Correction de l'examen du cours de Théorie de l'Information et Codage

1. Problème 1: On considère un code de Huffman pour une source U où $\mathbb{P}(u)$ est une puissance de $1/2$ pour tout u.

   (a) Montrer que le code de Huffman aura une longueur moyenne égale à l'entropie de la source.
   On considère maintenant une source discrète stationnaire: $X_0, X_1, \ldots$ c'est à dire on a $\mathbb{P}(X_0) = \mathbb{P}(X_k)$ et $\mathbb{P}(X_1|X_0) = \mathbb{P}(X_{k+1}|X_k)$ pour tout $k \geq 0$ (mais la suite de variables aléatoires n'est pas forcément i.i.d.). Pour simplifier, on supposera que toutes les probabilités considérées sont des puissances de $1/2$. Nous considérons deux méthodes:

   - Méthode 1: on considère les paires successives $(X_1, X_2), (X_3, X_4), \ldots$ et on encode chaque paire en utilisant le code de Huffman pour la paire.
   - Méthode 2: On encode chaque X_k en utilisant un code de Huffman basé sur la probabilité conditionelle de X_k sachant la précédente valeur X_{k-1} (Il faut donc utiliser en général $|\mathcal{X}|$ codes de Huffman différents).

   (b) Calculer la longueur moyenne des mots code par lettre pour la méthode 1.

   (c) Calculer la longueur moyenne des mots code par lettre pour la méthode 2.

   (d) Quelle méthode est la meilleure?

   - Soit $\ell(u) = -\log_2 \mathbb{P}(u)$ de telle sorte que $\sum_u 2^{-\ell(u)} = 1$. La suite des $\ell(u)$ vérifie donc l'inégalité de Kraft et il existe un code ayant pour la lettre u un mot-code de longueur $\ell(u)$. La longueur moyenne d'un mot-code est alors $-\sum_u \mathbb{P}(u) \log_2 \mathbb{P}(u) = H(U)$, l'entropie de la source. Comme le codage de Huffman fait au moins aussi bien et qu'il ne peut pas faire mieux que l'entropie, le code de Huffman a une longueur moyenne égale à l'entropie de la source.
   - Par stationnarité la distribution de (X_{2k-1}, X_{2k}) est la même pour tout k. Donc par la question précédente, la longueur moyenne des mots code est $H(X_1, X_2)$. Comme cette méthode encode deux lettres à la fois, la longueur moyenne par lettre est $\frac{1}{2}H(X_1, X_2)$.
   - Si la lettre précédente est y, la longueur moyenne est

   $$-\sum_x \mathbb{P}(X_2 = x|X_1 = y) \log_2 \mathbb{P}(X_2 = x|X_1 = y).$$

   Comme la probabilité que la lettre précédente soit y est $\mathbb{P}(X_1 = y)$, on a pour la longueur moyenne de la méthode 2:

   $$-\sum_y \mathbb{P}(X_1 = y) \sum_x \mathbb{P}(X_2 = x|X_1 = y) \log_2 \mathbb{P}(X_2 = x|X_1 = y) = H(X_2|X_1).$$

   - On a :

   $$H(X_1, X_2) = H(X_1) + H(X_2|X_1) \geq H(X_1|X_0) + H(X_2|X_1) = 2H(X_2|X_1),$$

   donc la seconde méthode est la meilleure.

2. Problème 2: La course de chevaux. Trois chevaux participent à la course. Si vous pariez 1 euro avant la course, vous en recevez 3 si votre cheval gagne (gain net de 2) et rien s'il n'est pas gagnant (perte de 1). Les probabilités de gagner sont connues et valent $(p_1, p_2, p_3) = (1/2, 1/4, 1/4)$. Vous distribuez la totalité de votre mise sur les chevaux. Soit b_i la fraction correspondant au cheval i. On note $\mathbf{b} = (b_1, b_2, b_3)$, $b_i \geq 0$ et $\sum b_i = 1$. Si le cheval i gagne la course, vous gagnez donc $3b_i$, les autres paris étant perdus. Il y a une infinité de courses (!) dont les résultats sont indépendants les uns des autres et à chaque course, vous misez l'intégralité de votre mise.

