

Solutions des Exercices du cours de Théorie de l'Information et Codage

cours 3 du 3 mars 2009.

1. Soient X_1, \dots, X_n , des v.a. discrètes. Montrer que

$$H(X_1, \dots, X_n) = H(X_1) + \sum_{k=2}^n H(X_k | X_1, \dots, X_{k-1}).$$

- Le cas $n = 2$ a été vu lors du premier cours. En fait, on a:

$$\begin{aligned} H(X, Y | Z) &= \sum_{x,y,z} p(x, y, z) \log \frac{1}{p(x, y | z)} \\ &= \sum_{x,y,z} p(x, y, z) \log \frac{1}{p(x | z) p(y | x, z)} \\ &= H(X | Z) + H(Y | X, Z) \end{aligned}$$

Le résultat s'obtient alors par itération:

$$\begin{aligned} H(X_1, X_2, X_3) &= H(X_1) + H(X_2, X_3 | X_1) \\ &= H(X_1) + H(X_2 | X_1) + H(X_3 | X_2, X_1), \end{aligned}$$

où la première égalité découle du cas $n = 2$ vu en cours.

2. On considère une chaîne de Markov (à espace d'états finis) ergodique S_0, S_1, \dots avec un état initial arbitraire. Montrer que $H(S_2 | S_1, S_0) = H(S_2 | S_1)$ et que pour tout $n \geq 2$

$$H(S_1, S_2, \dots, S_n | S_0) = \sum_{k=1}^n H(S_k | S_{k-1}).$$

Simplifier la formule quand S_0 suit la loi stationnaire. Pour une source Markovienne X_1, X_2, \dots montrer que $H(X_1, \dots, X_n | S_0) = H(S_1, \dots, S_n | S_0)$. Montrer que pour une source Markovienne stationnaire:

$$H(X_1, \dots, X_n | S_0) = nH(X | S).$$

Interpréter le résultat précédent pour le codage par blocs d'une source Markovienne.

- Le premier point découle des définitions:

$$H(S_2 | S_1, S_0) = \sum p(s_0, s_1, s_2) \log \frac{1}{p(s_2 | s_0, s_1)} = \sum p(s_0, s_1, s_2) \log \frac{1}{p(s_2 | s_1)}.$$

- Le deuxième point découle de l'exercice précédent. Dans le cas où S_0 suit la loi stationnaire, les loi des couples (S_{k-1}, S_k) sont les mêmes donc

$$H(S_1, S_2, \dots, S_n | S_0) = nH(S_1 | S_0).$$

D'après la définition d'une source Markovienne, on a $X_i = f(S_{i-1}, S_i)$ où f est une injection. Donc $(X_1, \dots, X_n) = g(S_0, S_1, \dots, S_n)$ avec g une injection et $H(X_1, \dots, X_n | S_0) = H(S_1, \dots, S_n | S_0)$ par l'exercice 4 de la première fiche d'exos et le dernier point suit facilement.

- Soit $X^n = (X_1, \dots, X_n)$ et $\mathbb{E}[L(X^n)]_{\min, s}$ la longueur moyenne minimale d'un code instantané pour X^n conditioné à l'événement: l'état de départ est s . D'après le Théorème 2.4 du cours, on a

$$H(X^n|s) \leq \mathbb{E}[L(X^n)]_{\min, s} < H(X^n|s) + 1.$$

Si l'état initial est connu du décodeur et S_0 suit la loi stationnaire alors:

$$H(X^n|S_0) \leq \mathbb{E}[L(X^n)]_{\min, S_0} < H(X^n|S_0) + 1$$

donc avec les résultats précédents:

$$H(X|S_0) \leq L_{\min, n} < H(X|S_0) + \frac{1}{n},$$

avec $L_{\min, n} = \frac{\mathbb{E}[L(X^n)]_{\min, S_0}}{n}$ la longueur moyenne minimum d'un mot code par symbole d'entrée dans l'état stationnaire.

