}_{n+\frac{N}{2}}$.

∘

Cette proposition indique que le spectre du signal sous-échantillonné d'un facteur deux s'obtient en superposant à lui-même le spectre du signal original avec un décalage de $\frac{N}{2}$. On dit qu'il y a repliement de spectre. Ainsi, le spectre du signal sous-échantillonné contient généralement des informations non présentes dans le spectre du signal de départ, ce qui se traduit sur le signal sous-échantillonné par l'apparition de structures périodiques n'ayant pas de lien direct avec le contenu du signal. Ceci est particulièrement frappant dans le cas des signaux bi-dimensionnels, pour lesquels on a un résultat identique à celui de la proposition 5. Nous montrons deux exemples d'images sous-échantillonnées aux figures 9.1.3 (image synthétique) et 9.1.3, exemple où l'apparition de structures périodiques est dûe à la superposition, lors du sous-échantillonnage, des hautes fréquences de l'image. La manipulation numérique à faire pour créer des effets de moiré dans une image est aussi simple que son interprétation est subtile : il suffit de prendre "un point sur deux" de l'image. L'interprétation de l'opération se fait en Fourier : on a créé de basses fréquences parasites en c_n qui correspondent au "repliement" de hautes fréquences $c_{n+\frac{N}{2}}$. D'où l'apparition de sinusoïdes qui n'ont rien à voir avec le signal original et qui créent des effets de moiré.

Le résultat de la proposition 5 se généralise dans le cas d'un sous-échantillonnage d'ordre plus élevé, comme le montre la proposition suivante:

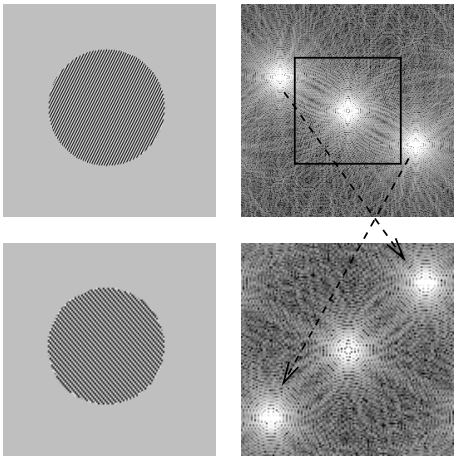


Figure 9.4: Exemple de repliement avec une image synthétique. En haut à gauche: image originale, à droite son spectre. En bas à gauche: l'image sous-échantillonnée d'un facteur deux dans chaque direction, à droite le spectre correspondant. Le spectre de l'image sous-échantillonnée est obtenu en périodisant le spectre de l'image originale avec pour période le carré visible en surimpression.

Proposition 9.7 *Soit $(v_k) = S_p((u_k))$ (on suppose que $N = pM$, pour un certain entier M). Alors $(\tilde{v}_k)$, la transformée de Fourier discrète de (v_k), s'écrit, pour $k = 1...M-1$,*

$$\tilde{v}_k = \sum_{a=-p+1}^{p-1} \tilde{u}_{k+\frac{aN}{p}}. \tag{9.8}$$

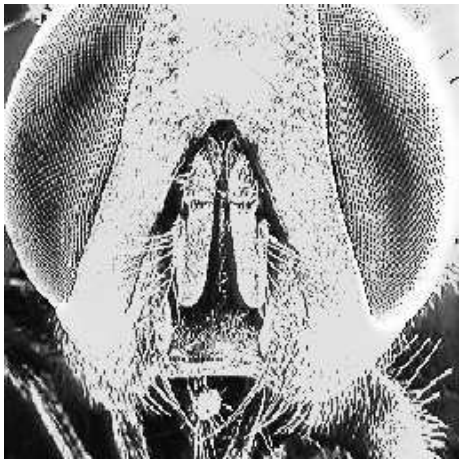
Démonstration Appliquer de nouveau le théorème 9.1 à l'unique polynôme trigonométrique à N coefficients qui a pour échantillons les u_k. Ce polynôme vérifie $c_n(P) = \tilde{u}_n$. ∘

On peut comparer les propositions 5 et 9.7 au théorème 9.1. Le théorème 9.1 nous donne notamment les conditions générales de Shannon et Whittaker pour qu'un signal soit correctement échantillonné : ces conditions sont que le spectre soit borné (nombre fini N de coefficients de Fourier) et que l'on dispose d'au moins N échantillons. Les propositions 5 et 9.7 sont plus pratiques : elles ne donnent aucune hypothèse sur le signal qui a été échantillonné et ont l'avantage de s'appliquer à un signal discret, quelconque, qu'il soit ou non issu d'un bon échantillonnage.

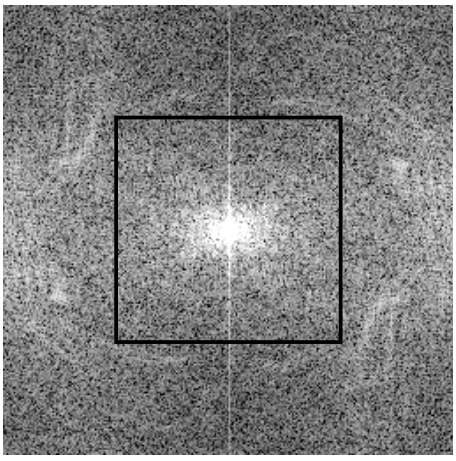
9.1.4 La transformée de Fourier rapide

transformée de Fourier!rapide

Comme nous l'avons vu plus haut, le calcul des coefficients de Fourier $\tilde{u}_n$ revient à l'évaluation d'un certain polynôme aux racines N-ièmes de l'unité. Dans le cas général, l'évaluation classique (ex. méthode de Hörner) d'un polynôme de degré $N-1$ en un point prend $\mathcal{O}(N)$ opérations. Donc si l'on répète cela pour les N racines de l'unité on devra effectuer $\mathcal{O}(N^2)$ opérations. L'algorithme de la Transformée de Fourier Rapide (TFR) permet

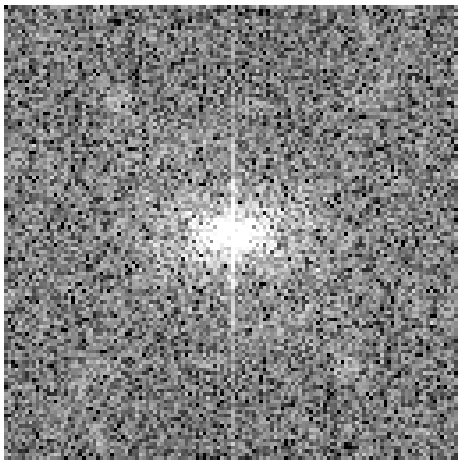


(a) Image originale



(b) Sa TFD, non nulle en dehors du carré visible en surimpression

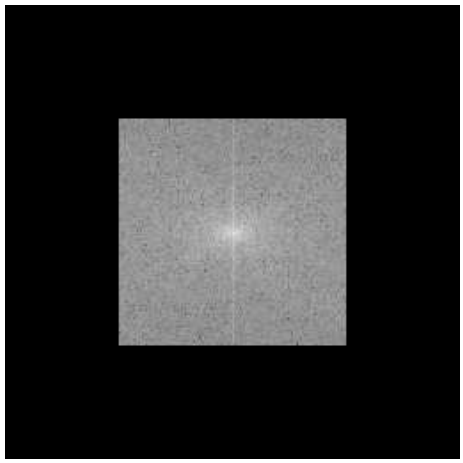
(c) Image sous-échantillonnée d'un facteur 2



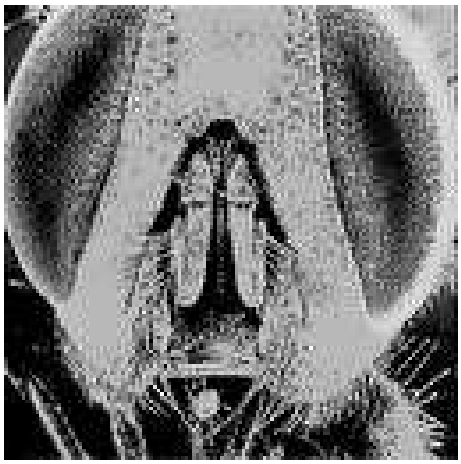
(d) La TFD correspondante, sur laquelle il y a repliement

Figure 9.5: Sous-échantillonnage et repliement: le cas d'une image mal échantillonnée. Pour les images (a), (b), (c), (d), le principe est le même que dans la figure 9.1.3, mais le détail de la transformation du spectre est plus difficile à suivre ! les effets du repliement (*aliasing* en anglais) sont particulièrement visibles sur les yeux de la mouche, image (c), qui présentent des oscillations à basse fréquence. Les structures quasi-périodiques de l'image originale sont visibles sous formes de taches et de filaments sur le spectre (b). Le repliement est dû à la présence de ces structures aux hautes fréquences: la TFD de l'image originale n'est pas nulle en dehors du carré visible en surimpression figure (b). Ce type d'effet de moiré est visible dans de nombreux DVD commerciaux.

