5ETI — S9

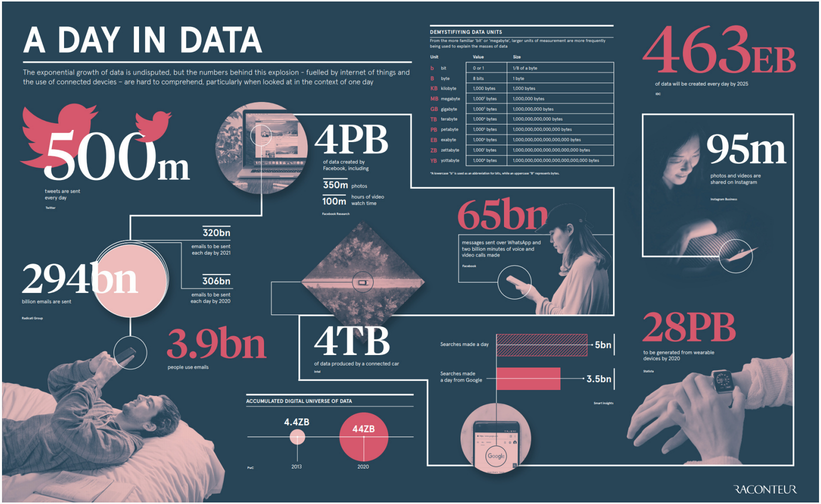
Compression et techniques avancées

Compression d'image

Éric Van Reeth CPE/CREATIS

Bureau B126A eric.van-reeth@cpe.fr

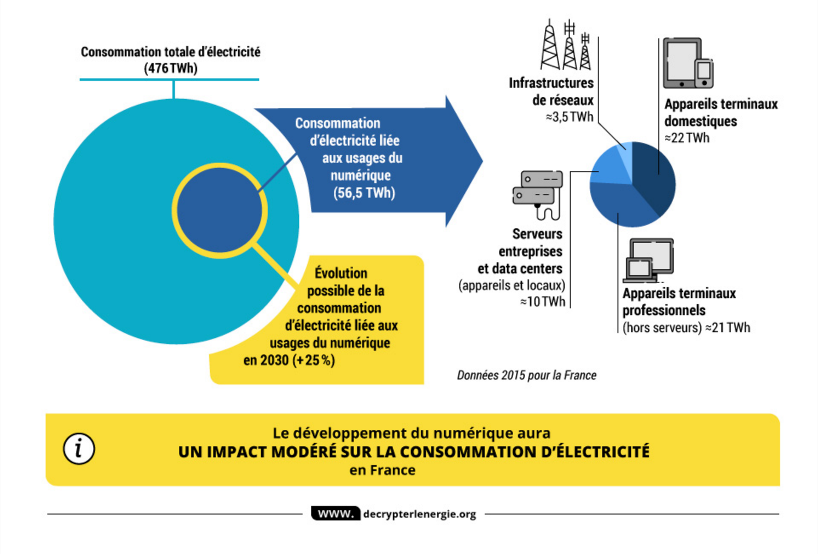
Contexte



Unit		Value	Size
b	bit	0 or 1	1/8 of a byte
B	byte	8 bits	1 byte
KB	kilobyte	1,000 bytes	1,000 bytes
MB	megabyte	$1{,}000^2$ bytes	1,000,000 bytes
GB	gigabyte	$1{,}000^3$ bytes	1,000,000,000 bytes
TB	terabyte	$1{,}000^4$ bytes	1,000,000,000,000 bytes
PB	petabyte	$1{,}000^5$ bytes	1,000,000,000,000,000 bytes
EB	exabyte	$1{,}000^6$ bytes	1,000,000,000,000,000,000 bytes
ZB	zettabyte	$1{,}000^7$ bytes	1,000,000,000,000,000,000,000 bytes
YB	yottabyte	$1{,}000^8$ bytes	1,000,000,000,000,000,000,000,000 bytes

*A lowercase "b" is used as an abbreviation for bits, while an uppercase "B" represents bytes.

