

CODES POUR DES SOURCES DISCRÈTES

Exercice 1

Test de l'ambiguïté d'un code

Le but de cet exercice est de donner un algorithme pour tester si un code est ambigu. Par exemple, on se donne le code $C = \{1, 011, 01110, 1110, 10011\}$

1. Montrer que C est un code ambigu.

Étant donné un code C sur un alphabet fini, on définit les langages suivants (avec les notations définies en cours de langage formels) :

$$\begin{aligned} U_1 &= C^{-1}C - \{\epsilon\}, \\ U_{n+1} &= C^{-1}U_n + U_n^{-1}C, \text{ pour } n \geq 1. \end{aligned}$$

2. Calculer U_1 , U_2 et U_3 pour le code C de la question 1.
3. Calculer U_n pour tout $n \geq 1$ pour le code $\{10, 00, 11, 110\}$.
4. Montrer par induction sur k que pour tout $n \geq 1$ et tout $k \in \{1, \dots, n\}$, on a

$$\epsilon \in U_n \Leftrightarrow \exists u \in U_k, i, j \in \mathbb{N} \text{ tels que } uC^i \cap C^j \neq \emptyset \text{ et } i + j + k = n.$$

5. En déduire le théorème suivant :

Théorème : Le code C est non-ambigu si et seulement si aucun des ensembles U_n ne contient le mot vide.

Exercice 2

Codage de Tunstall

La méthode de codage décrite dans ce sujet diffère du point de vue adopté jusque ici : ce n'est plus la longueur des mots-code qui est autorisée à varier en fonction du contenu des blocs de taille fixe émis par la source, mais l'inverse. Dans tout le problème, on fixe un alphabet-source $\mathcal{X}$ de taille $N \geq 1$, muni d'un ordre total arbitraire. On se donne également deux entiers $L \geq 1$ et $D \geq 2$.

Dictionnaires admissibles

Définition 1. On appelle **dictionnaire** sur l'alphabet $\mathcal{X}$ un ensemble fini $\mathcal{F}$ de mots non vides de $\mathcal{X}^*$. Ses éléments sont appelés **lexèmes**.

Le principe général du codage *taille variable* $\rightarrow$ *taille fixe* est le suivant : étant donné un dictionnaire $\mathcal{F}$ fixé une fois pour toute, la suite de lettres émises par la source est simplement découpée au fur et à mesure en lexèmes, puis chaque lexème est codé par le chiffre D -aire de taille L qui représente son indice lexicographique dans le dictionnaire.

Exemple 1. On désire coder les mots de l'alphabet-source $\mathcal{X} = \{a, b, c\}$ (de taille $N = 3$) à l'aide de mots-code de $L = 3$ bits chacun ($D = 2$ donc). On peut pour cela considérer le dictionnaire $\mathcal{F} = \{aaa, aab, aac, ab, ac, b, c\}$. Dans ce cas le mot-source $abacbaabacaaabc$ pourra être découpé en :

$$(ab)(aac)(b)(aab)(ac)(aaa)(b)(c),$$

puis codé par :

$$011\ 010\ 101\ 001\ 100\ 000\ 101\ 110.$$

Définition 2. On dira qu'un dictionnaire $\mathcal{F}$ sur l'alphabet $\mathcal{X}$ est :

- **valide** si tout mot suffisamment long sur $\mathcal{X}$ admet au moins un préfixe dans $\mathcal{F}$;
- **non ambigu** si tout mot sur $\mathcal{X}$ admet au plus un préfixe dans $\mathcal{F}$;
- **instantané** si aucun lexème n'est préfixe d'un autre lexème.

1. Montrer qu'un dictionnaire est non ambigu si et seulement s'il est instantané. Quelle est la taille maximum d'un tel dictionnaire pour que les séquences de mots-code D -aires de taille L obtenues soient entièrement déchiffrables ?

On ne considérera donc désormais que des dictionnaires valides et instantanés sur $\mathcal{X}$ de taille $|\mathcal{F}| \leq D^L$. Ces dictionnaires seront dits D^L -admissibles.

- Établir une bijection entre l'ensemble des dictionnaires D^L -admissibles et une certaine famille d'arbres finis que l'on définira soigneusement. En particulier, quel est le nombre de fils de chaque sommet ?