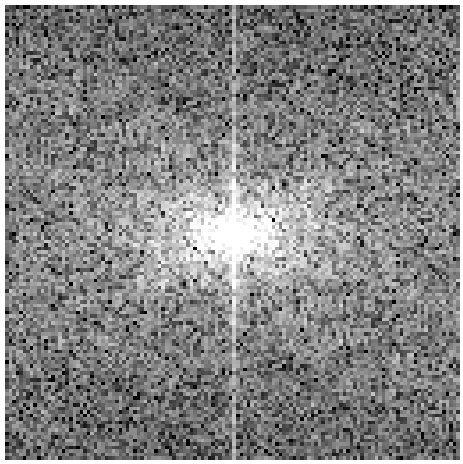
(a) Image obtenue par TFD inverse de b



(b) Image obtenue en mettant à zéro les hautes fréquences de 9.1.3-a



(c) Sous-échantillonnage: le repliement a disparu



(d) TFD de c

Figure 9.6: Une solution possible pour éviter les effets de repliement illustrés sur la figure 9.1.3. L'image (a) est l'image dont le spectre est le même que celui de l'image 9.1.3-(a) à l'intérieur du carré, et est nul à l'extérieur (filtrage passe-bas). L'image (c) est l'image sous-échantillonnée correspondante. On observe que l'effet de repliement a disparu.

de résoudre le problème en $\mathcal{O}(N \log N)$ opérations. Appelons "calcul d'ordre N" l'évaluation d'un polynôme de degré $N-1$ aux racines N-ièmes de l'unité. Et soit $T(N)$ le nombre d'opérations (additions et multiplications) demandées par ce calcul.

On se place dans le cas $N = 2^n$ et soit un polynôme

$$P(X) = \sum_{k=0}^{N-1} a_k X^k.$$

On pose

$$Q(X) = \sum_{k=0}^{\frac{N}{2}-1} a_{2k} X^k,$$

$$R(X) = \sum_{k=0}^{\frac{N}{2}-1} a_{2k+1} X^k.$$

Alors

$$P(\omega_N^k) = Q\left(\left(\omega_N^k\right)^2\right) + \omega_N^k R\left(\left(\omega_N^k\right)^2\right). \tag{9.9}$$

Or, si N est pair les $\left(\omega_N^k\right)^2$ sont exactement les racines d'ordre $\frac{N}{2}$ de l'unité. Il suffit donc d'évaluer les deux polynômes Q et R aux racines d'ordre $\frac{N}{2}$ de l'unité ce qui est un problème d'ordre $\frac{N}{2}$. On a donc, en tenant compte des additions et multiplications demandées par (9.9),

$$T(N) = 2T\left(\frac{N}{2}\right) + 2N.$$

On en tire aisément $T(N) = \mathcal{O}(N \log(N))$.

Remarque 9.1 *Les programmes usuels de calcul numérique ne calculent pas les coefficients $\tilde{u}_n$, mais les coefficients $\hat{u}_n$, définis par la formule suivante, pour $n = 0, \ldots, N-1$:*

$$\hat{u}_n = \begin{cases} \tilde{u}_n & si\ n = 0 \ldots \frac{N}{2} - 1 \\ \tilde{u}_{n-N} & si\ n = \frac{N}{2}, \ldots, N \end{cases}. \tag{9.10}$$

9.1.5 L'utilisation de la transformée de Fourier discrète pour définir zoom, translations et rotations des images

Le zoom Nous présentons une méthode d'interpolation reposant sur une extension de la TFD d'un signal ou d'une image. Nous détaillons la méthode, dite du "prolongement par des 0" ("0-padding"), en une dimension, le principe se généralisant sans mal pour une image. Comme précédement, considèrons des échantillons u_k, k variant de 1 à $N-1$, et $\tilde{u}_n = \frac{1}{N} \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} u_n \omega_N^{-kn}$. On suppose que N est pair et que l'on veut zoomer d'un facteur 2, c'est à dire que l'on veut construire un signal de taille deux fois plus grande que le signal de départ. On définit un nouveau signal v, de taille $2N$ comme étant la TFD inverse de $\tilde{v}$, donné par

$$\tilde{v}_n = \tilde{u}_n \ \text{ si } \ -\frac{N}{2} \leq n \leq \frac{N}{2} - 1, \ \ \tilde{v}_n = 0 \text{ si } n \in [-N, -\frac{N}{2} - 1] \cup [\frac{N}{2}, N-1]. \tag{9.11}$$

Proposition 9.8 *zoom discret par Fourier Le signal v dont la TFD est donnée par la formule (9.11) vérifie $v_{2k} = u_k$, pour $k = 0, \ldots, N-1$.*

Démonstration On a

$$v_{2k} = \sum_{-N}^{N-1} \tilde{v}_n \omega_{2N}^{2nk} = \sum_{-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{u}_n \omega_N^{nk} = u_k.$$

En effet, $\omega_{2N}^{2nk} = \omega_N^{nk}$. □

Remarque 9.2 *Ce résultat est évident sans démonstration : en effet, on peut considérer l'unique polynôme trigonométrique de degré $\frac{N}{2}$ passant par les échantillons u_k. Les échantillons v_k s'interprètent immédiatement comme des échantillons de ce même polynôme.*

Remarque 9.3 *On remarquera que les signaux obtenus par cette méthode peuvent être complexes, même lorsque le signal original est réel (ceci étant dû au terme d'aliasage $u_{-\frac{N}{2}}$).*

La méthode se généralise aux cas des images. Nous considérons une image numérique $(u_{k,l})$, et nous définissons une image zoomée $(v_{i,j})_{i,j=0,\ ...,\ 2N-1}$ comme étant la transformée de Fourier discrète inverse de $\tilde{v}_{i,j}$ définie pour $i, j = -N,\ ...,\ N-1$ par

$$\tilde{v}_{m,n} = \tilde{u}_{m,n} \text{ si } -\frac{N}{2} \leq m, n \leq \frac{N}{2} - 1, \quad \tilde{v}_{m,n} = 0 \text{ sinon.} \tag{9.12}$$

La figure 9.7 montre la partie réelle d'une partie de l'image 9.3 zoomée par TFD, ainsi que par réplication des pixels (chaque pixel est remplacé par quatre pixels de la même valeur). On remarque que le zoom par TFD produit une image bien plus régulière, et évite l'effet "marche d'escalier" visible sur l'image zoomée par réplication. La figure 9.8 illustre ce point sur un détail. Une autre remarque concerne l'effet de Gibbs (cf. paragraphe 8.4). Ce phénomène produit des rebonds le long de la frontière du domaine de l'image. En effet, et comme nous l'avons déjà mentionné, le calcul des coefficients de Fourier de l'image (dont les coefficients de la TFD sont une approximation) suppose l'image périodique, ce qui fait apparaître des discontinuités le long des frontières de son domaine de définition. Le phénomène de Gibbs est également visible le long des discontinuités dans l'image, les contours. Le phénomène est mis en évidence sur la figure 9.7. Expliquons pourquoi le phénomène apparaît dans le cas du zoom: une nouvelle image $v_{k,l}$ de taille $2N \times 2N$ est obtenue en utilisant les valeurs prises par le polynôme $P(x)$ entre les points dont on dispose au départ. Cette utilisation de P fait apparaître les oscillations qui étaient invisibles dans le cas de l'image de départ puisqu'il y avait interpolation des (u_k). Comme nous l'avons déjà évoqué, les oscillations aux frontières du domaine de l'image peuvent être supprimées par utilisation de la transformée en cosinus. En revanche, le problème subsistera le long des discontinuités présentes à l'intérieur de l'image.

La translation La méthode présentée au paragraphe précédent permet de définir une translation d'une quantité 1/2 (ou $a/(2N)$ pour revenir à notre définition première du signal u), en ne gardant que les points d'indice impair du signal zoomé v. Plus généralement, nous pouvons définir une translation d'un signal d'une quantité $0 < \alpha < 1$. Comme d'habitude, l'opération de translation sur la fonction u dont nous connaissons les échantillons u_k se fait sous l'hypothèse que celle-ci est un polynôme trigonométrique. En d'autres termes, on

PSfrag replacements

a
b

PSfrag replacements

a
b

Figure 9.7: Zoom sur une partie de l'image 9.3. Haut: zoom par TFD, bas: zoom par réplication des pixels. Le zoom par TFD est obtenu en prolongeant par des zéros le spectre de l'image initiale. Celui par réplication des pixels en remplaçant chaque pixel par quatre pixels de la même valeur. Remarquons tout d'abord la plus grande régularité du zoom par TFD, qui supprime les effets de "blocs" très visibles sur le zoom par réplication. En contrepartie, le phénomène de Gibbs (voir paragraphe 8.4) est très visible sur le zoom par TFD, puisque l'on a mis à zéro brutalement des coefficients de la TFD. Ce phénomène est particulièrement visible le long des frontières de l'image, qui correspondent à des discontinuités puisque l'image est périodisée (par exemple zone a), et des contours des objets (par exemple zone b).

