Série 11 : Solutions

1 Calcul d'entropie

Les probabilités d'apparition des lettres dans la séquence "HASTA LA VISTA BABY !" (de longueur 20) sont les suivantes :

lettre	A	␣	S	T	B	H	L	V	I	Y	!
p_j	$\frac{5}{20}$	$\frac{3}{20}$	$\frac{2}{20}$	$\frac{2}{20}$	$\frac{2}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$

On a donc :

$$H(\mathcal{X}) = \frac{1}{4}\log_2(4) + \frac{3}{20}\log_2\left(\frac{20}{3}\right) + 3\cdot\frac{1}{10}\log_2(10) + 6\cdot\frac{1}{20}\log_2(20)$$

En utilisant $\log_2(4) = 2$, $\log_2(10) = 1 + \log_2(5)$, $\log_2(20) = 2 + \log_2(5)$ et $\log_2(20/3) = 2 + \log_2(5) - \log_2(3)$, on obtient après regroupement des termes :

$$H(\mathcal{X}) = \frac{17}{10} - \frac{3}{20}\log_2(3) + \frac{3}{4}\log_2(5)$$

Numériquement : $H(\mathcal{X}) \simeq 1.7 - 0.15\cdot 1.58 + 0.75\cdot 2.32 \simeq 3.20$.

2 Un peu de magie noire

a) Calcul de l'entropie de la séquence (que l'on appelle $\mathcal{X}$ comme dans le cours) : le A a une probabilité d'apparition de $\frac{5}{12}$, le V et le D de $\frac{2}{12}$ et les 3 lettres restantes de $\frac{1}{12}$. Donc

$$\begin{aligned} H(\mathcal{X}) &= \frac{5}{12}\log_2\left(\frac{12}{5}\right) + 2\,\frac{2}{12}\log_2\left(\frac{12}{2}\right) + 3\,\frac{1}{12}\log_2(12) \\ &= \log_2(12) - \frac{5}{12}\log_2(5) - \frac{1}{3}\log_2(2) = \frac{5}{3} + \log_2(3) - \frac{5}{12}\log_2(5) \end{aligned}$$

ce qui donne numériquement : $H(\mathcal{X}) \simeq 2.28$.

b) Avec l'algorithme de Shannon-Fano, on peut trouver les dictionnaires suivants :

lettre	nb app.	1) nb Q	mot de code	2) nb Q	mot de code	3) nb Q	mot de code
A	5	2	11	2	11	1	1
V	2	2	10	2	10	3	011
D	2	3	011	2	01	3	010
K	1	3	010	3	001	3	001
E	1	3	001	4	0001	4	0001
R	1	3	000	4	0000	4	0000

Avec les dictionnaires 1) et 2), la séquence codée contient 29 bits ; avec le dictionnaire 3), la séquence codée contient 28 bits.

c) Avec l'algorithme de Huffman, on trouve le code suivant :

lettre	nb apparitions	mot de code
A	5	1
V	2	011
D	2	010
K	1	001
E	1	0001
R	1	0000

et la séquence codée contient 28 bits. On peut aussi trouver des dictionnaires différents suivant l'arbre que l'on construit, mais la longueur de la séquence codée reste invariablement de 28 bits dans ce cas.

d) Avec l'algorithme de Shannon-Fano, la longueur moyenne du code $L(C_{SF}) \simeq 2.42$ ou 2.33. Avec l'algorithme de Huffman, la longueur moyenne du code $L(C_H) \simeq 2.33$, et l'entropie de la séquence vaut $H \simeq 2.28$. On vérifie donc bien les inégalités du cours :

$$H(\mathcal{X}) \leq L(C_H) \leq L(C_{SF}) \leq H(\mathcal{X}) + 1$$

3 De bien mauvais dicos

Explorons toutes les raisons pour lesquelles les minions ont failli à trouver de bons dictionnaires :

a) Le dictionnaire proposé ici n'utilise même pas un code binaire, mais un code "unaire". L'encodage du mot BANANA avec ce dictionnaire donne 1111111111, autrement dit quelque chose de tout à fait illisible ! Il faudrait rajouter des caractères de séparation entre les mots de code si on voulait y comprendre quelque chose.

b) Ce dictionnaire est d'un côté celui qui utilise le moins de bits (7) pour représenter le mot BANANA, mais le code binaire ainsi généré n'est pas sans préfixe et est donc difficilement décodable : en particulier, quand on lit 11, on ne sait jamais si on doit interpréter ça comme B ou NN. De manière correspondante, la longueur moyenne du code binaire utilisé ici est $\frac{7}{6}$, qui est en dessous de la limite de Shannon (= l'entropie du mot $\simeq 1.46$; voir calcul ci-dessous).

