Théorie de l'Information

Devoir # 3 (Résolution)

SIC-SICOM

Maria-João Rendas

1. Lequel de ces codes ne peut pas être un code de Huffman?

   (a) $\{1, 01, 00\}$
   (b) $\{00, 01, 10, 110\}$
   (c) $\{01, 10\}$

   Justifiez toutes vos réponses.

   Résolution

   (a) $\{1, 01, 00\}$ est un code de Huffman pour la distribution $[1/2, 1/4, 1/4]$ (ses mots de code ont une longueur $\ell_i = -\log p_i$).

   (b) Le mot le plus long de ce code n'a pas de "jumeau," ce qui indique qu'il ne s'agit pas d'un code optimal (de Huffman): le code n'**est pas complet**, comme nous pouvons le vérifier en effectuant la somme dans l'inégalité de Kraft:

   $$\sum_{i=1}^{4} 2^{-\ell_i} = 0.875.$$

   Le code $\{00, 01, 10, 110\}$ peut être remplacé par le code plus court $\{00, 01, 10, 11\}$ sans perde la prossibilité de le décoder instantanément. Il n'est donc pas un code de Huffman.

   (c) L'arbre binaire qui représente le code $\{01, 10\}$ a **des noeuds internes qui n'ont qu'un seul descendant**, et donc il ne peut pas correspondre à un code optimal.

2. Considérez la variable aléatoire avec la loi de probabilité dans le tableau suivant:

$x \in \mathcal{X}$	p(x)
x_1	0.5
x_2	0.26
x_3	0.11
x_4	0.04
x_5	0.04
x_6	0.03
x_7	0.02

(a) Construisez un code de Huffman pour X.

(b) Calculez la longueur moyenne du code obtenu.

(c) Construisez un code ternaire pour X.

Résolution

(a) Le code de Huffman pour cette source est

Code	x_i							
1	x_1	0.5	0.5	0.5	0.5	0.5	0.5	1
01	x_2	0.26	0.26	0.26	0.26	0.26	0.5	
001	x_3	0.11	0.11	0.11	0.11	0.24		
00011	x_4	0.04	0.04	0.08	0.13			
00010	x_5	0.04	0.04	0.05				
00001	x_6	0.03	0.05					
00000	x_7	0.02						

(b) La taille moyenne du code binnaire obtenu est

$$L(X) = \sum_{i=1}^{7} p_i \ell(x_i) = 2 \text{ bits.}$$

Cette valeur est à moins d'un bit de l'entropie de la source:

$$H(X) = -\sum_{i=1}^{7} p_i \log_2 p(x_i) = 1.992 \text{ bits.}$$

C'est à dire, nous vérifions l'inégalité vue en cours :

$$L(X) \leq H(X) + 1 \,.$$

(c) Por construire un code ternaire, nous procédons comme pour le code binaire, mais à chaque itération nous devons ajouter un noeud dans un arbre ternaire, et regrouper donc les trois symboles (ou "meta-symboles") les moins probables. L'application de l'agorithme nous conduit au tableau suivant

Code	x_i				
0	x_1	0.5	0.5	0.5	1
1	x_2	0.26	0.26	0.26	
20	x_3	0.11	0.11	0.24	
21	x_4	0.04	0.04		
222	x_5	0.04	0.09		
221	x_6	0.03			
220	x_7	0.02			

La longueur moyenne de ce code est :

$$L_3(X) = \sum_{i=1}^{7} p_i \ell(x_i) = 1.33 \text{ symboles ternaires.}$$

L'entropie de la source calculée en symboles ternaires est

$$H_3(X) = \frac{1}{\log_2(3)} H(X) = 1.2566 < L_3(X)\,.$$

3. Des mots comme "Stop" ou "Feu" sont petits, pas car leur utilisation est fréquente, mais peut-être car on souhaite miniser le temps nécessaire pour les dire. Supposez que $X = i$ avec probabilité $p_i, i = 1, \ldots, m$. Soit ℓ_i le nombre de bits nécessaires pour coder $X = i$, et c_i le coût par lettre du mot X_i. Le coût moyen du code est donc

$$C = \sum_{i=1}^{m} p_i \ell_i c_i$$

   (a) Minimisez C défini par l'équation précédante sur toutes les longueurs $\{\ell_i\}_{i=1}^m$, telles que l'inégalité de Kraft est satisfaite:

$$\sum_{i=1}^{m} 2^{-\ell_i} \leq 1.$$

   Ignorez, pour cette minimisation, les contraintes qui imposent que les ℓ_i soient des entiers. Calculez les longueurs optimales $\ell_i^\star$, et le coût minimal qui leur est associé, $C^\star$.