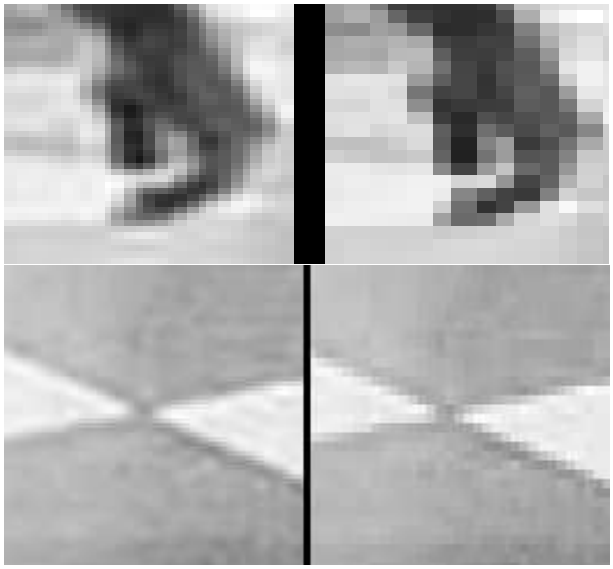


Figure 9.8: détails après zoom, à gauche par TFD, à droite par réplication des pixels.

translate le polynôme d'interpolation, la "vraie" fonction u étant inconnue en dehors des échantillons. Le polynôme d'interpolation est

$$P(x) = \sum_{-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{u}_n e^{\frac{2i\pi nx}{a}}.$$

En translatant de α, on obtient

$$\tau_\alpha P(x) = P(x-\alpha) = \sum_{-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{u}_n e^{-\frac{2i\pi n\alpha}{a}} e^{\frac{2i\pi nx}{a}}.$$

On a donc :

Proposition 9.9 *translatée!discrète par Fourier La TFD $(\tilde{v}_n)$ de $P(x-\alpha)$ s'obtient à partir de la TFD de $P(x)$, $\tilde{u}_n$, par*

$$\tilde{v}_n = \tilde{u}_n e^{-\frac{2i\pi n\alpha}{a}}.$$

Cette méthode de translation se généralise sans mal au cas des images, en remarquant qu'une translation à deux dimensions peut se décomposer en deux translations, une selon les lignes et une selon les colonnes.

La rotation Décrivons maintenant une méthode pour implémenter une rotation discrète, due à L. Yaroslavsky. En bref, cette méthode réduit une rotation à des translations en ligne ou en colonne de l'image. Commençons par remarquer que

$$R(-\theta) := \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} = \begin{pmatrix} 1 & \tan(\frac{\theta}{2}) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -\sin(\theta) & 1 \end{pmatrix} \begin{pmatrix} 1 & \tan(\frac{\theta}{2}) \\ 0 & 1 \end{pmatrix} := T(\theta)S(\theta)T(\theta) \tag{9.13}$$

Figure 9.9: Rotation de $\pi/4$ par TFD. La rotation est implémentée en remarquant qu'elle peut se décomposer en trois transformations consistant en des translations selon les lignes ou les colonnes de l'image (formule 9.13). Chacune de ces transformations est ensuite effectuée grâce à une TFD sur la ligne ou colonne considérée, en utilisant la méthode présentée au paragraphe précédent.

(sauf si $\theta = \pi$ auquel cas il suffit de retourner l'image).

Une rotation d'angle θ de l'image discrète $u(i,j)$ consiste à calculer $u(R(-\theta)(i,j))$ que l'on notera $(R(-\theta)u)(i,j)$. Mais on a $R(-\theta)u = T(\theta)S(\theta)T(\theta)u$. Donc il suffit d'expliquer comment calculer $T(\theta)u$ et $S(\theta)u$. Or ces deux opérations ont la même structure, à savoir une translation ligne par ligne ou une translation colonne par colonne. Traitons par exemple le cas de $T(\theta)$. On a $(T(\theta)u)(i,j) = u(i + j\tan(\frac{\theta}{2}), j)$. Donc partant de la matrice $u_{i,j}$, on translate sa première ligne de $\tan(\frac{\theta}{2})$, la deuxième de $2\tan(\frac{\theta}{2})$, etc.. Appliquer $S(\theta)$ revient à faire une opération similaire sur les colonnes. Enfin on réapplique $T(\theta)$ et on fait donc à nouveau une translation sur les lignes. Or comme on vient de le voir ces translations ligne à ligne ou colonne à colonne se font en temps $N \log N$ en utilisant la TFD à une dimension.

La figure 9.9 montre une image après une rotation de $\pi/4$ par la méthode décrite ci-dessus. Puis, pour illustrer la stabilité de la méthode, nous montrons figure 9.10 le résultat de l'application successive de douze rotations de $\pi/4$, et, à titre de comparaison, le résultat de ces douze rotations successives implémentés par interpolation bilinéaire (les valeurs aux nouveaux points sont des combinaisons linéaires des quatre points à coordonnées entières les plus proches). Cette figure illustre clairement la supériorité de la méthode par FFT dans le cas de rotations multiples.

Remarque 9.4 *Cette méthode présente un défaut. En effet, du fait que l'on manipule des fonctions périodiques, une translation conduit à faire sortir une partie de l'image par un bord pour la faire entrer par l'autre. Ce qui conduit à l'apparition, sur les bords de l'image d'un certain nombre de détails qui sont en fait mal placés. On se débarrasse facilement de ce*

Figure 9.10: Bas: après douze rotations successives de $\pi/4$ par TFD; haut: même expérience en utilisant une interpolation bilinéaire (la valeur en un nouveau point (x, y) est obtenue par combinaisons linéaires des valeurs aux quatre points à coordonnées entières de l'image originale les plus proches de (x, y)).

problème en insérant l'image dans un cadre deux fois plus grand...

Remarque 9.5 *La méthode de rotation n'est pas parfaite. En effet, l'image u continue associée à $u(i,j)$ est dans l'interpolation shannonienne supposée implicitement N-périodique, ce qui revient à dire qu'elle est de la forme (pour une image carrée)*

$$u(x,y) = \sum_{k,l=0}^{N-1} c_{i,j} e^{2i\frac{\pi}{N}(kx+ly)}.$$

Mais, si on lui applique une "translation" suivant l'axe des x de valeur λy, la formule devient

$$u_1(x,y) = \sum_{k,l=0}^{N-1} c_{i,j} e^{2i\frac{\pi}{N}(kx+(l-\lambda k)y)}.$$

La fonction u_1 n'est pas (pour $\lambda \notin \mathbf{Z}$) N-périodique en y. Or, après la première translation on ne dispose plus que des échantillons du signal u_1 sur une grille carrée $N \times N$. D'après la théorie de Shannon un tel ensemble de données ne permet pas de capturer toute l'information sur u_1 (à la seconde étape on effectue des translations suivant y qui est justement l'axe qui pose problème). On rencontre encore ce problème à la troisième translation. Le seul moyen d'avoir une rotation exacte serait d'évaluer u aux points de l'image de $[0,\ N-1] \times [0,\ N-1]$ par une rotation d'angle $-\theta$, mais cette méthode est en N^4 ce qui la rend inopérante...

9.1.6 Importances relatives de la phase et du module de la TFD pour une image

Nous nous intéressons à la pertinence visuelle des caractéristiques de la transformée de Fourier discrète dans le cas des images, et plus particulièrement à la phase et au module de la TFD, au moyen de deux exemples. Tout d'abord nous montrons, figure 9.11, deux images A et B, ainsi que les images obtenues en échangeant les phases de leurs TFD. Nous remarquons grâce à cette expérience qu'une part très importante de l'information géométrique d'une image est contenue dans la phase de sa TFD. Rappelons que si l'on translate une fonction, les coefficients de sa série de Fourier sont multipliés par des exponentielles complexes de module 1, et que par conséquent la phase de la TFD contient en en sens des informations sur le placement des constituants de l'image.

Dans la figure 9.12, nous montrons deux images de textures, qui visuellement semblent invariantes par translation, ainsi que les deux images obtenues à partir de ces textures en ne conservant que le module de leur TFD, et en tirant au hasard les phases (selon une loi uniforme). On voit cette fois que le module de la TFD contient l'information. Cette propriété est caractéristique des textures homogènes du type présenté figure 9.12, et l'on peut même donner une définition des "microtextures" comme images caractérisées uniquement par le module de leur transformée de Fourier.