c) Ici, le code est sans préfixe et donc décodable de manière unique, mais la longueur moyenne du code $L(C) = \frac{10}{6} = \frac{5}{3}$ n'est pas optimale ; le mot de code associé à la lettre B est en fait inutilement trop long.

d) Ce dictionnaire utilise un code sans préfixe (du fait que tous les mots de code ont la même longueur) et est donc décodable de manière unique, mais il n'atteint pas la performance optimale, car sa longueur moyenne $L(C) = 2$ (forcément, puisque tous les mots de code sont de longueur 2) est plus éloignée de l'entropie du mot que nécessaire.

e) Le code utilisé ici est aussi sans préfixe, donc décodable de manière unique, mais le mot de code court (1) est attribué à la lettre la moins fréquente. La longueur moyenne du code résultante vaut $L(C) = \frac{11}{6} \simeq 1.88$, qui est loin d'être optimale.

f) Ce code cumule deux défauts : sa longueur moyenne n'est pas optimale et il n'est pas sans préfixe (donc difficilement décodable).

Finalement, un bon dictionnaire est le suivant (qu'on obtient indifféremment par l'algorithme de Shannon-Fano ou de Huffman) :

A	N	B
0	10	11

Il utilise 9 bits en tout pour la représentation du mot BANANA et sa longueur moyenne vaut donc $L(C) = \frac{9}{6} = \frac{3}{2}$, qui est très proche de l'entropie du mot :

$$H(X) = \frac{1}{2}\log_2(2) + \frac{1}{3}\log_2(3) + \frac{1}{6}\log_2(6) = \frac{2}{3} + \frac{1}{2}\log_2(3) \simeq 1.46$$

4 À la recherche d'un trésor

a) On doit chaque fois se mettre au point sur la grille qui est au milieu des possibilités qui nous restent, de sorte à diviser cet ensemble de possibilités en quatre parties égales ; vu qu'il y a 64 possibilités au départ, le nombre de questions à poser est $\log_4(64) = 3$.

b) (5,5), (7,7), (6,8) (et la suite des réponses correspondantes de l'oracle est NE, NO, SE).

c) Deux possibilités :

• En vous basant sur la suite des questions ci-dessus, vous encodez chaque réponse obtenue avec 2 bits, p.ex. : NE=00, NO=01, SE=10, SO=11 (on peut aussi dire N=0, S=1, O=0, E=1, en se rappelant alors qu'on indique d'abord la direction nord-sud avant la direction est-ouest). Dans le cas présent, la séquence est donc 000110.

• Sans rapport avec le jeu des questions ci-dessus, on peut aussi simplement encoder la position de la case avec 6 bits également : 3 bits pour la position horizontale (de 1 à 8 : plus précisément, on va encoder le nombre moins 1, comme dans l'exercice sur le codage par plages : ainsi, 000 encode 1, 001 encode 2, jusqu'à 111 qui encode 8), et 3 bits pour la position verticale. Avec cette représentation, la position du trésor ci-dessus est (6,7), et donc l'encodage est 101110 (on utilise la convention que l'abscisse vient avant l'ordonnée : pas besoin donc d'utiliser un bit pour ça : ça fait partie de la manière dont on définit notre dictionnaire).

5 Questions d'examens passés

a)

Une source X émet les symboles suivants :

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

avec les probabilités respectives :

$$\left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right\}$$

On note : $H(X)$ l'entropie de la source, $L_H(X)$ la longueur moyenne du code de Huffman et $L_{\text{SF}}(X)$ la longueur moyenne du code Shannon–Fano.

Quelle est la valeur de $L_H(X)$?

- ☐ 1.5
- ☐ 1.625
- ✓ 1.75
- ☐ 2.0

Quelle relation est correcte ?

- ☐ $H(X) < L_H(X) = L_{\text{SF}}(X)$
- ✓ $H(X) = L_H(X) = L_{\text{SF}}(X)$
- ☐ $H(X) = L_H(X) < L_{\text{SF}}(X)$
- ☐ $H(X) < L_H(X) < L_{\text{SF}}(X)$

Solution

Le code de Huffman optimal est : A $\longrightarrow$ `0` (1 bit), B $\longrightarrow$ `10` (2 bits), C $\longrightarrow$ `110` (3 bits), D $\longrightarrow$ `111` (3 bits). D'où :

$$L_H(X) = 1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 3 \cdot \frac{1}{8} + 3 \cdot \frac{1}{8} = \frac{1}{2} + \frac{1}{2} + \frac{3}{8} + \frac{3}{8} = \frac{7}{4} = \mathbf{1.75}.$$

Comme les probabilités sont toutes des puissances de 2, l'entropie vaut exactement :

$$H(X) = 1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 3 \cdot \frac{1}{8} + 3 \cdot \frac{1}{8} = 1.75.$$

L'algorithme de Shannon–Fano donne le même arbre ici, donc $L_{\text{SF}}(X) = 1.75$ également. Réponse : $H(X) = L_H(X) = L_{\text{SF}}(X)$.

b) Pour une source utilisant un alphabet de taille n, l'entropie $H(X)$ vérifie $0 \leq H(X) \leq \log_2 n$.

