Réseaux, information et communications (INFO-F303)
Partie Théorie de l'Information
1. Notion de code

Christophe Petit

Université libre de Bruxelles

Plan du cours

1. Notion de code
2. Source aléatoire et codes efficaces
3. Entropie et codage efficace
4. Compression sans perte
5. Canal bruité
6. Codes correcteurs d'erreurs
7. Codes linéaires
8. Quelques familles de codes linéaires

A. Rappels mathématiques (chapitre 7.1 du syllabus)

Chapitre 1 : notion de code

- Terminologie et notations
- Codes univoques
- Codes en bloc
- Codes sans préfixe et arbre de code
- Inégalité de Kraft
- Théorème de McMillan

Définitions

- Information représentée par des **symboles**, éléments d'un ensemble $S = \{s_1, s_2 \dots s_q\}$
- **Alphabet** du code, noté C, est de taille $r = |C|$
- **Fonction de codage**

$$K\colon S \to C^* : s \mapsto c_1 c_2 \dots c_\ell$$

  où $\ell \geq 1$ et C^* est l'ensemble des chaînes non vides de C
- $K(S) \subset C^*$ est le **code** ou ensemble des **mots du code**

Définitions (suite)

- On supposera souvent $r = 2$ (alphabet binaire)
- $\ell_i = |K(s_i)| > 0$
- $\ell_{\max}$: longueur maximale des mots

$$\ell_{\max} = \max_{i=1\ldots q}\{K(s_i)\}$$

- On étend naturellement K à S^*

$$K^* \colon S^* \to C^* : s_1 s_2 \ldots s_n \mapsto K(s_1)K(s_2)\ldots K(s_n)$$

Exemples

- $S = \{non, oui\}$, $C = \{0, 1\}$,
  et K tel que
  $$\begin{cases} K(non) = 0 \\ K(oui) = 1 \end{cases}$$

- $S = \{lun, mar, mer, jeu, ven, sam, dim\}$,
  $C = \{semaine, weekend\}$,
  et K tel que
  $$K(x) = \begin{cases} semaine & \text{si } x \in \{lun, mar, mer, jeu, ven\} \\ weekend & \text{si } x \in \{sam, dim\} \end{cases}$$

Codes univoques

- Un code est **univoque** ou **décodable** si la fonction de codage est inversible
- Pas de perte d'information
- Seuls codes considérés dans ce cours
- Le premier exemple ci-dessus est univoque, pas le deuxième

Codes en bloc

- Un code est un **code en bloc** si tous les mots ont la même longeur ℓ

$$K : S \to C^\ell : s \mapsto c_1 c_2 \ldots c_\ell$$

- Décodage “à la volée” facilité
- $q \leq r^\ell$ si univoque
- Très fréquents : ASCII, UTF-32, la plupart des codes correcteurs d'erreurs, ...
- Mais ce n'est pas obligatoire, et sous-optimal dans beaucoup de contextes

Codes à longueur variable

- Peuvent réduire la longueur du message codé (par exemple, en codant un “E” avec un mot plus court et “Z” avec un mot plus long)
- Mais potentielle ambiguité lors du décodage de chaînes (cfr exemple ci-dessous)

Codes à longueur variable : exemple

- Considérons $S = \{a, b, c\}$, $C = \{0, 1\}$ et

$$\begin{cases} K(a) = 0 \\ K(b) = 1 \\ K(c) = 01 \end{cases}$$

- Le décodage de *chaînes* peut être ambigu
  - La chaîne 01 peut se décoder comme *ab* ou comme *c*
  - Le code n'est pas univoque (pas inversible)
    (sauf si on restreint le codage à un symbole unique)
  - Problème vient du fait que $K(a)$ est un *préfixe* de $K(c)$

Code sans préfixe

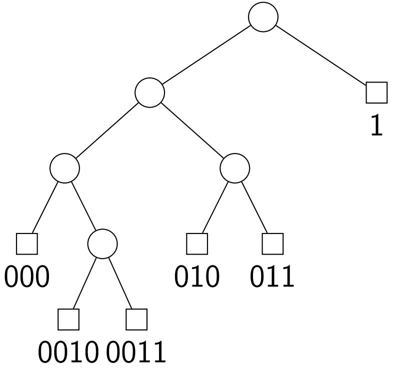
- Code dans lequel aucun mot du code n'est le préfixe d'un autre mot du code
- Permet le décodage à la volée (comme pour les codes en blocs)
- Peut être représenté sous forme d'arbre

Arbre associé à un code sans préfixe

- Chaque sommet est associé à un mot de C^*
- Mots du code sont les feuilles de l'arbre
- Racine est le mot vide
- Descendants d'un noeud sont tous les mots ayant ce noeud en préfixe