9.2 Lien avec la théorie de Shannon

Théorème 9.2 *Shannon!théorème pour les polynômes trigonométriques* (de Shannon pour les polynômes trigonométriques) *Soit un signal trigonométrique*

$$f(t) = \sum_{n=-N}^{N} c_n e^{2i\pi\lambda_n t}.$$

(a) Image A

(b) Image A

(c) Module de la TF de A et phase de B

(d) Module de la TF de B et phase de A

Figure 9.11: Haut: les deux images de départ; bas: les deux images après échange des phases de leurs TFD. L'information géométrique est contenue dans la phase ! Les formes sont principalement codées dans l'argument des coefficients de Fourier de l'image. Bien que les images (a) et (c) d'une part, et (b) et (d) d'autre part, aient des modules complétement différents, on y distingue les mêmes formes géométriques. Remarquons également que les directions horizontales et verticales très présentes sur l'image (a) apparaissent sous forme de texture dans l'image (c). Cette remarque est précisée par l'expérience de la figure 9.12.

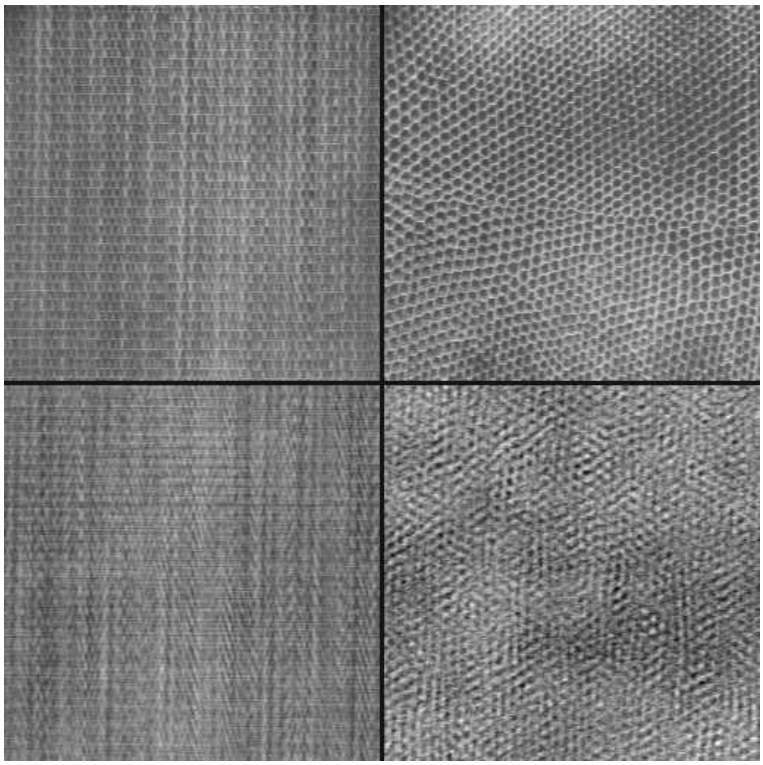


Figure 9.12: Haut: deux images de textures; bas: les deux images après remplacement des phases de leurs TFD par des phases aléatoires. Une information essentielle sur la texture est donc présente dans le module des coefficients de Fourier de l'image. Pour la texture de gauche, il semble que la plupart de l'information soit contenue dans le module de la TFD. A droite, quelques aspects de la texture sont perdus. Nous renvoyons le lecteur intéressé à un article (en anglais) sur une méthode de synthèse de texture par modification de la phase: [Van Wijk]

On a encore la formule de Shannon

$$\forall a \in \left]0, \frac{1}{2\lambda_c}\right[, \ \forall t \in \mathbb{R}, \ f(t) = \sum_{n=-\infty}^{+\infty} f(na) \frac{sin\frac{\pi}{a}(t-na)}{\frac{\pi}{a}(t-na)},$$

avec $\lambda_c = \max\{|\lambda_n|\}$. *La convergence est ponctuelle.*

Remarque 9.6 *Ce théorème complète le théorème de Shannon pour un signal qui est ni périodique ni dans* L^2.

Démonstration

il suffit de démontrer le résultat dans le cas d'une seule onde. Soit donc

$$f(t) = e^{2i\pi\lambda t}, \quad \lambda \in \mathbb{R}.$$

Soit g périodique de période $\frac{1}{a}$ et égale à f sur $\left(-\frac{1}{2a}, \frac{1}{2a}\right)$. les coefficients de Fourier de f sont

$$c_n = \frac{a\sin\frac{\pi}{a}(\lambda - na)}{\pi(\lambda - na)}.$$

Donc

$$g(t) = \sum_{-\infty}^{+\infty} \frac{\sin\frac{\pi}{a}(\lambda - na)}{\frac{\pi}{a}(\lambda - na)} e^{2i\pi nat}.$$

Comme f est C^1 sur $\left]-\frac{1}{2a}, \frac{1}{2a}\right[$, Cette égalité est ponctuelle pour $t \in \left]-\frac{1}{2a}, \frac{1}{2a}\right[$ (principe de localisation). D'où

$$\forall \lambda \in \mathbb{R}, \quad e^{2i\pi\lambda t} = \sum_{n=-\infty}^{+\infty} e^{2i\pi nat} \frac{\sin\frac{\pi}{a}(\lambda - na)}{\frac{\pi}{a}(\lambda - na)}, \quad |t| < \frac{1}{2a}.$$

En intervertissant λ et t, on obtient

$$\forall t \in \mathbb{R}, \quad e^{2i\pi\lambda t} = \sum_{n=-\infty}^{+\infty} e^{2i\pi na\lambda} \frac{\sin\frac{\pi}{a}(t - na)}{\frac{\pi}{a}(t - na)}, \quad |\lambda| < \frac{1}{2a}.$$

Chapter 10

La compression des images et la norme JPEG

Figure 10.1: Une photo de fleur compressée en JPEG, avec des compressions de plus en plus fortes, de gauche à droite.

Dans ce chapitre, la norme JPEG est décrite en détail sous ses aspects DCT, quantification, codage, compression avec ou sans perte. Ce chapitre reprend, avec quelques précisions, l'article de Wikipedia

http://fr.wikipedia.org/wiki/Norme_JPEG.

10.1 Introduction

La norme JPEG est une norme qui définit le format d'enregistrement et l'algorithme de décodage pour une réprésentation numérique compressée d'une image fixe. JPEG est l'acronyme de *Joint Photographic Experts Group*. C'est un comité d'experts qui édite des normes de compression pour l'image fixe. La norme communément appelée JPEG est le résultat de l'évolution des travaux qui ont débuté dans les années 1978 à 1980 avec les premiers essais en laboratoire de compression d'images. Le groupe JPEG qui a réuni une trentaine d'experts internationaux, a spécifié la norme en 1991. Mais la norme officielle et définitive n'a été adoptée qu'en 1992. La norme dont nous allons parler est

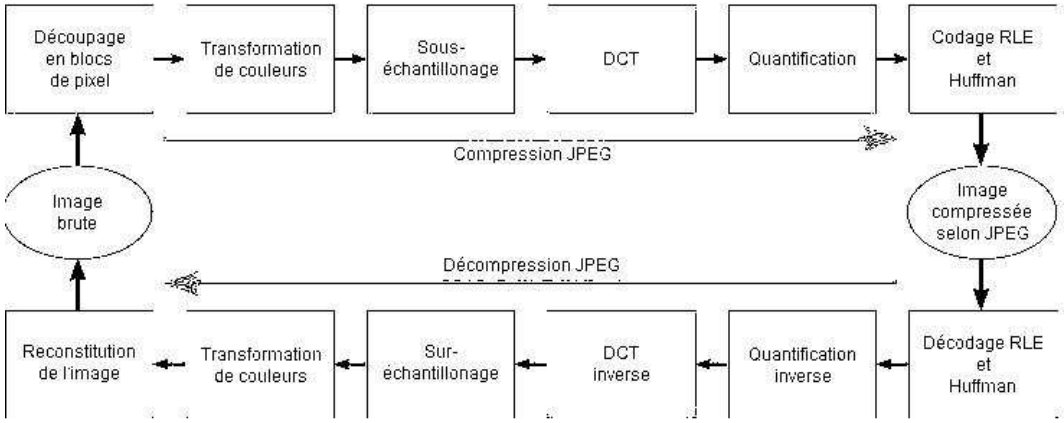


Figure 10.2: Organigramme de compression. Cette figure ne représente pas un cycle !

basée sur la DCT. Une norme plus récente, JPEG 2000, est basée sur la transformée en ondelettes, qui généralise la transformée de Fourier. JPEG normalise uniquement l'algorithme et le format de décodage. Le processus d'encodage est laissé libre à la compétition des industriels et universitaires, du moment que l'image produite est décodable par un décodeur standard. La norme propose un jeu de fichiers de tests appelés fichiers de conformance qui permettent de vérifier qu'un décodeur respecte bien la norme. Un décodeur est alors dit conforme s'il est capable de décoder tous les fichiers de conformance. JPEG définit deux classes de processus de compression : *avec pertes* ou compression irréversible. C'est le JPEG "classique". Il permet des taux de compression de 3 à 100. Le second est le processus sans pertes ou compression *réversible*. Il n'y a pas de perte d'information et il est donc possible de revenir aux valeurs originales de l'image. Les gains en terme de compression sont alors plus modestes, avec un taux de compression de l'ordre de 2. Cette partie fait l'objet d'une norme spécifique: JPEG-LS.