3. On considère le problème d'encodage d'une suite binaire $x^n \in \{0, 1\}^n$.

- Algorithme 1: nous lisons entièrement la suite, puis nous procédons en deux étapes: nous envoyons d'abord le nombre de 1 dans la suite puis nous envoyons les emplacements de ces 1 dans la suite. Montrer que la longueur du mot code est bornée par

$$\ell(x^n) \leq H(k/n) + 1/2 \log n - 1/2 \log \left(\pi \frac{k(n-k)}{n^2} \right) + 3,$$

où $k = \sum_i x_i$. On pourra utiliser l'encadrement suivant:

$$\sqrt{\frac{n}{8k(n-k)}} \leq \binom{n}{k} 2^{-nH(k/n)} \leq \sqrt{\frac{n}{\pi k(n-k)}}. \quad (1)$$

- Algorithme 2: utiliser un code arithmétique avec la règle de Laplace pour le modèle prédictif. Montrer que la longueur du mot code est bornée par

$$\ell(x^n) \leq H(k/n) + 1/2 \log n - 1/2 \log \left(\pi \frac{k(n-k)}{n^2} \right) + 2.$$

Quel est l'intérêt de la seconde méthode?

- Attention, il y avait une erreur dans l'énoncé.
- Algorithme 1: la première étape nécessite $\lceil \log(n+1) \rceil$ bits et la seconde étape, $\lceil \log \binom{n}{k} \rceil$ bits donc au total:

$$\begin{aligned} \ell(x^n) &\leq \log(n+1) + \log \binom{n}{k} + 2 \\ &\leq \log n + H(k/n) - \frac{1}{2} \log n - \frac{1}{2} \log \left(\pi \frac{k(n-k)}{n^2} \right) + 3 \end{aligned} \quad (2)$$

$$= H(k/n) + \frac{1}{2} \log n - \frac{1}{2} \log \left(\pi \frac{k(n-k)}{n^2} \right) + 3. \quad (3)$$

- Algorithme 2: on a vu en cours que le codage arithmétique avec la règle de Laplace, revient à coder pour $q(x_1, \dots, x_n)$ une mixture uniforme de distributions de Bernoulli, c.à.d. $q(x_1, \dots, x_n) = \int_0^1 \theta^k (1 - \theta)^{n-k} d\theta = A(n, k)$ (défini en cours) où $k = \sum_i x_i$. On a donc pour la longueur d'un mot code

$$\begin{aligned} \ell(x^n) &= \lceil \log \frac{1}{q(x^n)} \rceil \\ &\leq \log(n+1) + \log \binom{n}{k} + 1. \end{aligned}$$

- Dans les deux cas, le 'coût' pour décrire la suite est approximativement de $1/2 \log n$ bits en plus du coût optimal avec un code de Shannon pour une distribution de Bernoulli de paramètre k/n . Cependant, le premier algorithme nécessite de lire la séquence entière avant de commencer tandis que le second algorithme atteint le même résultat 'à la volée'.
- Noter que l'encadrement (1) n'est valable que pour $k \neq 0$ et $k \neq n$. En particulier, la borne (3) est infinie si $k = 0$ ou, n . Dans le cas de l'algorithme 2, on peut utiliser la borne

$$\binom{n}{k} \leq 2^{nH(k/n)}, \quad (4)$$

pour obtenir une borne sur la redondance de l'algorithme de $\log n$ (si $k = 0$ ou, n , on a $H(k/n) = 0$). Ceci est relativement mauvais et provient du fait que le modèle probabiliste ne donne pas assez de masse aux séquences avec $k = 0$ ou, n .

Preuve de (4):

$$\begin{aligned} 1 &\geq \binom{n}{k} \left(\frac{k}{n}\right)^k \left(1 - \frac{k}{n}\right)^{n-k} \\ &= \binom{n}{k} 2^{n\left(\frac{k}{n} \log \frac{k}{n} + \frac{n-k}{n} \log \frac{n-k}{n}\right)}. \end{aligned}$$