   (b) Comment pouvez-vous utiliser l'algorithme de Huffmann pour minimiser C sur tous les codes (binaires) uniquement décodables ? Soit $C_{Huffmann}$ le coût de ce code optimal.

   (c) Montrez que

$$C^\star \leq C_{Huffmann} \leq C^\star + \sum p_i c_i.$$

Résolution

   (a) Nous souhaitons minimizer $C = \sum_i p_i c_i \ell_i$ sous la contrainte $\sum_i 2^{-\ell_i} \leq 1$. Nous allons considérer que la contrainte est satisfaite avec égalité et utiliser les notations

$$r_i = 2^{-\ell_i}, \qquad Q = \sum_i p_i c_i, \qquad q_i = \frac{1}{Q} p_i c_i,$$

   de façon que q est aussi une loi de probabilité. Avec ces définitions

$$\begin{aligned} C &= \sum_i p_i c_i \ell_i \\ &= Q \sum_i q_i \log \frac{1}{r_i} \\ &= Q \left(\sum_i q_i \log \frac{q_i}{r_i} - \sum_i q_i \log q_i \right) \\ &= Q \left(D(q||r) + H(q) \right). \end{aligned}$$

Cette expression est minimisée quand $q = r$, c'est à dire pour des longueurs

$$\ell_i^\star = -\log \frac{p_i c_i}{\sum_j p_j c_j},$$

où nous avons ignoré que ℓ_i doivent être des entiers. Le coût optimal associ é à ce choix est

$$C^\star = QH(q),$$

car $D(q||r)$ est dans ce cas égale à zéro.

(b) Si nous utilisons l'algorithme de Huffmann avec q à la place de p, nous obtiendrons un code qui minimise C.

(c) Si nous prennons en compte que les ℓ_i doivent être des entiers

$$\ell_i = \lceil -\log q_i \rceil.$$

Alors

$$-\log q_i \leq \ell_i \leq -\log q_i + 1.$$

Si nous multiplions cette équation par $p_i c_i$ et nous effectuons la somme sur toutes les valeurs de i nous obtenons

$$C^\star \leq C_{\textit{Huffmann}} \leq C^\star + \sum_i p_i c_i = C^\star + Q.$$

4. Soit $\mathcal{X} = \{1, 2, 3, 4, 5\}$. Considérez les deux lois de probabilité p et q d éfinies sur cet alphabet, et les deux codes C_1 et C_2 dans le tableau suivant

$x \in \mathcal{X}$	$p(x)$	$q(x)$	C_1	C_2
1	0.5	0.5	0	0
2	0.25	0.125	10	100
3	0.125	0.125	110	101
4	0.0625	0.125	1110	110
5	0.0625	0.125	1111	111

(a) Calculez $H(p)$, $H(q)$, $D(p||q)$ et $D(q||p)$.

(b) Vérifiez que C_1 est optimal pour p et que C_2 est optimal pour q (calculez leurs longueurs moyennes).

(c) Admettez que l'on utilise C_2 pour une source $X \sim p$. Quelle est la longueur moyenne des mots de code? De combien elle dépasse l'entropie $H(p)$?

(d) Quelle est la pénalité si nous utilisons C_1 pour une source $X \sim q$?

Résolution

(a)

$$H(p) \quad = \quad -\sum_i p_i \log p_i = \frac{15}{8}.$$

La même expression nous conduit à $H(q) = 2$.

$$D(p||q) = \sum_i p_i \log \frac{p_i}{q_i} = \frac{1}{8}.$$

Nous pouvons constater que pour ces deux distributions (cela nest pas nécessairement le cas!) $D(q||p) = 1/8$.

(b) La taille moyenne pour le code C_1 est

$$L_{C_1}(p) = \sum_i p_i \log \ell_i^{C_1} = \frac{15}{8} = H(p),$$

et donc C_1 est bien optimal pour la distribution p. Pour le code C_2 nous obtenons $L_{C_2}(q) = 2 = H(q)$, ce qui montre que C_2 est optimal pour q.

(c) Dans ce cas nous aurons

$$L_{C_2}(p) = \sum_i p_i \ell_i^{C_2} = 2,$$

qui est supérieur à l'entropie de la source $H(p) = 15/8$. La pénalité est

$$L_{C_2}(p) - H(p) = \frac{1}{8} = D(p||q).$$

(d) Comme l'entropie relative $D(q||p) = D(p||q)$ la pénalité sera la même que nous avons calculé dans l'alinéa précédante: $D(q||p) = 1/8$.