Contexte


Consommation totale d'électricité
(476 TWh)
Consommation d'électricité liée aux usages du numérique (56,5 TWh)
Évolution possible de la consommation d'électricité liée aux usages du numérique en 2030 (+25 %)
Infrastructures de réseaux ≈3,5 TWh
Appareils terminaux domestiques ≈22 TWh
Serveurs entreprises et data centers (appareils et locaux) ≈10 TWh
Appareils terminaux professionnels (hors serveurs) ≈21 TWh
Données 2015 pour la France
Le développement du numérique aura
UN IMPACT MODÉRÉ SUR LA CONSOMMATION D'ÉLECTRICITÉ
en France
www. decrypterlenergie.org

Théorie de l'information

Code binaire

Le code binaire est le support numérique le plus utilisé pour stocker ou transmettre de l'information

L'élément de base du code binaire est le **bit** : 0 ou 1

Une succession de 8 bits forme un **octet**, unité de base du stockage en mémoire :

$$
\begin{aligned}
1\,\text{Mo} &= 10^{6}\ \text{octets}\\
1\,\text{Go} &= 10^{9}\ \text{octets}\\
1\,\text{To} &= 10^{12}\ \text{octets}\\
1\,\text{Po} &= 10^{15}\ \text{octets}\\
1\,\text{Eo} &= 10^{18}\ \text{octets}\\
1\,\text{Zo} &= 10^{21}\ \text{octets}
\end{aligned}
$$

Exercice : codage uniforme

Soit le message suivant : $\{0, 1, 1, 3, 2, 1\}$

1. Combien de bits sont nécessaires pour coder ce message ?
2. Donner le code binaire du message

Exercice : codage à longueur variable

Soit le message suivant : $\{0, 1, 1, 3, 2, 1\}$

Considérons l'assignation suivante :

$$0 \mapsto 001 \qquad 1 \mapsto 1 \qquad 2 \mapsto 01 \qquad 3 \mapsto 000$$

1. Donner le code binaire du message
2. Quel est le taux de compression obtenu par rapport au code précédent ?

Notions de base

Nombre de bits moyen

On note $\bar{L}$, le nombre de bits moyen permettant de représenter les symboles d'un message :

$$\bar{L} = \sum_{n=0}^{N-1} l(r_n)p(r_n)$$

où $l(r_n)$ et $p(r_n)$ désignent respectivement la longueur de code et la probabilité d'occurrence du symbole r_n.

Source d'information

Processus de génération d'une série de variables aléatoires :

$$\ldots, X_{k-1}, X_k, X_{k+1}, \ldots$$

Elle est dite **sans mémoire** si les variables sont statistiquement indépendantes les unes des autres

La source est associée à un **alphabet** : ensemble des symboles

On associe à chaque symbole sa probabilité d'apparition

Quantification de l'information

Quantité d'information

La quantité d'information apportée par un nouvel évènement s'exprime tel que :

$$I(r_k) = -\log_m(p(r_k))$$

L'unité associée à cette grandeur dépend de m, la base du logarithme utilisée :

bits pour $m = 2$

nats pour $m = e$

dits pour $m = 10$

Quantification de l'information

Entropie

L'entropie est la quantité moyenne d'information d'une source aléatoire :

$$H = -\sum_{n=0}^{N-1} p(r_n) \log_m(p(r_n))$$

Elle s'exprime généralement en bits ($m = 2$)

Par convention, pour une probabilité nulle on aura :

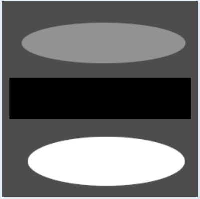
$$0 \log_2(0) = 0$$

Exercice : entropie

Calculer l'entropie du message suivant : $\{0, 1, 1, 3, 2, 1\}$

Exercice : entropie d'une image

Donner l'entropie de l'image suivante de taille (256×256) :



Le rectangle noir contient 13107 pixels

La zone ovale grise contient 8520 pixels

La zone ovale blanche contient 9830 pixels

Quantification de l'information

Borne de Shannon

Le **nombre de bits moyen** nécessaire pour le codage d'une source (sans mémoire) **ne peut être inférieur à son entropie** :

$$H \leq \bar{L}$$

Quantification de l'information ⚙

Exercice : entropie minimale et maximale

1. Quelle est l'entropie minimale qu'une image peut avoir ? Donner l'histogramme de l'image correspondante.
2. Quelle est l'entropie maximale qu'une image peut avoir ? Donner l'histogramme de l'image correspondante.

