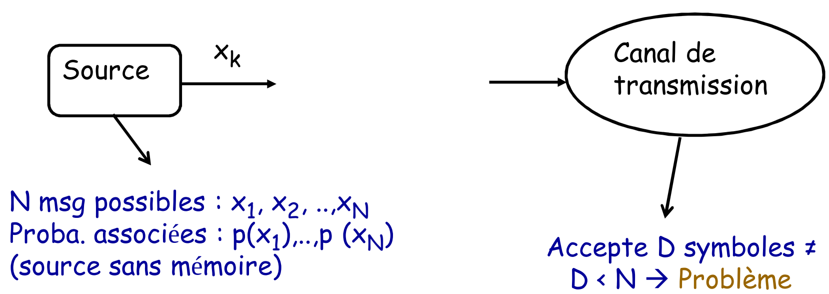
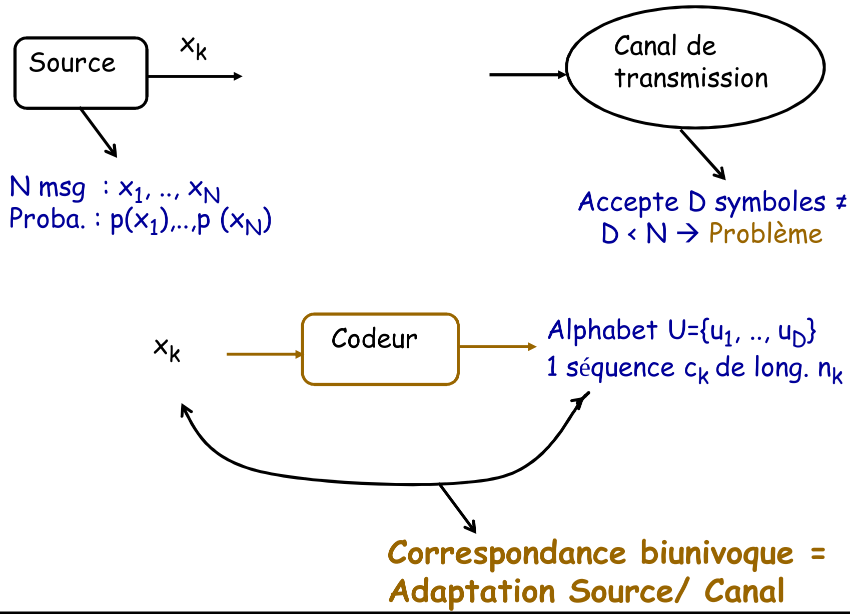
Codage Source

M. Belkasmi

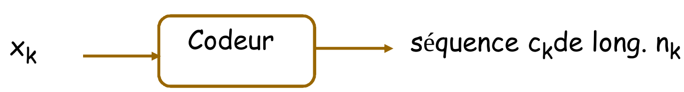
ENSIAS 2012-2013

Problématique





Remarque



- Quand nk ↗ → on peut toujours coder les N symboles émis par la source
- Or codage efficace ➔ les séquences c_k les plus courtes possibles

Efficacité du codeur

Désignons par $\bar{n}$ le nombre moyen de symboles {u1,..,uD} nécessaire par message de la source :

$$\bar{n} = \sum_{k=1}^{N} n_k p(x_k)$$

Le codeur doit être conçu de telle sorte a ce que $\bar{n}$ soit le plus faible possible.

Efficacité du codeur

Par définition l'efficacité du codeur a pour expression :

$$E = \frac{H(X)}{\bar{n} \log D}$$

Il nous faut maintenant trouver des codes efficaces. Comment on va y arriver ?

CONSTRUCTION DE CODES

Définition (Codes Séparables et codes Irréductibles)

L'utilisation de code en bloc (quand les séquences ck ont la même longueur n) ne pose pas de problème particulier pour le récepteur pour retrouver les messages émis.

Si longueurs variables : l'utilisateur risque de ne pas savoir faire la séparation des messages.

CONSTRUCTION DE CODES

Exemple (D=2)

Supposons x1--->1101, x2--->110110, x3--->101101,...