Arbre de code : exemple

$$K = \{1, 010, 011, 000, 0010, 0011\}$$



Arbre de code : propriétés

- Degré (arity) de l'arbre est $\leq r$
- Hauteur de l'arbre est la longueur maximale du code
- Pouvoir représenter un code sous forme d'arbre de code est une propriété nécessaire et suffisante au fait qu'il s'agisse d'un code sans préfixe

Inégalité de Kraft et Théorème de McMillan

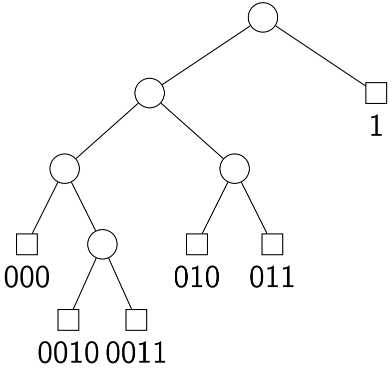
- **Inégalité de Kraft** (1948)
  Pour tout ensemble de longueurs de mots du code $\{\ell_1, \ell_2 \ldots \ell_q\}$ donné, **il existe (au moins) un code univoque sans préfixe** correspondant si et seulement si l'inégalité suivante, dite inégalité de Kraft, est satisfaite.

$$\sum_{i=1}^{q} r^{-\ell_i} \leq 1$$

- **Théorème de McMillan** (1956)
  Tout code univoque satisfait l'inégalité de Kraft

Intuition

- Code sans préfixe
  si et seulement si arbre
  de code correspondant
- Assigner un poids $r^{-\ell_i}$
  à chaque feuille au
  niveau ℓ_i
- Pour chaque sommet,
  assigner un poids égal
  à la somme des poids
  des descendants directs



Démonstration (inégalité de Kraft)

- Fixons $r = 2$ (cas général similaire)
- Supposons (sans perte de généralité) $\ell_1 \leq \ell_2 \leq \ldots \leq \ell_q$
- Pour $q = 1$, on a bien $2^{-\ell_1} \leq 1$, et le code $C = \{00\ldots0\}$ (avec ℓ_1 zeros) satisfait cette inégalité
- Par induction si $q > 1$.
  Soit $K(s_i)$ fixé pour $i = 1, \ldots, q-1$, sans préfixe, et comptons les options pour $K(s_q)$
  - Nous avons au plus 2^{ℓ_q} mots possibles
  - Chaque autre mot du code $K(s_i)$ exclut $2^{\ell_q - \ell_i}$ mots

Démonstration (inégalité de Kraft)

- On a donc comme condition nécessaire et suffisante

$$1 \leq \#\text{valeurs pour } K(s_q) = 2^{\ell_q} - \sum_{i=1}^{q-1} 2^{\ell_q - \ell_i}$$

$$1 \leq 2^{\ell_q} \cdot \left(1 - \sum_{i=1}^{q-1} 2^{-\ell_i}\right)$$

$$2^{-\ell_q} \leq 1 - \sum_{i=1}^{q-1} 2^{-\ell_i}$$

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

Borne de Kraft atteinte ssi “localement complet”

- Egalité atteinte si et seulement si l'arbre du code est *localement complet* de degré r : tous les noeuds internes (càd pas les feuilles) de l'arbre ont exactement r fils
- Preuve : associer à chaque sommet la somme des contributions $r^{-\ell_i}$ des mots de code (= feuilles) descendants

Théorème de McMillan (1956)

- **Tout code univoque satisfait l'inégalité de Kraft** (y compris les codes univoques avec préfixe)
- Corollaire : pour tout code univoque,

$$\ell_{\max} \geq \lceil \log_r q \rceil \geq \log_r q$$

(preuve : $1 \geq \sum_{i=1}^{q} r^{-\ell_i} \geq q.r^{-\ell_{\max}}$, donc $r^{\ell_{\max}} \geq q$)

McMillan vs Kraft

- Inégalité de Kraft : pour tout ensemble de longueurs de mots du code $\{\ell_1, \ell_2 \ldots \ell_q\}$, **il existe (au moins) un code univoque sans préfixe** correspondant si et seulement si l'inégalité de Kraft est satisfaite
- Théorème de McMillan : **tout code univoque** satisfait l'inégalité de Kraft

- "tout code" vs "il existe un code"
- "code univoque" vs "code univoque sans préfixe" (code $\{1, 10\}$ pas sans préfixe mais néanmoins univoque)