5. On nous donne 6 bouteilles de vin. On sait qu'une des bouteilles est mauvaise. L'analyse visuelle des bouteilles permet de déterminer la probabilité pour que chacune d'entre elles soit celle qui est mauvaise:

bouteille # i	p_i
1	7/26
2	5/26
3	4/26
4	4/26
5	3/26
6	3/26

Nous souhaitons déterminer la mauvaise bouteille.
Supposez que l'on goute les vins un à un. Choisissez l'ordre par laquelle ils doivent être goutés pour que le nombre d'essais soit minimisé. Rappelez-vous que si les premières 5 bouteilles sont bonnes, vous ne devez pas gouter la dernière.

(a) Quel est le nombre moyen d'essais qui doivent être réalisés?

(b) Quelle bouteille doit être testée en premier?

Admettez maintenant que vous pouvez gouter le mélange du contenu de plusieurs bouteilles, au lieu de les gouter une à une.

(c) Quelle est la valeur minimale du nombre moyen de tests qui doivent être faits pour déterminer la mauvaise bouteille?

(d) Quel mélange doit être testé en premier ?

Résolution

(a) Considérez que nous essayons les vins **un à un**, et soient $n_i, i = 1, 6$ l'ordre par laquelle nous les goutons. Alors, si la bouteille n_1 est la mauvaise, nous aurons à faire un seul test. Si la bouteille n_2 est la mauvaise, nous devrions effectuer 2 tests, ainsi de suite. Le tableau suivant résume ce raisonnement

Bouteille	# tests t_i
n_1	1
n_2	2
n_3	3
n_4	4
n_5	5
n_6	5

Le nombre de tests moyens réalisés est

$$\sum_i p(n_i)t_i$$

où $p(n_i)$ est la probabilité que la bouteille n_i soit la mauvaise.

Cette somme sera minimale si nous choisssons les n_i de façon que si $t_i > t_j \Rightarrow p(n_i) < p(n_j)$. Nous devons donc essayer d'abord les bouteilles qui ont une plus forte chance d'être les mauvaises. Le nombre moyen de tests réalisés est

$$\sum_i p_i t_i = 1 \times \frac{7}{26} + 2 \times \frac{5}{26} + 3 \times \frac{4}{26} + 4 \times \frac{4}{26} + 5 \times \frac{3}{26} + 5 \times \frac{3}{26} = 2.88$$

(b) La bouteille qui doit être testée en premier est celle qui a une probabilité de $7/26$ d'être mauvaise.

(c) Nous utilisons l'algorithme de Huffman pour construire un code binaire pour la loi de probabilité p_i :

01	7/26	7/26	8/26	11/26	15/26	1
11	5/26	6/26	7/26	8/26	11/26	
000	4/26	5/26	6/26	7/26		
001	4/26	4/26	5/26			
100	3/26	4/26				
101	3/26					

Ce code fait associer à chaque bouteille une séquence binaire, d'une manière univoque, et de façon que le nombre de bits est en moyenne minimal. Nous interprétons alors chaque bit des mots de ce code binaire comme le résultat d'un test binaire (pour lequel le résultat indique (1) - ou pas (0) - la présence de mauvais vin (1) dans le mélange essayé. Le mélange testé dans chaque test est indiqué par les bouteilles pour lesquelles le bit correspondant est égal à un. Par exemple, dans le premier test le élange des bouteilles 2, 5, et 6 doit être testé.

Le nombre moyen de tests réalisés est

$$\sum_i p_i \ell_i = 2.54.$$

(d) Le premier test doit être réalisé sur le mélange des bouteilles 2, 5 et 6. Le deuxième sur le mélange des bouteilles 1 et 2. Si le résultat du premier test est 0 et le deuxième est positif, nous savons que c'est la bouteille 1 qui est mauvaise. Si les deux tests sont positifs, nous savons que c'est la bouteille 2 qui est mauvaise. Dans tous les autres cas nous devons effectuer un test suplémentaire, avec le mélange des bouteilles 4 et 6. La mauvaise bouteille est alors identifiée de la façon suivante:

- Si dans les deux premiers tests nous avons obtenu 00 alors si le troisième test est négatif c'est la bouteille 3 qui est mauvaise, sinon c'est la numéro 4.
- Si dans les deux premiers tests nous avons obtenu 10 alors si le troisième test est négatif c'est la bouteille 5 qui est mauvaise, sinon c'est la numéro 6.