10.2 L'algorithme avec pertes

On peut diviser la compression et la décompression JPEG en six étapes données dans l'organigramme de la figure 10.2.

Les étapes sont:

Découpage en blocs

Le format JPEG, comme le font généralement les algorithmes de compression à perte, commence par découper l'image en blocs ou carreaux généralement carrés de 64 (8 x 8) ou 256 (16 x 16) pixels. L'utilité de ces petits blocs est que les chances augmentent que le bloc soit homogène, et donc facilement résumable en quelques coefficients de Fourier.

Transformation des couleurs

(Référence: http://fr.wikipedia.org/wiki/YUV). JPEG est capable de coder les couleurs sous n'importe quel format, toutefois les meilleurs taux de compression sont obtenus avec des codages de couleur de type luminance/chrominance tels que YUV, car l'oeil est assez sensible à la luminance mais peu à la chrominance. Les coordonnées YUV, où Y désigne la luminance et U et V désignent les composantes de chrominance peuvent en gros s'interpréter comme suit : Y est le niveau de gris, U est en gros la différence entre le bleu et le vert et V est à peu près la différence entre le rouge et le vert. Dans le cas

d'une image grise (image noir et blanc), comme $R = G = B$, on a $U = V = 0$, ce qui veut dire que l'image n'a pas de chrominance, mais seulement une luminance. Plus précisément,

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ -0,147 & -0,289 & 0,436 \\ 0,615 & -0,515 & 0,100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}.$$

Remarquer que la somme des coefficients des deux dernières lignes est nulle. La compression va traiter sommairement les composantes U, V et va être plus fidèle pour la composante Y, qui contient l'information géométrique de l'image.

Sous-échantillonnage

La façon la plus simple d'exploiter la faible sensibilité de l'oeil à la chrominance est simplement de sous-échantillonner les signaux de chrominance. Généralement on utilise un sous-échantillonnage de type 2h1v ou 2h2v. Dans le premier cas (le plus utilisé) on a un sous-échantillonnage 2:1 horizontalement et 1:1 verticalement, dans le deuxième cas on a un sous-échantillonnage 2:1 horizontalement et verticalement. Ces sous-échantillonnages sont utilisés pour les chrominances, pour la luminance on n'utilise jamais de sous-échantillonnage.

Transformée en cosinus discrète

La transformée DCT (*Discrete Cosine Transform*) est une transformation numérique qui est appliquée à chaque bloc et pour chaque " couleur ". Cette transformée est une variante de la transformée de Fourier. Cette méthode permet de décrire chaque bloc en une carte de fréquences et amplitudes plutôt qu'en pixels et couleurs. La valeur d'une fréquence reflète l'importance et la rapidité d'un changement, tandis que la valeur d'une amplitude correspond à l'écart associé à chaque changement de couleur. À chaque bloc de pixels sont ainsi associées 64 fréquences. La transformée DCT directe est

$$c(m,n) =: \frac{2}{N} C(m)C(n) \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} u(k,l) \cos\frac{(2k+1)m\pi}{2N} \cos\frac{(2l+1)n\pi}{2N},$$

et la transformée DCT inverse s'exprime par

$$u(k,l) =: \frac{2}{N} C(k)C(l) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} u(m,n) \cos\frac{(2m+1)k\pi}{2N} \cos\frac{(2n+1)l\pi}{2N}.$$

Dans les deux cas, la constante vaut $C(m) = \frac{1}{\sqrt{2}}$ pour $m = 0$ et $m = 1$ sinon. Pourquoi la DCT plutôt que la DFT (*discrete Fourier transform*)? On rappelle que la DCT est en fait une DFT appliquée à une image quatre fois plus grande obtenue en symétrisant l'image par rapport à ses cotés gauche et bas et par rapport à son coin bas et gauche. Cette nouvelle image reste continue quand on la périodise. Ainsi, l'analyse de Fourier s'applique à une image vraiment périodique, ce qui évite les forts coefficients de Fourier associés aux discontinuités produites par une périodisation directe.

Pour illustrer la compression, a été repris un exemple complet provenant de "Digital Images Compression Techniques" de Majid Rabbani et Paul W. Jones. Matrice (bloc de pixels) de base :

$$f = \begin{pmatrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 159 & 160 & 161 & 162 & 162 & 155 & 155 & 155 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 163 & 162 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 \end{pmatrix}$$

En effectuant la transformée DCT on obtient la matrice des fréquences suivante :

$$\begin{pmatrix} 1260 & -1 & -12 & -5 & 2 & -2 & -3 & 1 \\ -23 & -17 & -6 & -3 & -3 & 0 & 0 & -1 \\ -11 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -7 & -2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 2 & 0 & -1 & 1 & 1 \\ 2 & 0 & 2 & 0 & -1 & 1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 2 & 1 & -1 \\ -3 & 2 & -4 & -2 & 2 & 1 & -1 & 0 \end{pmatrix}$$

Le calcul d'une DCT est l'étape qui coûte le plus de temps et de ressources dans la compression et la décompression JPEG, mais c'est peut-être la plus importante car elle permet de séparer les basses fréquences et les hautes fréquences présentes dans l'image. Remarquer que les coefficients grands se concentrent dans les fréquences basses.

Quantification

La quantification est l'étape dans laquelle on perd réellement des informations (et donc de la qualité visuelle), mais c'est celle qui fait gagner beaucoup de place (contrairement à la DCT, qui ne compresse pas). La DCT a retourné, pour chaque bloc, une matrice de 8×8 nombres (dans l'hypothèse que les blocs de l'image font 8×8 pixels). La quantification consiste à diviser cette matrice par une autre, appelée matrice de quantification, et qui contient 8×8 coefficients savamment choisis par le codeur. Le but est ici d'atténuer les hautes fréquences, c'est-à-dire celles auxquelles l'oeil humain est très peu sensible. Ces fréquences ont des amplitudes faibles, et elles sont encore plus atténuées par la quantification (les coefficients sont même ramenés à 0). Voici le calcul permettant la quantification :

$$F^\star(u,v) =: \lfloor \frac{F(u,v) + \lfloor \frac{Q(u,v)}{2} \rfloor}{Q(u,v)} \rfloor \simeq \text{ entier le plus proche de } \left(\frac{F(u,v)}{Q(u,v)} \right),$$

avec $\lfloor x \rfloor$ désignant l'entier directement inférieur à x. Et pour la quantification inverse :

$$\hat{F}(u,v) = F^\star(u,v) Q(u,v).$$

Comme le montre l'image ci-dessous la quantification ramène beaucoup de coefficients à 0 (surtout en bas à droite dans la matrice, là où sont les hautes fréquences). Seules quelques informations essentielles (coin en haut à gauche) sont gardées pour représenter le bloc. L'intérêt est qu'au moment de coder le résultat dans le fichier, la longue suite de zéros nécessitera très peu de place. Mais si la quantification est trop forte (= taux de compression trop élevé), il y aura trop peu de coefficients non nuls pour représenter fidèlement le bloc ; dès lors, à l'écran la division en blocs devient visible, et l'image apparaît " pixellisée ". Dans notre exemple nous avons pris la matrice de quantification suivante :

$$Q =: \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

Ce qui donne comme matrice des fréquences quantifiée :

$$F^\star =: \begin{pmatrix} 79 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Codage RLE

(Voir http://fr.wikipedia.org/wiki/Run-length-encoding.) L'étape suivante demande un *run-length encoding*, en français codage par plages. C'est un algorithme élémentaire de compression de données en informatique que nous allons expliquer maintenant.

Le RLE s'applique essentiellement à des documents scannés en noir et blanc : au lieu de coder un bit par point, on dispose d'un compteur en général sur un octet indiquant combien de points blancs ou noirs se suivent. Comme il est rare de ne pas avoir au moins 8 pixels noirs ou 8 pixels blancs qui se suivent, et que 256 ne sont pas rares sur les endroits vierges ou les à-plats noirs, le système a bien pour effet une compression. S'il y a plus de 256 bits de la même couleur, on peut placer ensuite un octet spécifiant 0 bit de la couleur opposée, puis coder le nombre de bits qui restent...

Par exemple, considérons un écran de texte noir sur fond blanc. Il sera constitué de longues séquences de pixels blancs pour le fond, et de courtes séquences de pixels noirs pour le texte. Représentons une ligne d'un tel écran, avec B pour les pixels noirs et W pour les pixels blancs :

$$vvvvvvvvvvvvbvvvvvvvvvvvvvvvbbbvvvvvvvvvvvvvvvvvvvvvvvvbvvvvvvvvvvv$$

Un encodage RLE consiste alors à indiquer pour chaque suite de pixels d'une même couleur, le nombre de pixels de cette séquence. Le résultat comporte en général moins de caractères, bien que ce ne soit pas une obligation. On obtient par exemple pour la ligne précédente :

$$12v1b14v3b23v1b11v,$$

tandis que :

$$vbvbvbvbvb$$

donnerait

$$1v1b1v1b1v1b1v1b1v1b,$$

qui est passablement plus longue.