Preuve du Théorème de McMillan

- Soit $c = \sum_{i=1}^{q} r^{-\ell_i}$. On va montrer $\lim_{n\to\infty} \frac{c^n}{n} < \infty$
- On a

$$
\begin{aligned}
c &= \sum_{i=1}^{q} r^{-\ell_i} \\
c^2 &= \left(\sum_{i_1=1}^{q} r^{-\ell_{i_1}}\right)\left(\sum_{i_2=1}^{q} r^{-\ell_{i_2}}\right) = \sum_{i_1=1}^{q}\sum_{i_2=1}^{q} r^{-\left(\ell_{i_1}+\ell_{i_2}\right)} \\
&\vdots \\
c^n &= \sum_{i_1=1}^{q}\sum_{i_2=1}^{q}\dots\sum_{i_n=1}^{q} r^{-\left(\ell_{i_1}+\ell_{i_2}+\dots+\ell_{i_n}\right)}
\end{aligned}
$$

Preuve du Théorème de McMillan (suite)

- On a
$$c^n = \sum_{i_1=1}^{q}\sum_{i_2=1}^{q}\dots\sum_{i_n=1}^{q} r^{-\left(\ell_{i_1}+\ell_{i_2}+\cdots+\ell_{i_n}\right)}$$
- Considérons un message de longueur n : son code $K(s_{i_1}s_{i_2}\dots s_{i_n})$ est de longueur $j = \ell_{i_1} + \ell_{i_2} + \cdots + \ell_{i_n}$
- Soit $\mu_j(n)$ le nombre de tous les messages de longueur n produisant un code de longueur j
- Soit $\mu_j = \sum_{n=1}^{j} \mu_j(n)$ le nombre de tous les messages de longueur quelconque produisant un code de longueur j
- On a $\mu_j \leq r^j$ (car code univoque)
- Donc
$$c^n = \sum_{j=1}^{n\cdot\ell} \mu_j(n)\cdot r^{-j} \leq \sum_{j=1}^{n\cdot\ell} \mu_j \cdot r^{-j} \leq \sum_{j=1}^{n\cdot\ell} r^j\cdot r^{-j} = n\cdot\ell$$

Questions ?

?

Crédits et remerciements

- Mes transparents suivent fortement les notes de cours développées par le Professeur Yves Roggeman pour le cours INFO-F303 à l'Université libre de Bruxelles
- Une partie des transparents et des exercices ont été repris ou adaptés des transparents développés par le Professeur Jean Cardinal pour ce même cours
- Je remercie chaleureusement Yves et Jean pour la mise à disposition de ce matériel pédagogique, et de manière plus large pour toute l'aide apportée pour la reprise de ce cours
- Les typos et erreurs sont exclusivement miennes (merci de les signaler !)

Corrections par rapport au cours

- Un code univoque est un code pour lequel la fonction de codage est inversible
- La "fonction de codage" K est d'abord définie sur des symboles uniques, mais ensuite étendue à des suites de symboles
- Dans la version précédente des slides, j'avais défini "univoque" uniquement à partir de la fonction de codage de base (pas l'étendue). Définir "univoque" à partir de la fonction de codage étendue mène à une définition plus restrictive d'un code univoque (i.e. incluant moins de codes, et en particulier pas celui qui nous posait problème en cours)

Corrections par rapport au cours

- Le syllabus utilise la version restrictive de la définition (i.e. pour la fonction de codage étendue).
  L'énoncé du théorème I.2.3 doit être compris comme cela
- Le théorème de McMillan ne concerne que les codes univoques au sens de la fonction de codage étendue ; pas tous les codes univoques uniquement au sens de la fonction de codage d'un seul symbole
  (comme suggéré au cours dans un premier temps)
- Avec l'autre définition, le résultat n'est pas correct : on a vu un contre-exemple au cours. Comme soupçonné lors du cours, l'inégalité $\mu_j \leq r^{-j}$ intervenant dans la preuve n'est alors plus vérifiée

Corrections par rapport au cours

- Pour compléter cette discussion, notez que le théorème de McMillan est plus général que celui de Kraft : d'une part les codes univoques sont plus généraux que les codes univoques sans préfixe (même pour la définition restrictive, prenez par exemple $K(S) = \{1, 10\}$) ; et d'autre part le théorème de Kraft garantit uniquement l'existence d'un code avec les propriétés données, là où le théorème de McMillan concerne tous les codes univoques.
- J'ai corrigé les transparents pour refléter les observations ci-dessus et être cohérent avec le syllabus.

Corrections par rapport au cours

- Toutes mes excuses pour cette confusion.
- Merci à tous d'avoir travaillé ensemble à résoudre la contradiction apparue lors du cours. Je pense personnellement que cet épisode m'a / nous a permis de mieux comprendre la preuve que en la lisant de manière passive.
- N'hésitez pas à me faire part encore de vos observations sur des typos et erreurs éventuelles additionnelles !