Nicolas Sendrier

Majeure d'informatique

Introduction à la théorie de l'information

Cours n°3

Algorithmes de codage de source

Codage et décodage à l'aide d'un code préfixe

Codage : Accès à une table indexée par les lettres.

Décodage : Parcours dans l'arbre du code.

- On part de la racine de l'arbre.
- À chaque bit lu de la séquence à décoder on descend à droite ou à gauche suivant sa valeur.
- Lorsque l'on atteint une feuille, on obtient une lettre du message et on retourne à la racine.

Code optimal

Les théorèmes de Kraft et Mac Millan ont un corollaire immédiat :
Corollaire S'il existe un code à décodage unique dont les K mots ont pour longueur $n_1, n_2, \ldots, n_K$ alors il existe un code préfixe avec les mêmes longueurs.

Définition Un code à décodage unique d'une source X est *optimal* s'il n'existe pas de code à décodage unique de X ayant une longueur moyenne strictement inférieure.

Proposition Pour toute source il existe un code préfixe optimal.

Code de Huffman

Soit la source discrète X d'alphabet $\mathcal{X} = \{a_1, \ldots, a_{K-2}, a_{K-1}, a_K\}$ munie de la loi P. Sans perdre de généralité, nous pouvons supposer que $P(a_1) \geq \ldots \geq P(a_{K-1}) \geq P(a_K) > 0$.

Nous définissons la source Y d'alphabet $\mathcal{Y} = \{a_1, \ldots, a_{K-2}, b_{K-1}\}$ munie de la loi

$$\begin{aligned} Q(a_k) &= P(a_k), & k = 1 \ldots, K-2 \\ Q(b_{K-1}) &= P(a_{K-1}) + P(a_K) \end{aligned}$$

Algorithme (Huffman) Nous voulons construire φ un code préfixe de X. Si $K = 2$, les mots de code sont $\varphi(a_1) = 0$ et $\varphi(a_2) = 1$. Si $K > 2$, soit ψ un code de Huffman de Y,
- $\varphi(a_k) = \psi(a_k)$ pour $k = 1 \ldots, K-2$,
- $\varphi(a_{K-1}) = \psi(b_{K-1}) \,\|\, 0$,
- $\varphi(a_K) = \psi(b_{K-1}) \,\|\, 1$,

Le code de Huffman est optimal

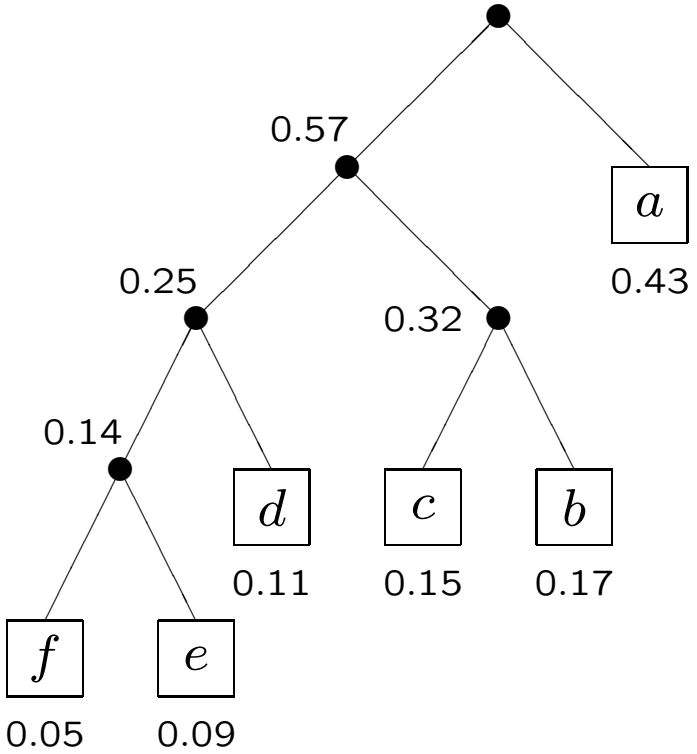
Lemme Il existe un code préfixe optimal dans lequel les deux lettres les moins probables sont codées par des mots de longueur maximale. Ces mots sont identiques sauf pour leur dernier symbole.

Ces deux mots de codes sont donc de la forme $(m \parallel 0)$ et $(m \parallel 1)$.

Lemme Si ψ est un code optimal de Y alors φ est un code optimal de X (notations de l'algorithme de Huffman).

Proposition Le code de Huffman est optimal.