Les formats d'images utilisent cette compression en considérant que toutes les lignes de pixels sont jointes pour former une unique séquence de couleur.

Le format BMP de Windows et OS/2 permet d'utiliser la compression RLE pour les images en 1, 4 et 8 bits/pixel (respectivement noir et blanc, 16 couleurs et 256 couleurs). Ce n'est pas une très bonne compression, dès que la gamme de couleurs est riche, mais elle est excellente pour les images en noir et blanc. Le format PCX utilise également le principe de la compression RLE pour les images en 8 et 24 bits/pixel. Dans le cas des images en 24 bits/pixel, l'image est en fait découpée en trois plans de couleur (rouge, vert et bleu) où chaque plan est encodé comme une image en 8 bits/pixel. RLE est aussi utilisé pour les faxs, ce qui est son usage le plus fréquent hors informatique.

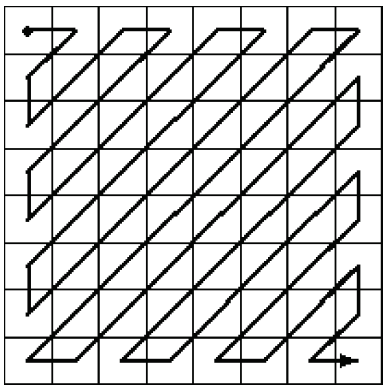


Figure 10.3: Ordre de codage défini par la norme JPEG.

Codage JPEG: compression RLE et Huffman

Revenons à JPEG. Le codage RLE des coefficients de la DCT quantifiés s'effectue en zigzag comme le montre la figure 10.3 et se termine par un caractère de fin :

Par exemple le codage de notre exemple est :

$$79,\ 0,\ -1,\ -1,\ -1,\ 0,\ 0,\ -1,\ \ \text{EOB}.$$

Ce résultat est ensuite compressé avec un RLE basé sur la valeur 0 (le codage RLE intervient uniquement sur cette dernière), puis un codage entropique de type Huffman ou arithmétique qui utilise le fait que la distribution des valeurs de la DCT est concentrée autour de quelques valeurs autour de 0, donc une distribution discrète à entropie faible. Avec le schéma de codage très simplifié suivant on remarque que le codage nous délivre deux tables (quatre pour une image couleur). Ces tables étant enregistrées dans le fichier final peuvent être choisies par le compresseur.

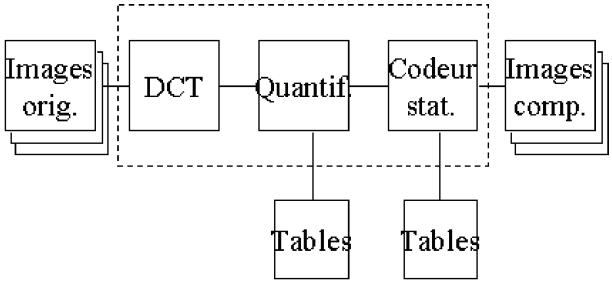


Figure 10.4: Schéma de codage simplifié.

La décompression JPEG

Les étapes de la décompression s'effectuent dans l'ordre inverse de la compression suivant les méthodes définies précédemment (en même temps que la compression). Voici dans notre exemple le résultat de la décompression :

$$\begin{pmatrix} 144 & 146 & 149 & 152 & 154 & 156 & 156 & 156 \\ 148 & 150 & 152 & 154 & 156 & 156 & 156 & 156 \\ 155 & 156 & 157 & 158 & 158 & 157 & 156 & 155 \\ 160 & 161 & 161 & 162 & 161 & 159 & 157 & 155 \\ 163 & 163 & 164 & 164 & 162 & 160 & 158 & 156 \\ 163 & 163 & 164 & 164 & 162 & 160 & 158 & 157 \\ 160 & 161 & 162 & 162 & 162 & 161 & 159 & 158 \\ 158 & 159 & 161 & 161 & 162 & 161 & 159 & 158 \end{pmatrix}$$

Ainsi que la matrice d'erreur :

$$\begin{pmatrix} -5 & -2 & 0 & 1 & 1 & -1 & -1 & -1 \\ -4 & 1 & 1 & 2 & 3 & 0 & 0 & 0 \\ -5 & -1 & 3 & 5 & 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & -2 & -1 & 0 & 2 & 4 \\ -1 & 0 & 1 & -2 & -1 & 0 & 2 & 4 \\ -2 & -2 & -3 & -3 & -2 & -3 & -1 & 0 \\ 2 & 1 & -1 & 1 & 0 & -4 & -2 & -1 \\ 4 & 3 & 0 & 0 & 1 & -3 & -1 & 0 \end{pmatrix}$$

Les erreurs sont au maximum de 5 et en moyenne 1,6 sur environ 150 ce qui nous donne une erreur moyenne d'environ 1, et tout cela pour un passage de 64 à 10 valeurs (avec le caractère de fin) ; à cela il faut rajouter la matrice de quantification, mais comme généralement on compresse de gros fichiers, elle n'influence que peu.

10.3 JPEG, codage sans pertes

À la place de la DCT, le codage sans pertes utilise un prédicteur qui permet de coder une différence entre valeur prédite et valeur observée, au lieu de la valeur elle-même. Pour se faire une première idée de comment marche un tel codage prédictif, on examinera un compresseur utilisant la prédiction la plus banale. Il consiste à lire l'image ligne à ligne et de gauche à droite. La première valeur $u(i,1)$ de chaque ligne est gardée telle quelle, et les valeurs suivantes sont remplacées par $u(i+1,j) - u(i,j)$. L'image étant régulière, ces valeurs sont statistiquement petites: leur distribution est concentrée autour de zéro. Il en résulte que l'entropie de l'ensemble des valeurs $u(i+1,j) - u(i,j)$ a un nombre de bits nettement plus petit que huit (en pratique, proche de 4). On peut donc pratiquer une compression sans perte en utilisant un codage de Hufman de ces valeurs. Ce codage prédictif s'améliore facilement avec un prédicteur un peu plus sophistiqué: utilisant toujours la régularité locale de l'image, on "prédit" la valeur $u(i+1,j+1)$ à partir des trois valeurs $u(i,j)$, $u(i+1,j)$, $u(i,j+1)$, la prédiction est tout bonnement linéaire. La valeur prédite est

$$\tilde{u}(i+1,j+1) =: u(i+1,j) + u(i,j+1) - u(i,j).$$

Ensuite on fait un codage de Huffman de la séquence des différences $\tilde{u}(i,j) - u(i,j)$. Pour démarrer la prédiction, les valeurs $u(1,1)$, $u(1,2)$ et les premières valeurs de chaque ligne $u(k,1)$ sont gardées telles quelles.

10.4 Exercices et implémentation Matlab

Exercice 55 1) Récupérer sur le web un code public JPEG, de préférence en Matlab, et commenter en détail chaque partie, en particulier, discuter la partie "lossless" (compression sans perte) dans son lien avec la théorie du codage de Shannon, et la partie compression avec perte, dans son lien avec l'analyse de Fourier et le filtrage.

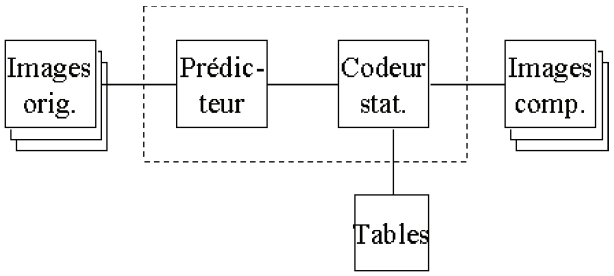


Figure 10.5: Schéma de compression JPEG sans pertes.

2) Appliquer JPEG à plusieurs images avec des taux de compression allant de faible à très fort et discuter la qualité visuelle des images.

Exercice 56 Compression sans perte

1) Implémenter une compression sans perte avec le premier prédicteur banal décrit plus haut.

2) Implémenter le prédicteur linéaire à trois valeurs indiqué plus haut

3) Essayer d'inventer un prédicteur plus sophistiqué (le premier se base sur l'idée que l'image est localement constante, le second sur l'idée qu'elle est localement linéaire, donc il faut passer à une prédiction quadratique...)

Chapter 11

Ondelettes de Malvar-Wilson et segmentation de la voix

Exercice 57 La DCT (*discrete cosine transform*, transformée de Fourier discrète.) On veut analyser un signal discret $(u_0,\ ...,\ u_{N-1})$ de $l^2(0,\ 1, ..., N-1)$ en évitant de créer un saut en 0 par périodisation. Pour cela, on procède comme pour les bases en cosinus : on étend le signal en un signal de $l^2(0,\ ...,\ 2N-1)$ en faisant une symétrie par rapport à $N-\frac{1}{2}$. On a créé ainsi une suite à $2N$ coefficients, que l'on analyse par transformée de Fourier discrète. Sa $2N$-périodisée n'a plus de discontinuité artificielle.