Codage entropique

Codage de Huffman

- Introduit en 1952 pour réduire la redondance de codage d'une source (sans perte)
- Nécessite la connaissance des probabilités de la source
- Converge vers le code optimal pour une source sans mémoire infiniment longue

Codage de Huffman

Étape 1 : réduction de source

1. Tri des symboles par probabilité décroissante
2. Regroupement des symboles de probabilités les plus faibles
3. Itération du processus jusqu'à réduction de la source à 2 symboles

Étape 1 : réduction de source

Symboles	**Probas**	**1**	**2**	**3**	**4**
r_1	0.4	0.4	0.4	0.4	0.6
r_5	0.3	0.3	0.3	0.3	0.4
r_0	0.1	0.1	0.2	0.3	
r_3	0.1	0.1	0.1		
r_2	0.06	0.1			
r_4	0.04				

Codage de Huffman

Étape 2 : codage des symboles

1. Attribution d'un code à chacun des symboles de la dernière réduction
2. Décomposition du symbole agrégé à l'étape précédente et attribution d'un symbole à chacun
3. Itération du processus jusqu'à attribution de chaque symbole original

Codage de Huffman

Étape 2 : codage des symboles

Symboles	**Probas**	**1**	**2**	**3**	**4**
r_1	0.4 1	0.4 1	0.4 1	0.4 1	0.6 **0**
r_5	0.3 00	0.3 00	0.3 00	0.3 0**0**	0.4 **1**
r_0	0.1 011	0.1 011	0.2 01**0**	0.3 0**1**	
r_3	0.1 0100	0.1 010**0**	0.1 01**1**		
r_2	0.06 0101**0**	0.1 010**1**			
r_4	0.04 0101**1**				

Exercice : codage du Huffman

Considérons l'image suivante :

21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243

1. Calculer l'entropie de l'image.
2. Compresser l'image en utilisant le codage de Huffman, et donner la longueur moyenne du code.
3. Quel taux de compression est atteint par rapport au codage uniforme ?

Codage de Huffman

Exercice : codage du Huffman (suite)

Considérons l'image suivante :

21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243

4. Considérons l'encodage des différences entre pixels voisins de l'image. Donner le nouveau code de Huffman obtenu et sa longueur moyenne.
5. Calculer l'entropie de l'image de différences. Comment expliquer la différence par rapport à l'entropie de l'image ?

Décodage

Décodage unique (pas d'ambiguïté possible)

La correspondance symbole/code doit être transmise pour le décodage

Décoder le message suivant à partir de la correspondance de la figure précédente :

01100010011101011000101100

Principe général

- Nécessite la connaissance des probabilités des symboles
- Il attribue **un code à une suite de symboles** plutôt qu'un code par symbole
- Le message est codé sous la forme d'un **nombre décimal** compris dans l'intervalle [0, 1[

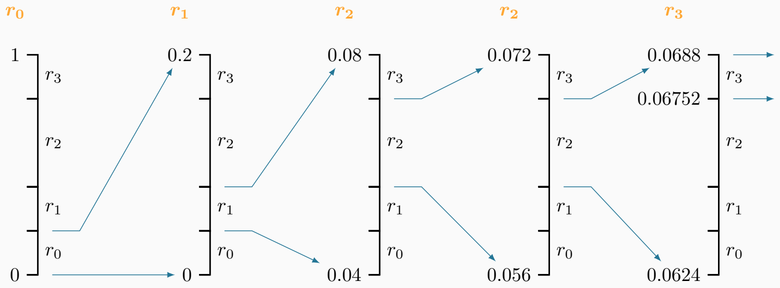
Principe de codage

- Découpage de l'intervalle $[0, 1[$ en autant de sous-intervalles qu'il y a de symboles à coder
- La dimension des sous-intervalles est liée à la probabilité du symbole :

Symboles	Probas	Intervalles
r_0	0.2	$[0.0, 0.2[$
r_1	0.2	$[0.2, 0.4[$
r_2	0.4	$[0.4, 0.8[$
r_3	0.2	$[0.8, 1.0[$

Principe de codage

Restriction successive de l'intervalle à l'intervalle du symbole à coder :



Codage arithmétique

Principe de codage

Le message entier est codé en choisissant un nombre décimal compris dans l'intervalle défini par le dernier symbole

Dans l'exemple précédent, on choisira par exemple 0.068

Le code final contient 3 décimales (soit $3/5 = 0.6$ décimale par symbole)

$$H = -\left(3 \times 0.2 \log_{10}(0.2) + 0.4 \log_{10}(0.4)\right) = 0.58\,\text{dits}$$