   (a) Montrer que si à chaque course, vous jouez toujours le cheval 1 (celui qui a le plus de chance de gagner), vous allez perdre votre mise initiale avec probabilité 1 au bout d'un temps fini.

   (b) On définit
   $$W(\mathbf{b}) = \sum_{i=1}^{3} p_i \log 3b_i.$$
   Maximiser W et en déduire une stratégie (fixe dans le temps) qui permet de gagner de l'argent.

   - On perd après un temps géométrique de paramètre 1/2.
   - On a
   $$\begin{aligned} W(\mathbf{b}) &= \sum_i p_i \log 3 + \sum_i p_i \log p_i - \sum_i p_i \log \frac{p_i}{b_i} \\ &= \log 3 - H(\mathbf{p}) - D(\mathbf{p}||\mathbf{b}) \\ &\leq \log 3 - H(\mathbf{p}), \end{aligned}$$
   avec égalité ssi $\mathbf{p} = \mathbf{b}$. Donc $\mathbf{b}^* = \mathbf{p}$. On a alors pour la somme d'argent après n itérations:
   $$\begin{aligned} S_n &= \prod_j 3b(X_j) \\ &= 2^{n\left(\frac{1}{n}\sum_j \log 3b(X_j)\right)} \\ &\to 2^{nW(\mathbf{b})}. \end{aligned}$$
   Donc pour $\mathbf{b} = \mathbf{b}^*$, on a $S_n = 2^{nW^*} = (1,06)^n$.

3. Problème 3: quel est le découpage effectué par l'algorithme de Lempel-Ziv pour la suite constante $11111\ldots$? Montrer que le nombre de bits codants par symbole pour cette suite tend vers zéro quand $n \to \infty$.

   - découpage: $1, 11, 111, 1111, \ldots$
   - On a donc $c_n = O(\sqrt{n})$, et il faut de l'ordre de $\frac{c_n \log c_n}{n}$ bits d'encodage par symbole.

4. Problème 4: On considère le canal binaire sans mémoire à effacement: alphabet d'entrée $A_X = \{0, 1\}$, alphabet de sortie $A_Y = \{0, *, 1\}$, probabilités de transition $p(y|x))$ données par $p(0|0) = p(1|1) = 1 - \alpha$ et $p(*|0) = p(*|1) = \alpha$. On considère que le coût $b(x)$ est donné par $b(0) = 1$ et $b(1) = 2$. Calculer la fonction capacité coût du canal.

- D'après l'étude du canal à effacement, on sait que $I(X;Y) = (1-\alpha)H(\pi)$ où $\pi = \mathbb{P}(X = 1)$. Il faut donc maximiser $(1-\alpha)H(\pi)$ sous la contrainte de coût $\pi + 1 \leq \beta$. On a donc:

$$C(\beta) = \begin{cases} 1-\alpha & 1/2 \leq \beta - 1 \\ (1-\alpha)H(\beta-1) & 0 \leq \beta - 1 \leq 1/2 \\ 0 & \beta - 1 < 0. \end{cases}$$

5. Problème 5: Calculer la capacité d'une cascade de canaux binaires sans mémoires symmétriques:

$$X_0 \to \boxed{\text{BSC 1}} \to X_1 \to \cdots \to X_{n-1} \to \boxed{\text{BSC n}} \to X_n,$$

chacun avec probabilité d'erreur p. Montrer que si $p \neq 0, 1$, on a $\lim_{n\to\infty} I(X_0, X_n) = 0$.

- La cascade de canals est équivalente à un canal binaire symétrique de paramètre: $q = \frac{1}{2}(1 - (1-2p)^n)$, ceci se montre par récurrence sur n.

6. Problème 6: On considère un code binaire ayant M mots code de longueur n. On suppose que ce code peut corriger jusqu'à e erreurs.