1) Calculer la transformée de Fourier discrète $(\tilde{v}_n)_{n=-N,\ ...N-1}$ du nouveau signal discret

$$(v_0, \ldots,\ v_{2N-1}) = (u_0,\ ...,\ u_{N-1},\ u_{N-1},\ u_{N-2},\ ..., u_0).$$

Montrer que

$$\tilde{v}_n = \frac{1}{N}\omega_{2N}^{\frac{n}{2}} \sum_{l=0}^{N-1} u_l \cos(n(l+\frac{1}{2})\frac{\pi}{N}),\ \ n = -N,\ ...,\ N-1.$$

2) Vérifier que $\tilde{v}_n = \tilde{v}_{-n}\omega_{2N}^n$ et $\tilde{v}_{-N} = 0$.

3) En déduire que la transformation $u = (u_0, \ldots,\ u_{N-1}) \to (\frac{\tilde{v}_0}{\sqrt{2}},\ \tilde{v}_1, \ldots,\ \tilde{v}_{N-1})$ est une isométrie, à un facteur près que l'on précisera. On pourra commencer par vérifier que la transformée de Fourier discrète est une isométrie à un facteur multiplicatif. Pour cela: vérifier que sa matrice est proportionnelle à une matrice unitaire. On rappelle qu'une matrice unitaire U est une matrice telle que $U\,{}^t\overline{U} = Id$.

4) Dans la suite, on pose, pour $1 \leq n \leq N-1$, $\tilde{w}_n = \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} (u_l \cos n(l+\frac{1}{2})\frac{\pi}{N})$ and $\tilde{w}_0 = \frac{1}{\sqrt{2N}} \sum_{l=0}^{N-1} u_l$. Déduire de la question précédente que l'application $u \to w$ est aussi une isométrie. On l'appelle "transformée en cosinus discrète" (DCT).

5) Montrer sans calcul que la transformation inverse de la DCT est donnée par

$$u_l = \frac{1}{\sqrt{N}} \sum_{n=1}^{N-1} \tilde{w}_n \cos(n(l+\frac{1}{2})\frac{\pi}{N}) + \tilde{w}_0 \frac{1}{\sqrt{2N}}.$$

Exercice 58 Montrer que la suite $u_k(t) = \sqrt{\frac{2}{\pi}}\cos(k+\frac{1}{2})t,\ k \in I\!N$, est une base orthonormée de $L^2(0,\pi)$. Indication : considérer l'espace E des fonctions 4π-périodiques, paires, et vérifiant $f(2\pi - t) = -f(t)$ et écrire le développement de Fourier de ces fonctions. Remarquer ensuite que ces fonctions sont entièrement déterminées par leur restriction à $[0,\ \pi]$.

Exercice 59 Les ondelettes de Malvar discrètes (Yves Meyer, Ondelettes et Algorithmes Concurrents, Hermann, pp.19-23).

Soit un signal de parole déja échantillonné. On le prend de longueur arbitraire et il appartient donc à $l^2(\mathbf{Z})$. Par ailleurs, on va supposer qu'il est déja segmenté, ce qui veut dire qu'on sait le couper en tranches temporelles pertinentes $[a_j, a_{j+1}]$ avec $a_j \to \pm\infty$ quand $j \to \pm\infty$. On suppose que $a_j + \frac{1}{2} \in \mathbf{Z}$ et que les a_j forment une suite strictement croissante (les a_j sont placés à mi-chemin entre deux échantillons pour couper proprement). On pose $l_j = a_{j+1} - a_j$ et on considère une suite $\eta_j \in I\!N^*$ telle que $\eta_j + \eta_{j+1} \leq l_j$. Les tranches temporelles $[a_j, a_{j+1}]$ correspondent à des notes, ou des voyelles, ou des consonnes, ayant une certaine cohérence fréquentielle. L'analyse de Fourier est justifiée et on peut utiliser la DCT dans chaque tranche. On a ainsi une décomposition orthogonale de $l^2(\mathbf{Z})$ en espaces $E_j = l^2(\mathbf{Z} \cap [a_j, a_{j+1}]))$, par exemple, et chacun de ces espaces dispose à son tour d'une base orthogonale par DCT.

1) Formaliser le raisonnement précédent et montrer qu'il amène à conclure que le système

$$u_{j,k}(x) = \frac{1}{\sqrt{l_j}} \cos(k(x - a_j)\frac{\pi}{l_j}) \mathbb{1}_{[a_j, a_{j-1}]},\ 1 \leq k \leq l_j - 1 \ \text{ et } \ u_{j,0}(x) = \frac{1}{\sqrt{2l_j}} \mathbb{1}_{[a_j, a_{j-1}]},\ j \in \mathbf{Z} \quad (11.1)$$

est une base orthonormée de $l^2(\mathbf{Z})$.

Saucissonnage mou pour les sons: pourquoi? On obtient donc à peu de frais une base orthonomale de $l^2(\mathbf{Z})$. Il est toutefois plus habile d'utiliser un découpage "mou" de la droite temporelle, afin d'éviter les effets de bord dûs au saucissonnage du signal. Ces effets de bord sont essentiellement liés au fait que la transformée de Fourier sur un intervalle de longueur T traite le signal comme T-périodique, et donc crée une discontinuité artificielle au bord. Une première réponse est d'utiliser la transformée en cosinus, qui revient à symétriser le signal par rapport à une des bornes de l'intervalle, puis à $2T$-périodiser le signal obtenu, qui reste alors continu. Ce procédé, qui donne des résultats satisfaisants pour les images, n'est encore pas suffisant pour les sons. Par la DCT, la dérivée du signal aux bornes de l'intervalle reste discontinue, et l'application de la DCT à des intervalles du son par saucissonnage dur provoque des "clics" désagréables à l'audition. Il faut donc faire un découpage mou, où chaque portion du son naît très doucement au début de l'intervalle et meurt très doucement à la fin. Cela ne peut se faire qu'avec des intervalles recouvrants.

Le problème est alors de maintenir l'orthogonalité, puisque les intervalles d'un découpage mou (une partition de l'unité) se recouvrent partiellement. C'est ce problème qu'ont résolu indépendamment il n'y a pas si longtemps (1990) un prix Nobel de physique, Kenneth Wilson, et un spécialiste de traitement du signal, Enrique Malvar. La présentation simple que nous donnons suit une construction de Ronald Coifman et Yves Meyer (1997). La partition de l'unité est donnée par des fenêtres $w_j(x)$, $x \in \mathbf{Z}$, vérifiant

$$0 \leq w_j \leq 1 \text{ et } w_j(x) = 1 \text{ si } a_j + \eta_j \leq x \leq a_{j+1} - \eta_{j+1}, \quad (11.2)$$

$$w_j(x) = 0 \text{ si } x \leq a_j - \eta_j \text{ ou si } x \geq a_{j+1} + \eta_{j+1}, \quad (11.3)$$

$$\text{si } x = a_j + t \text{ et } |t| \leq \eta_j, \text{ alors } w_{j-1}(a_j - t) = w_j(a_j + t) \text{ et } w_{j-1}^2 + w_j^2(x) = 1. \quad (11.4)$$

2) Remarquer que les w_j^2 forment une partition de l'unité, c'est-à-dire $\sum_j w_j^2 = 1$.

3) Définir et dessiner des exemples de fonctions w_j vérifiant les conditions (11.2-11.3-11.4).

Le but des deux questions qui suivent est de comprendre la difficulté du problème posé, à savoir : trouver une base de Fourier qui soit à la fois localisée sur les intervalles $[a_j - \eta_j,\ a_{j+1} + \eta_{j+1}]$ et orthonormale. La tentative la plus naturelle est d'écrire pour toute suite u de $l^2(\mathbf{Z})$: $u = \sum_j w_j^2 u$. Ensuite, les tranches molles $w_j u$ peuvent subir une analyse de Fourier locale sur les intervalles $[a_j - \eta_j,\ a_{j+1} + \eta_j]$. Voyons que ça marche, mais que ce n'est pas parfait.

4) Montrer en utilisant la partition de l'unité précédente et la DCT que tout signal de $l^2(\mathbf{Z})$ peut s'écrire sous la forme

$$u = \sum_{j\in\mathbf{Z}} \sum_{0\le k\le l'_j-1} c_{j,k} w_j(x) \cos\frac{k\pi}{l'_j}(x-a'_j),$$

où on a posé $l'j = a_{j+1} + \eta_{j+1} - a_j - \eta_j$ et $a'j = a_j - \eta_j$. On précisera les coefficients $c_{j,k}$.