Principe de décodage

- Nécessite de conserver les probabilités des symboles
- Le premier symbole est celui dont l'intervalle contient le code final
- Mise à jour du code avec la formule :

$$c^{k+1} = \frac{c^k - b_k}{p_k}$$

- b_k : borne inférieure de l'intervalle du symbole r_k
- p_k : probabilité du symbole r_k

Codage arithmétique

Principe de décodage

1. Code final : $c = 0.068 \in [0, 0.2[$
   Le symbole 1 est donc : r_0
2. Code actualisé : $c = (0.068 - 0)/0.2 = 0.34 \in [0.2, 0.4[$
   Le symbole 2 est donc : r_1
3. Code actualisé : $c = (0.34 - 0.2)/0.2 = 0.7 \in [0.4, 0.8[$
   Le symbole 3 est donc : r_2
4. Code actualisé : $c = (0.7 - 0.4)/0.4 = 0.75 \in [0.4, 0.8[$
   Le symbole 4 est donc : r_2
5. Code actualisé : $c = (0.75 - 0.4)/0.4 = 0.875 \in [0.8, 1[$
   Le symbole 5 est donc : r_3

Exercice : codage arithmétique

Considérons une source composée de 4 symboles $\{a, b, c, d\}$, de probabilités $\{0.1, 0.4, 0.3, 0.2\}$

Donner le codage arithmétique du message suivant :

$$bbadc$$

Exercice : décodage arithmétique

Considérons la source suivante suivante :

Symboles	Probas
a	0.2
e	0.3
i	0.1
o	0.2
u	0.1
!	0.1

Décoder le message suivant :

$$0.23355$$

Remarques générales

- Le codage arithmétique est typiquement utilisé pour coder une série de symboles
- Nécessité d'un indicateur de fin de message qui fait baisser l'efficacité de la méthode
- Limitation du fait de la précision numérique des systèmes
- Pas toujours évident de connaître les probabilités des symboles (utilisation d'a priori)

Réduction de la redondance spatiale

Constat

La plupart des images naturelles présentent une forte redondance spatiale



Codage par plages

Principe

Introduit dans les années 1950

Permet de coder efficacement de **longues plages de répétition** du même symbole

Ces plages sont compressées grâce à une paire de symboles :

1. le premier symbole représentant **la valeur du symbole** (i.e. l'intensité d'un pixel)
2. le deuxième symbole représentant **le nombre de répétitions** consécutives de ce symbole

Réduction de la redondance spatiale

Codage par plages

Codage par plages

Exemple : codage par plages

Soit un message de 17 symboles à encoder :

$$AAABBAAAAACAABBBB$$

Codage par plages

Exemple : codage par plages

Soit un message de 17 symboles à encoder :

$$AAABBAAAAACAABBBB$$

Son codage par plage sera composé des 12 symboles suivants :

$$A3B2A5C1A2B4$$

Application au cas binaire

- Régulièrement utilisée pour coder des messages binaires
- Forte chance d'avoir des plages de répétitions importantes
- On code uniquement la longueur des chaînes de 0 et de 1
- On considère que le message démarre par 1

Exercice : codage par plages

Coder le message binaire suivant :

0111100111111100000001000 1

Codage LZW

Principe

- Proposé en 1984 par Lempel, Ziv et Welch
- Fait partie des méthodes de codage par dictionnaire (ne nécessite pas la connaissance des probabilités des symboles)
- Un dictionnaire initial contenant les symboles de la source est établi
- Ce dictionnaire sera étendu avec chaque nouvel enchaînement de symbole rencontré

Codage LZW

Exemple :

Considérons la région d'une image codée sur 8 bits, contenant un contour vertical :

$$\begin{pmatrix} 20 & 20 & 100 & 100 \\ 20 & 20 & 100 & 100 \\ 20 & 20 & 100 & 100 \\ 20 & 20 & 100 & 100 \end{pmatrix}$$

Utilisation d'un dictionnaire de 512 éléments :

Adresse	0	1	...	255	256	...	511
Code	0	1	...	255	-	...	-

Codage LZW

Exemple :

Pixel courant	Séq. Reconnue	Code	Nouveau code	Adresse
20	20	**20**	20-20	256
20	20	**20**	20-100	257
100	100	**100**	100-100	258
100	100	**100**	100-20	259
20	20-20	**256**	20-20-100	260
100	100-100	**258**	100-100-20	261
20	20-20-100	**260**	20-20-100-100	262
100	100-20	**259**	100-20-20	263
20	20-100	**257**	20-100-100	264
100	100	**100**	-	-