   (a) Pour un mot code x, soit S l'ensembles des suites binaires pour lesquelles le décodeur décide x. Montrer que

$$|S| \geq \sum_{i=0}^{e} \binom{n}{i}.$$

   (b) Montrer que le nombre de mots code M satisfait:

$$M \leq \frac{2^n}{\sum_{i=0}^{e} \binom{n}{i}}.$$

   (c) Montrer que les codes de Hamming $[n = 2^m - 1, k = n - m, d = 3]$ sont les codes corrigeant une erreur qui pour cette longueur ont le taux le plus élevé possible.

   (d) Un code BCH binaire de longueur $n = 2^m - 1$ et de paramètre $\delta = 2t + 1$ a pour distance minimale d où $d \geq 2t + 1$ est impaire. Montrer que si

$$\sum_{i=0}^{t+1} \binom{2^m - 1}{i} \geq 2^{mt},$$

   alors $d = 2t + 1$.

- Le décodeur doit décider x pour tout y qui diffère de x par au plus e bits. Donc $|S| \geq \sum_{i=0}^{e} \binom{n}{i}$.
- Soit S_x l'ensemble des suites pour lesquelles le décodeur décide x. Ces ensembles sont disjoints donc $M \sum_{i=0}^{e} \binom{n}{i} \leq 2^n$.
- Pour les codes de Hamming, on a $n = 2^m - 1$, $M = 2^{2^m - m - 1}$ et $e = 1$ donc $\sum_{i=0}^{e} \binom{n}{i} = 1 + n = 2^m$ et les bornes sont atteintes.

- Supposons $d \geq 2t+3$. La dimension du code est $\geq n-mt$ donc par (b), $2^{n-mt}\sum_{i=0}^{t+1}\binom{n}{i} \leq 2^n$, contradiction.

7. Problème 7: On considère le code de Hamming étendu: chaque mot code d'un code de Hamming est étendu d'un bit qui vaut 0 si le mot code contient un nombre paire de 1 et 1 sinon. Par exemple le mot code 1000011 devient 10000111. Montrer qu'un tel code a une distance minimale de 4, peut corriger 1 erreur et en détecter 2.

   - Soit x et x' deux mots code différents. On note z et z' les parties correspondant au code de Hamming et p, p' le bit ajouté. Comme $x \neq x'$ on a aussi $z \neq z'$ et donc $d_H(z,z') \geq 3$. Si $d_H(z,z') = 3$ alors z et z' ont des parités différentes et donc $d_H(x,x') = 4$. Si $d_H(z,z') \geq 4$ alors clairement $d_H(x,x') \geq 4$. Donc la distance minimale du nouveau code est 4.

     On considère la méthode de décodage suivante: pour une suite y reçue, la comparer à tous les mots code et calculer les distances de hamming correspondantes. S'il existe un unique mot code ayant une distance minimale, décoder par ce mote code. Sinon retourner erreur.

     Si la distance minimale d'un code est $d = 2j$ alors si x est émis et y reçu et que $d_H(x,y) \leq j-1$ alors le décodage sera bon. Par contre si $d_H(x,y) = j$, il peut y avaoir un mot code x' tq $d_H(x',y) = j$ et on aura alors une erreur détectée mais pas corrigée.

8. Problème 8: Si $n = ab$, montrer que le code binaire BCH de longueur n et paramètre a a pour distance minimale a. On pourra montrer que $1 + x + \cdots + x^{(a-1)b}$ est un mot code.

   - Soit α une racine n-ième primitive de l'unité. Donc $\alpha^{ib} \neq 1$ pour $i < a$. Comme $x^n - 1 = (x^b-1)(1 + x^b + x^{2b} + \cdots + x^{(a-1)b})$, les éléments $\alpha, \alpha^2, \ldots \alpha^{a-1}$ ne sont pas racines de $x^b - 1$ et donc sont racines de $1 + x^b + x^{2b} + \cdots + x^{(a-1)b}$. En particulier ce dernier est un mot code de poids a.