Facteur de compression

Dans cette partie, on fixe un dictionnaire D^L -admissible $\mathcal{F}$, ainsi qu'une variable aléatoire X à valeurs dans l'alphabet-source $\mathcal{X}$. On se donne alors une suite infinie $X_1, X_2, \dots$ de copies indépendantes de X , et l'on note $Y_1, Y_2, \dots$ la factorisation en lexèmes (définie récursivement) de la suite $X_1, X_2, \dots$.

- Montrer que les variables aléatoires $Y_1, Y_2, \dots$ (à valeurs dans $\mathcal{F}$) sont les copies indépendantes d'une même variable aléatoire Y . En déduire, lorsque le nombre n de mots-code produits tend vers l'infini, la limite $\kappa(X, \mathcal{F})$ du **facteur de compression** de la source X par le dictionnaire $\mathcal{F}$ (nombre de lettres produites/nombre de lettres lues).

Soit $\mathcal{T}$ l'arbre associé au dictionnaire $\mathcal{F}$. Par construction, l'ensemble des sommets de $\mathcal{T}$ peut être identifié à l'ensemble $\mathcal{V}$ des préfixes des mots de $\mathcal{F}$. En particulier, la racine est le mot vide ϵ et les feuilles sont les lexèmes $f \in \mathcal{F}$. À tout sommet $v = v_1 \dots v_k \in \mathcal{V}$, on peut alors associer le nombre $P(v) = \mathbf{P}(X_1 = v_1, \dots, X_k = v_k)$ et $P(\epsilon) = 1$. On note $|v|$ la profondeur d'un sommet v dans l'arbre $\mathcal{T}$, et par extension $|f|$ désigne la profondeur du sommet associé au mot $f \in \mathcal{F}$.

- Étant donnée une fonction de pondération arbitraire $\pi : \mathcal{V} \setminus \{\epsilon\} \rightarrow \mathbb{R}$ sur l'ensemble des sommets de $\mathcal{T}$ (racine exceptée), on définit la hauteur pondérée $h_\pi(f)$ d'une feuille $f \in \mathcal{F}$ comme la somme des poids des sommets le long de l'unique chemin reliant ϵ (exclue) à f (incluse). Établir la relation :

$$\sum_{f \in \mathcal{F}} P(f) h_\pi(f) = \sum_{v \in \mathcal{V} \setminus \{\epsilon\}} P(v) \pi(v).$$

- En déduire :

$$(1) \mathbf{E}[|Y|] = \sum_{v \in \mathcal{V} \setminus \mathcal{F}} P(v); \quad (2) H_D(Y) = H_D(X) \mathbf{E}[|Y|]; \quad (3) \kappa(X, \mathcal{F}) = L \frac{H_D(X)}{H_D(Y)}.$$

Quelle borne naturelle obtient-on pour le facteur de compression ?

Algorithme de Tunstall

Ainsi le dictionnaire D^L -admissible optimal pour une source sans mémoire X est celui dont l'arbre maximise la somme des probabilités associées aux sommets internes. Il est donc naturel de considérer la stratégie gloutonne suivante, suggérée par Tunstall :

Algorithme 1: Algorithme de Tunstall (1968)

- Au départ, l'arbre est constitué de la racine et de ses N fils;
 - Tant que le nombre de feuilles est inférieur ou égal à $D^L - (N - 1)$, choisir une feuille dont la probabilité est maximale et l'éclater en un sommet interne et N feuilles.
-
- Construire le dictionnaire de Tunstall pour $\mathcal{X} = \{a, b\}$, $P(a) = 1 - P(b) = 0.7$, $L = 3$ et $D = 2$.
 - Démontrer que l'algorithme de Tunstall produit un dictionnaire D^L -admissible $\mathcal{F}_{D,L}^*$ dont le facteur de compression est minimal.
 - Démontrer que ce facteur de compression satisfait

$$H_D(X) \leq \kappa(X, \mathcal{F}_{D,L}^*) \leq \frac{H_D(X)L}{\log_D(|\mathcal{F}_{D,L}^*| p_{\min})},$$

où $p_{\min} = \min_{x \in \mathcal{X}} \mathbb{P}(X = x)$, et en déduire finalement : $\kappa(X, \mathcal{F}_{D,L}^*) \xrightarrow{L \rightarrow \infty} H_D(X)$.