Code de Huffman : exemple



x	$P(x)$	$-\log_2(P(x))$	$\varphi(x)$	n_x
a	0.43	1.22	0	1
b	0.17	2.56	100	3
c	0.15	2.74	101	3
d	0.11	3.18	110	3
e	0.09	3.47	1110	4
f	0.05	4.32	1111	4

$H = 2.248$ $\quad \bar{n} = 2.28$

$E = 98.6\%$

Nombres 2-adiques

Dans l'intervalle $[0,1]$, ils sont de la forme

$$\sum_{i=1}^{\infty} d_i\, 2^{-i}, \text{ où } d_i \in \{0,1\}$$

On écrira $0.d_1 d_2 d_3 \ldots$

Par exemple

0.25	$\rightarrow$	0.01	0.43	$\rightarrow$	0.0110111000...
0.125	$\rightarrow$	0.001	0.71	$\rightarrow$	0.1011010111...
0.625	$\rightarrow$	0.101	$1/\sqrt{2}$	$\rightarrow$	0.1011010100...

Certains nombres ont plusieurs développements, par exemple $0.25 \rightarrow 0.01000\ldots$ et $0.25 \rightarrow 0.00111\ldots$. Dans ce dernier cas on choisira le développement de valuation minimale (le plus court).

Code de Shannon-Fano-Elias

Soit X une source discrète d'alphabet $\mathcal{X}$ et de loi de probabilité P.

Nous supposons que $\mathcal{X}$ est muni d'une relation d'ordre total. Pour tout $x \in \mathcal{X}$, nous noterons* $l(x) = \lceil -\log_2 P(x) \rceil$, et nous définissons les fonctions suivantes (probabilités cumulées) :

$$S(x) = \sum_{x'<x} P(x') \quad \text{et} \quad \bar{S}(x) = \frac{1}{2}P(x) + S(x).$$

(*Pour la plus petite lettre* $S(x)$ *vaut 0*)

Nous définissons $\varphi(x)$ pour tout $x \in \mathcal{X}$ comme les $l(x) + 1$ premiers bits du développement 2-adique de $\bar{S}(x)$.

Proposition Le code φ est préfixe et sa longueur moyenne $\bar{n}$ vérifie

$$H(X) \leq \bar{n} < H(X) + 2$$

*$\lceil \cdot \rceil$ est la partie entière arrondie supérieurement

Code de Shannon-Fano-Elias : exemple

x	$P(x)$	$l(x)$	$\bar{S}(x)$		$\varphi(x)$	Huffman
a	0.43	2	0.215	0.0011011 ...	001	0
b	0.17	3	0.515	0.1000001 ...	1000	100
c	0.15	3	0.675	0.1010110 ...	1010	101
d	0.11	5	0.805	0.1100111 ...	11001	110
e	0.09	4	0.905	0.1110011 ...	11100	1110
f	0.05	5	0.975	0.1111100 ...	111110	1111

Code de Shannon-Fano-Elias : autre exemple

x	$P(x)$	$l(x)$	$\bar{S}(x)$		$\varphi(x)$	Huffman
a	0.25	2	0.125	0.001	001	10
b	0.5	1	0.5	0.1	10	0
c	0.125	3	0.8125	0.1101	1101	110
d	0.125	3	0.9375	0.1111	1111	111

Si on enlève le dernier bit à φ le code n'est plus préfixe.

x	$P(x)$	$l(x)$	$\bar{S}(x)$		$\varphi(x)$	Huffman
b	0.5	1	0.25	0.01	01	0
a	0.25	2	0.625	0.101	101	10
c	0.125	3	0.8125	0.1101	1101	110
d	0.125	3	0.9375	0.1111	1111	111

Si on enlève le dernier bit à φ le code reste préfixe.

Code de Shannon (?)

Le code de Shannon est défini de la même manière que celui de Shannon-Fano-Elias, à deux exceptions près :
– les lettres sont rangées par probabilités décroissantes,
– le mot codant x est constitué des $l(x)$ premiers bits de $S(x)$.
(En particulier, le mot codant la plus petite lettre est composé que de $l(x)$ '0').

Proposition Le code de Shannon est préfixe et sa longueur moyenne $\bar{n}$ vérifie

$$H(X) \leq \bar{n} < H(X) + 1$$

Preuves

Lemme Pour tous réels u et v dans $[0,1[$, et pour tout entier $l > 0$, si $|u - v| \geq 2^{-l}$ alors les l premiers bits des développements 2-adiques de u et v ne sont pas tous identiques.