- ✓ VRAI
- ☐ FAUX

Solution

VRAI. Par définition $H(X) = \sum_j p_j \log_2(1/p_j) \geq 0$ (chaque terme est positif ou nul). Le maximum $\log_2 n$ est atteint lorsque tous les symboles sont équiprobables ($p_j = 1/n$ pour tout j).

c) La longueur moyenne d'un code de Huffman est strictement inférieure à l'entropie de la source.

- ☐ VRAI
- ✓ FAUX

Solution

FAUX. Le théorème de Shannon garantit $H(X) \leq L_H(X) \leq H(X) + 1$. La longueur moyenne de Huffman est donc toujours *supérieure ou égale* à l'entropie (et l'égalité est atteinte lorsque les probabilités sont des puissances de 2, comme dans la question précédente).

d) On souhaite transmettre la phrase suivante, espaces inclus :

`LA BASE SALSA EST LA`

A. Calculez la fréquence de chaque caractère (lettres et espaces).

B. Construisez un arbre de Huffman et donnez le code binaire de chaque caractère.

C. Calculez le nombre total de bits nécessaires pour encoder le message (i) avec le code de Huffman (B_{Huffman}) et (ii) avec un encodage à taille fixe minimale ($B_{\text{fixe-min}}$). Donnez ensuite le gain de compression en utilisant :

$$\text{gain de compression} = 1 - \frac{B_{\text{Huffman}}}{B_{\text{fixe-min}}}.$$

Les résultats peuvent être donnés sous forme de fraction ou de valeur décimale.

D. Comparez la longueur moyenne théorique minimale à la longueur moyenne obtenue par votre code de Huffman.

Solution

A. La phrase comporte 20 caractères. Fréquences :

Caractère	A	␣	S	L	E	B	T
Nb app.	5	4	4	3	2	1	1
p_j	$\frac{5}{20}$	$\frac{4}{20}$	$\frac{4}{20}$	$\frac{3}{20}$	$\frac{2}{20}$	$\frac{1}{20}$	$\frac{1}{20}$

B. Construction de l'arbre de Huffman (on fusionne itérativement les deux nœuds de plus faible poids) :

1. Fusionner B(1) + T(1) ⟶ BT(2)
2. Fusionner BT(2) + E(2) ⟶ BTE(4)
3. Fusionner L(3) + ␣(4) ⟶ L␣(7) [ou S(4) selon les ex aequo]
4. Fusionner S(4) + BTE(4) ⟶ SBTE(8)
5. Fusionner A(5) + L␣(7) ⟶ AL␣(12)
6. Fusionner SBTE(8) + AL␣(12) ⟶ racine(20)

Un code valide obtenu (d'autres sont possibles selon les ex aequo) :

Caractère	Code	Longueur
A	`10`	2
␣	`111`	3
S	`00`	2
L	`110`	3
E	`011`	3
B	`0100`	4
T	`0101`	4

C. Total Huffman :

$$B_{\text{Huffman}} = 5\times 2 + 4\times 3 + 4\times 2 + 3\times 3 + 2\times 3 + 1\times 4 + 1\times 4 = 10 + 12 + 8 + 9 + 6 + 4 + 4 = 53 \text{ bits.}$$

Encodage à taille fixe : 7 symboles distincts, donc $\lceil \log_2 7 \rceil = 3$ bits/symbole, soit $B_{\text{fixe-min}} = 20 \times 3 = 60$ bits.

$$\text{gain de compression} = 1 - \frac{53}{60} = \frac{7}{60} \approx 11.7\%.$$

D. Entropie de la source :

$$H(X) = \frac{11}{10} + \frac{3}{4}\log_2(5) - \frac{3}{20}\log_2(3) \approx 1.1 + 1.74 - 0.237 \approx 2.60 \text{ bits/symbole.}$$

Longueur moyenne du code de Huffman : $L_H(X) = 53/20 = 2.65$ bits/symbole. On vérifie bien $H(X) \le L_H(X) \le H(X) + 1$, soit $2.60 \le 2.65 \le 3.60$.