5) Montrer que les fonctions $w_j(x)\cos\frac{k\pi}{l'_j}(x-a'_j),\ j\in\mathbf{Z},\ 0\le k\le l_j-1$ ne forment pas une base orthogonale de $l^2(\mathbf{Z})$.

Pour restaurer l'orthogonalité, on a besoin d'une construction du type précédent, mais un peu plus sophistiquée. On appelle *ondelettes de Malvar* les fonctions $u_{j,k}(x)$, $j\in\mathbf{Z}$, $0\le k\le l_j-1$ définies par

$$u_{j,k}(x) = \sqrt{\frac{2}{l_j}} w_j(x) \cos\left(\pi(k+\frac{1}{2})(\frac{x-a_j}{l_j})\right). \tag{11.5}$$

Théorème 11.1 *La suite $u_{j,k}$ est une base orthonormée de $l^2(\mathbf{Z})$.*

6) Comparer la forme des ondelettes de Malvar à la base orthonormale de $l^2(\mathbf{Z})$ donnée par (11.1).

La démonstration du théorème va se faire en deux étapes que ce problème va détailler :
(I) Décomposer $l^2(\mathbf{Z})$ en une somme d'espaces orthogonaux E_j de dimension l_j.
(II) Vérifier que pour chaque j les fonctions $u_{j,k}$, $0\le k\le l_j-1$, forment une base orthonormée de E_j.

Définissons E_j. Soit F_j l'espace des fonctions $g\in l^2(\mathbf{Z})$ nulles hors $[a_j-\eta_j,\ a_{j+1}+\eta_{j+1}]$ et vérifiant

$$g(a_j+t) = g(a_j-t) \text{ si } |t|\le\eta_j \tag{11.6}$$

$$g(a_{j+1}+t) = -g(a_{j+1}-t) \text{ si } |t|\le\eta_{j+1} \tag{11.7}$$

7) Montrer que $\dim F_j = l_j$. On dira que $f\in E_j$ si et seulement si $f = w_j g$ où $g\in F_j$.

8) Dessiner sur un même graphe : w_j, w_{j+1}, et une fonction $g\in F_j$. On peut résumer la situation ainsi : On part d'un signal $u\in l^2(\mathbf{Z})$, on considère sa restriction u_j à l'intervalle $[a_j, a_{j+1}]$. Cette restriction définit une unique fonction $g\in F_j$, en étendant u_j à gauche de l'intervalle par parité et à droite par imparité. La fonction $g\in F_j$ obtenue est ensuite ramenée vers zéro aux bords de l'intervalle $[a-j-\eta_j,\ a_{j+1}+\eta_{j+1}]$: il suffit de la multiplier par la fonction fenêtre w_j. Cette construction va permettre d'obtenir l'orthogonalité des E_j, grâce à l'alternance de prolongements pairs et impairs.

9) Vérifier que les E_j sont orthogonaux entre eux : commencer par remarquer qu'il suffit de considérer E_j et E_{j-1}. Si $f_j\in E_j$ et $f_{j-1}\in E_{j-1}$, on peut écrire $f_j = w_j g_j$, $f_{j-1} = w_{j-1} g_{j-1}$. L'orthogonalité se déduit des propriétés de parité ou d'imparité des fonctions considérées sur l'intervalle $[a_j-\eta_j,\ a_j+\eta_j]$.

10) Vérifier que les fonctions $u_{j,k}$ sont dans E_j.

12) Montrer en utilisant (11.2-11.4) que si f_1 et f_2 appartiennent à E_j,

$$\sum_{z\in\mathbf{Z}} f_1(x)\overline{f_2(x)} = \sum_{a_j<x<a_{j+1}} g_1(x)\overline{g_2(x)}. \tag{11.8}$$

13) En déduire que l'application $U_j : E_j \to l^2(a_j+\frac{1}{2},\ldots,\ a_{j+1}-\frac{1}{2})$ est un isomorphisme isométrique.

14) Montrer que les fonctions $\sqrt{\frac{2}{N}}\cos\left(\frac{\pi}{N}(k+\frac{1}{2})(x+\frac{1}{2})\right)$, $0\le k<N-1$ forment une base orthonormée de $l^2(0,\ 1,\ldots,\ N-1)$. Remarque : leur nombre est égal à la dimension N de l'espace et il suffit

donc de vérifier leur orthonormalité. Pour montrer l'orthogonalité, utiliser les identités $2\cos a\,\cos b = \cos(a-b) - \cos(a+b)$, puis

$$\sum_{0\leq x<N} \cos\left[\frac{\pi}{N} m(x+\frac{1}{2})\right] = 0 \text{ si } 1 \leq m \leq 2N-1.$$

Cette dernière identité s'obtient en calculant $\sum_{0\leq x<N} e^{i\frac{\pi}{N}m(x+\frac{1}{2})}$.

15) Déduire des questions précédentes que le résultat (II) est juste.

Il nous faut maintenant montrer que si $f \in l^2(\mathbf{Z})$ est orthogonale à tous les E_j, alors elle est nulle.

16) Commencer par vérifier que si $x_0 \in [a_j + \eta_j,\ a_{j+1} - \eta_{j+1}]$, alors la fonction égale à 1 en x_0 et à 0 ailleurs appartient à E_j. En déduire que f est nulle sur ces intervalles.

17) En utilisant des fonctions adéquates dans E_j et E_{j+1}, montrer que si on pose $x = a_j + t$ et $x' = a_j - t$, où $|t| \leq \eta_j$, $t - \frac{1}{2} \in \mathbf{Z}$, alors on a

$$f(x)w_j(x) + f(x')w_j(x') = 0 \text{ et } f(x)w_{j-1}(x) - f(x')w_{j-1}(x') = 0.$$

En déduire que $f(x) = f(x') = 0$ et conclure.

Exercice 60 Commentaire dirigé de l'article *Entropy-based Algorithms for Best Basis Selection*, de Ronald R. Coifman et Mladen V. Wickerhauser

1) Lire l'article.

2) Etablir le lien entre les fonctions $S_{i,k}$ introduites dans la page 2 de l'article et les ondelettes de Malvar-Wilson telles qu'elles sont décrites dans le problème précédent.

3) Démontrer la relation additive donnée page 4 de l'article que les auteurs commentent en disant *This Shannon's equation for entropy ....*

4) Démontrer la première proposition de la page 4 concernant la dimension d'un vecteur.

5) Démontrer la deuxième proposition de la page 4, concernant la relation entre dimension d'un vecteur et concentration de ses coefficients.

6) Développer la preuve de la dernière proposition, qui justifie l'algorithme de meilleure base. La preuve donnée est-elle correcte? Avez-vous une opinion sur l'algorithme proposé: trouve-t-il vraiment la meilleure base de Malvar-Wilson?

Conseils de lecture :

Claude E. Shannon et Warren Weaver The mathematical theory of communication University of Illinois Press 1998.
Thomas M. Cover et Joy A. Thomas Elements of Information Theory, Wiley Series Telecommunications, (chapitres 2, 5 et 8), 1991.
Pierre Brémaud Introduction aux probabilités, Springer.
J.M. Bony, Cours d'Analyse de l'Ecole Polytechnique, Ecole Polytechnique, 1994 (polycopié).
J.M. Bony, Cours de Méthodes Mathématiques pour les Sciences Physiques, Ecole Polytechnique, (1997). (polycopié).
C. Gasquet et P. Witomski, Analyse de Fourier et applications, Masson (1995).
S. Mallat, A Wavelet Tour of Signal Processing, Academic Press, (1997).
S. Mallat, Une exploration des signaux en ondelettes, Editions de l'Ecole Polytechnique, 2000.
S. Mallat Traitement du Signal, polycopié de la Majeure de Mathématiques Appliquées, Ecole Polytechnique, 1998.
Y. Meyer, Wavelets, Algorithms and Applications, SIAM (1993), translated and revised by Robert Ryan.
Y. Meyer, Ondelettes et Algorithmes concurrents. Hermann, Paris (1992).
Y. Meyer, cours de DEA ondelettes (1994-1997 (manuscrits).
L. Yaroslavsky, M. Eden, Fundamentals of Digital Optics, Birkhäuser, (1996). Sources: Claude E. Shannon A mathematical source of communication.
Pierre Brémaud Introduction aux probabilités, Chapitre 5, Springer.
Thomas M. Cover, Joy A. Thomas, Elements of information theory (pages 194-197), Wiley, 1991.
Thomas M. Cover et Joy A. Thomas Elements of Information Theory, Wiley Series Telecommunications.
Eva Wesfreid, Travaux pratiques sur l'analyse du son, DEA MVA, ENS Cachan, 2002.
Agnès Desolneux, Travaux pratiques sur l'entropie et la théorie de l'information, préparation à l'agrégation, ENS Cachan, 2002.
Sylvie Fabre, Jean-Michel Morel, Yann Gousseau, Notes du cours d'analyse, ENS Cachan 1ère année, 2007.