Codage LZW

Exemple :

Code transmis :

20 20 100 100 256 258 260 259 257 100

Composé de 10 symboles de 9 bits = 90 bits

Initialement 16 symboles de 8 bits = 128 bits

Taux de compression obtenu : 1.4

Codage LZW

Remarques

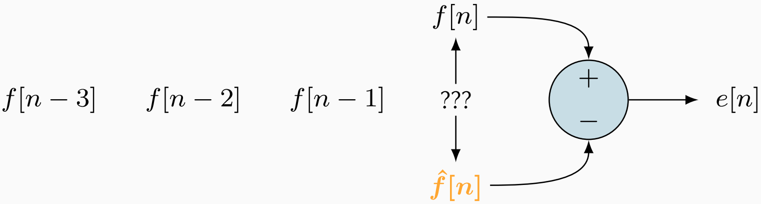
- La transmission du dictionnaire n'est pas nécessaire au décodage
- Stratégies de la gestion du dictionnaire :
  - Réinitialisation du dictionnaire une fois plein
  - Réinitialisation du dictionnaire lorsque il ne permet plus une compression efficace, quand la redondance spatiale devient faible.
  - Remplacement de l'élément du dictionnaire le moins fréquemment utilisé ou utilisé il y a le plus longtemps

Réduction de la redondance spatiale

Codage prédictif

Principe

- Technique avec ou sans perte
- **Encodage de la nouvelle information** présente dans le pixel courant



Codage prédictif sans perte

Schéma d'encodage

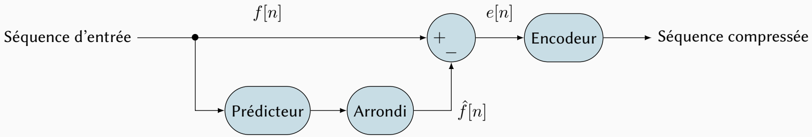
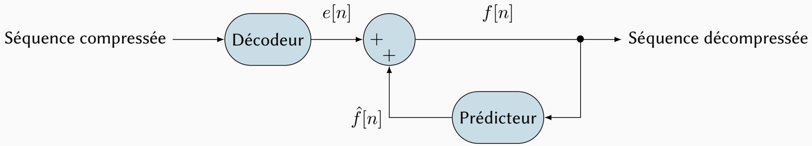


Schéma de décodage



Codage de l'erreur de prédiction

L'arrondi permet de limiter la taille du dictionnaire de $e[n]$

L'erreur est codée par codage entropique

$\rightarrow$ Histogramme de $e[n]$ centré en 0

$\rightarrow$ Faible entropie et faible corrélation

$\rightarrow$ Optimisation du codage entropique

Codage prédictif sans perte

Prédiction

En tenant compte de l'arrondi, le prédicteur s'écrit :

$$\hat{f}[n] = \mathrm{int}\left[\sum_{i=1}^{m} b[i] f[n-i]\right]$$

On parle de **filtrage prédictif** d'ordre m

Si $f[n]$ est aléatoire, $e[n]$ est un bruit blanc de variance σ_e^2

Codage prédictif sans perte

Mise en œuvre

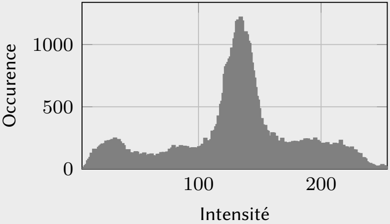
- Pour le codage d'une image/vidéo, les pixels voisins sont utilisés pour la prédiction (ligne, colonnes et trames voisines)
- Les m premiers éléments de $f[n]$ sont codés et transmis directement
- À la réception, la séquence originale est retrouvée en réalisant la **même prédiction** :

$$f[n] = e[n] + \hat{f}[n]$$

Codage prédictif sans perte

Exemple : codage prédictif sans perte

Considérons l'encodage de l'image suivante :