Le code de Shannon-Fano-Elias est préfixe :
Pour tous $x, y \in \mathcal{X}$ tels que $l(y) \geq l(x)$

$$|\bar{S}(x) - \bar{S}(y)| \geq \frac{P(x)}{2} \geq \frac{1}{2^{l(x)+1}}$$

Le code de Shannon est préfixe :
Pour tous $x, y \in \mathcal{X}$ tels que $y > x$ (donc $l(y) \geq l(x)$)

$$|S(x) - S(y)| \geq P(x) \geq \frac{1}{2^{l(x)}}$$

Codage et décodage

Si l'alphabet $\mathcal{X}$ est de petite taille, on fait comme pour un code préfixe (aucun avantage par rapport à Huffman).

Si l'alphabet est grand, il faut pour le codage
- une fonction qui calcule $l(x)$ (c'est l'indice du premier '1' dans le développement 2-adique de $P(x)$),
- une fonction qui calcule $\bar{S}(x)$ (ou $S(x)$).

Pour le décodage
- une fonction qui calcule $l(x)$,
- une fonction qui calcule le + petit x tel que $\varphi(x) \leq \bar{S}(x)$ (ou $S(x)$). (*on identifie $\varphi(x)$ et le réel dont il est le développement 2-adique*)

Codage arithmétique

Nous allons coder la source d'alphabet $\mathcal{X}^L$ munie de la loi produit. Pour tout entier $n \leq L$, posons

$$\begin{aligned} l(x_1,\ldots,x_n) &= \lfloor -\log_2 P(x_1,\ldots,x_n) \rfloor \\ S(x_1,\ldots,x_n) &= \sum_{(y_1,\ldots,y_n)<(x_1,\ldots,x_n)} P(y_1,\ldots,y_n) \\ \bar{S}(x_1,\ldots,x_n) &= S(x_1,\ldots,x_n) + P(x_1,\ldots,x_n)/2 \end{aligned}$$

Cela nous permet d'utiliser le code de Shannon-Fano-Elias sur X^L.
Proposition L'efficacité de ce codage est $> 1 - \frac{2}{LH(X)}$.

Proposition On pose $S(x_n \mid x_1,\ldots,x_{n-1}) = \sum_{y_n<x_n} P(y_n \mid x_1,\ldots,x_{n-1})$,

$$S(x_1,\ldots,x_n) = S(x_1,\ldots,x_{n-1}) + S(x_n \mid x_1,\ldots,x_{n-1}) P(x_1,\ldots,x_{n-1})$$

Si la source est sans mémoire

$$S(x_1,\ldots,x_n) = S(x_1,\ldots,x_{n-1}) + S(x_n) \prod_{i=1}^{n-1} P(x_i)$$

Codage/Décodage

L'avantage du codage arithmétique est de permettre le codage et le décodage sans jamais utiliser explicitement l'arbre associé.

Nous codons la source X^L à l'aide du code de Shannon-Fano-Elias, que nous noterons φ_L. Pour pouvoir coder, il faut calculer le nombre réel $\bar{S}(x_1,\ldots,x_L)$, qui peut-être calculé récursivement (en L étapes) avec une précision de $l(x_1,\ldots,x_L)$ bits.

Soit $r=\varphi(x_1,\ldots,x_L)$ le nombre réel dont le développement 2-adique est le mot codant $(x_1,\ldots,x_L)\in\mathcal{X}^L$. Les lettres $x_1,\ldots,x_L$ sont les seules vérifiant

$$\begin{array}{rcccl} S(x_1) & < & r & < & S(x_1)+P(x_1) \\ S(x_1,x_2) & < & r & < & S(x_1,x_2)+P(x_1,x_2) \\ & & \vdots & & \\ S(x_1,\ldots,x_L) & < & r & < & S(x_1,\ldots,x_L)+P(x_1,\ldots,x_L) \end{array}$$

Décodage (suite)

On pose $r_0 = r$ la séquence codée. Pour tout $i > 0$, on note

$$r_i = \frac{r_{i-1} - S(x_i)}{P(x_i)}$$

Les lettres $x_1, \ldots, x_L$ sont les seules vérifiant

$$\begin{array}{ccccc} S(x_1) & < & r_0 & < & S(x_1) + P(x_1) \\ S(x_2) & < & r_1 & < & S(x_2) + P(x_2) \\ & & \vdots & & \\ S(x_L) & < & r_{L-1} & < & S(x_L) + P(x_L) \end{array}$$

On peut donc construire le décodeur du code arithmétique à partir d'un décodeur du code de Shannon-Fano-Elias de $\mathcal{X}$.

À condition de calculer les r_i avec une précision suffisante