Entropie de l'image : $H_I = 7.4\,\text{bits}$

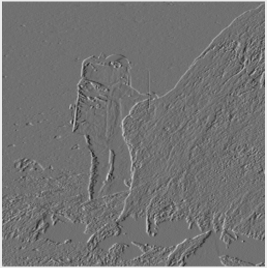
Codage prédictif sans perte

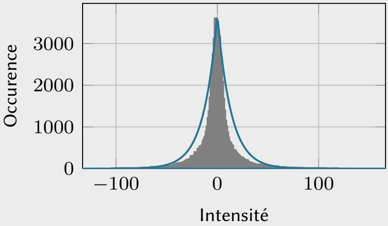
Exemple : codage prédictif sans perte

Considérons le prédicteur d'ordre 1 suivant :

$$\hat{f}[i,j] = f[i,j-1]$$

Erreur de prédiction et son histogramme :





Exemple : codage prédictif sans perte

Entropie de l'image d'erreur : $H_e = 5.9$ bits

Gain moyen de 1.5 bits par pixel $\rightarrow$ taux de compression de 1.23 par rapport au codage uniforme

Modélisation de l'erreur de prédiction par une distribution Laplacienne de moyenne nulle :

$$p_e(e) = \frac{1}{\sqrt{2}\sigma_e} e^{-\frac{\sqrt{2}|e|}{\sigma_e}}$$

Codage prédictif avec perte

Objectif

- Améliorer le taux de compression
- Suppression de l'étape d'arrondi
- Insertion d'une étape de quantification de l'erreur $\rightarrow$ perte d'information
- On notera $\dot{e}[n]$ et $\dot{f}[n]$ l'erreur et la séquence quantifiées

Codage prédictif avec perte

Schéma d'encodage

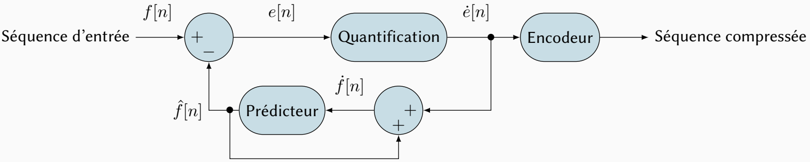
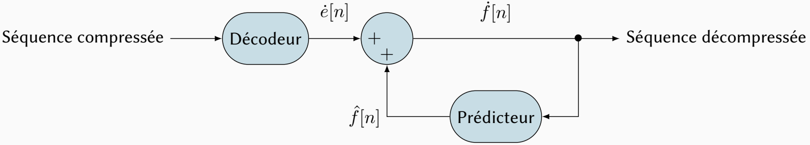


Schéma de décodage



Codage prédictif avec perte

Fonctionnement

L'erreur quantifiée $\dot{e}[n]$ est restreinte à un nombre fini de valeurs

La séquence quantifiée tient compte de la quantification :

$$\dot{f}[n] = \dot{e}[n] + \hat{f}[n]$$

Même formule au décodage $\rightarrow$ pas de propagation des erreurs lors du décodage

Codage prédictif avec perte

Modulation delta

Forme simple et commune de codage prédictif avec perte

L'étape de prédiction est donnée par :

$$\hat{f}[n] = \sum_{i=1}^{m} b[i]\dot{f}[n-i]$$

La quantification consiste à attribuer deux valeurs à $e[n]$:

$$\dot{e}[n] = \begin{cases} +\delta & \text{si } e[n] > 0 \\ -\delta & \text{sinon} \end{cases}$$

avec $\delta \in \mathbb{R}^{+*} \rightarrow$ codage de $\dot{e}[n]$ **sur 1 bit**

Codage prédictif avec perte

Exemple : modulation delta

Codage prédictif d'ordre 1 d'une séquence $f[n]$ avec $b[1] = 1$

À partir de l'indice $n = 1$, on peut évaluer :

$$\hat{f}[1] = \dot{f}[0] \;\rightarrow\; e[1] = 1 \;\rightarrow\; \dot{e}[1] = +\delta \;\rightarrow\; \dot{f}[1] = \hat{f}[1] + \delta$$

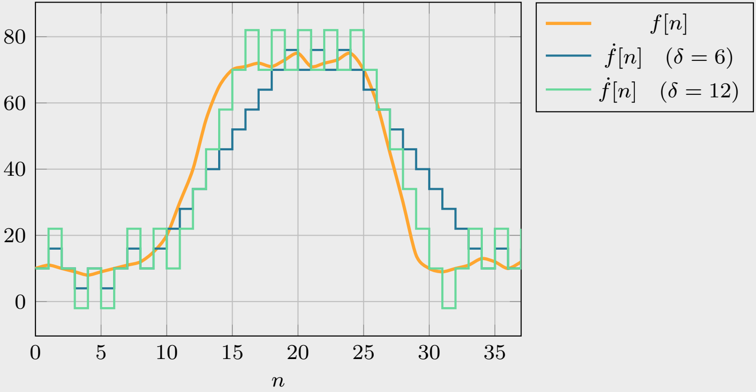
Le tableau suivant détaille les séquences caractéristiques du processus d'encodage pour $\delta = 12$:

n	$f[n]$	$\hat{f}[n]$	$e[n]$	$\dot{e}[n]$	$\dot{f}[n]$
0	10	-	-	-	10
1	11	10	1	12	22
2	10	22	-12	-12	10
3	9	10	-1	-12	-2
4	8	-2	-10	12	10
...	...	...	...	...	...

Codage prédictif avec perte

Exemple : modulation delta

Résultats pour δ = 6 et δ = 12



Codage prédictif optimal

Objectif

Minimiser la variance de l'erreur de prédiction en optimisant l'étape de prédiction

On définit l'erreur quadratique de prédiction telle que :

$$
\begin{aligned}
E\left\{e^2[n]\right\} &= E\left\{\left(f[n]-\hat{f}[n]\right)^2\right\} \\
&= E\left\{\left(f[n]-\sum_{i=1}^{m} b[i] f[n-i]\right)^2\right\}
\end{aligned}
$$

On cherche les m coefficients $b[i]$ optimaux (DPCM pour *differential pulse code modulation*)

Codage prédictif optimal

Principe

- Synthèse d'un **filtrage optimal (Wiener)** (TSA-TP4) mettant en œuvre le **principe d'orthogonalité**
- La prédiction du filtre est optimale lorsque l'erreur de prédiction est orthogonale à la séquence d'entrée :

$$\forall n, k > 1, \qquad E\left\{e[n]f[n-k]\right\} = 0$$

Codage prédictif optimal

Calcul des coefficients

Utilisation de l'autocorrélation de $f[n]$ pour exprimer les coefficients optimaux :

$$
\begin{aligned}
& E\left\{\left(f[n]-\hat{f}[n]\right)f[n-k]\right\} = 0 \\
\Leftrightarrow\quad & E\{f[n]f[n-k]\} - E\left\{\hat{f}[n]f[n-k]\right\} = 0 \\
\Leftrightarrow\quad & \gamma_f[k] - E\left\{\sum_{i=1}^{m} b[i]f[n-i]f[n-k]\right\} = 0 \\
\Leftrightarrow\quad & \gamma_f[k] - \sum_{i=1}^{m} b[i]E\{f[n-i]f[n-k]\} = 0 \\
\Leftrightarrow\quad & \gamma_f[k] = \sum_{i=1}^{m} b[i]\gamma_f[k-i]
\end{aligned}
$$

Codage prédictif optimal

Calcul des coefficients

En injectant l'expression connue de la puissance de l'erreur de prédiction donnée par :

$$\sigma_e^2 = \gamma_f[0] - \sum_{i=1}^{m} b[k]\gamma_f[k]$$

Les coefficients de filtre optimaux peuvent être obtenus à partir du système suivant :

$$\begin{pmatrix} \gamma_f[0] & \gamma_f[1] & \dots & \gamma_f[m] \\ \gamma_f[1] & \gamma_f[0] & \dots & \gamma_f[m-1] \\ \vdots & \vdots & \vdots & \vdots \\ \gamma_f[m] & \dots & \dots & \gamma_f[0] \end{pmatrix} \begin{pmatrix} 1/\sigma_e^2 \\ -b[1]/\sigma_e^2 \\ \vdots \\ -b[m]/\sigma_e^2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Codage prédictif optimal

Application pratique

- Complexité algorithmique accrue (calcul de $\boldsymbol{R}$, $\boldsymbol{R^{-1}}$, γ_f) au codage et décodage
- Gain de compression peu évident au delà de $m = 3$
- Prise en compte du voisinage 2D peu évident
- Utilisation de coefficients génériques (globaux) estimés à partir de modèles d'images basés sur des sources de Markov 2D

Codage prédictif optimal

Exemple : prédiction optimale d'une ligne d'une image

Évolution de la puissance du bruit de prédiction et de l'entropie associée en fonction de l'ordre du filtre :