Lorsque l'utilisateur aura reçu 1101101101…

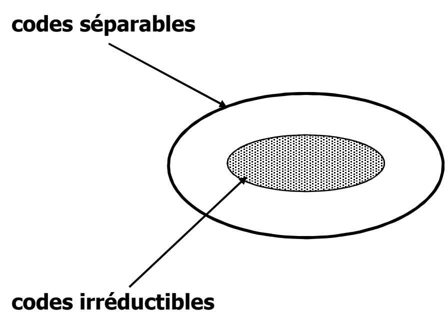
------> interprétation x1x3 ? ou x2x1 ?

Les codes qui permettent une telle séparation sans ambiguïté s'appellent des codes séparables

Définitions :

- Condition Suffisante : Un code est séparable si aucune séquence n'est le début d'une séquence plus grande.

- Un tel code est dit obéir à la propriété du préfixe et s'appelle code irréductible (ou code préfixe).



Exemple de code séparable non irréductible :

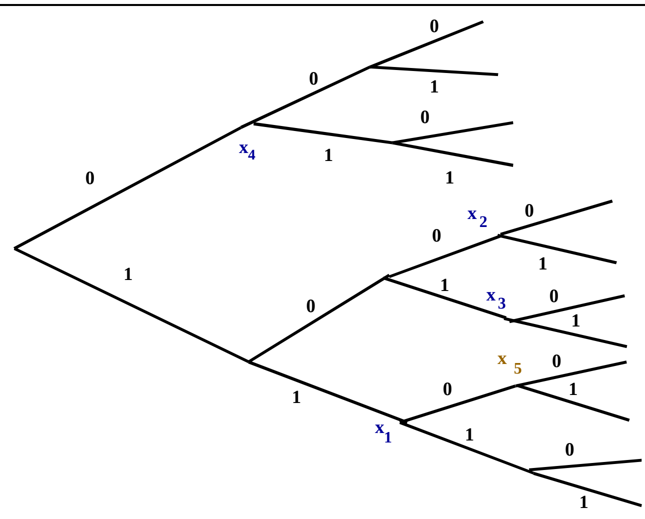
N = 3, D = 2

x1 --->1
x2 --->10
x3 --->100

Méthode de l'arbre

- Il est commode de représenter les codes irréductibles par une méthode graphique illustrée par l'exemple suivant (D=2) :
- Considérons le code avec les messages suivants :

x1 ---> 11
x2 ---> 100
x3 ---> 101
x4 ---> 0



Arbre correspondant au code de l'exemple (points en bleu)

- Sur l'arbre sont portés les messages x1, x2, x3 et x4 du code. Mais :

- Seuls les nœuds terminaux de l'arbre ↔ mots de code = (séquences ck).
- Si ajout de x5 ---> 110 sur l'arbre ➔ x1 non terminal : risque de confusion.

- Cas général de D digits ➔ chaque nœud admet D fils.

Inégalités de Kraft et Mac Millan

On doit coder N messages x1, x2, ...,xN auxquels il faut faire correspondre N séquences (mots de code) de longueur nk.

Cas des codes irréductibles :

- Pour que le codage soit possible tout en satisfaisant la condition de préfixe, nous avons vu que ces N messages doivent correspondre à N nœuds terminaux de l'arbre.

Inégalités de Kraft et Mac Millan

- Kraft a montré qu'une condition nécessaire et suffisante pour pouvoir assigner N nœuds terminaux de longueur nk aux N messages {xk } s'écrivait :

$$\sum_{k=1}^{N} D^{-n_k} \leq 1 \quad (I)$$

- L'égalité $\longleftrightarrow$ tous les nœuds terminaux sont utilisés ($=^{def}$ l'arbre est complet).

Inégalités de Kraft et Mac Millan

Cas général

Mac Millan a démontré que l'inégalité de Kraft était encore valable pour les codes simplement séparables (là aussi c'est une Condition Nécessaire et Suffisante).

Résultat :

On peut alors limiter l'étude du $\bar{n}$ optimum aux codes irréductibles,

en effet :

- Étant donné un code simplement séparable, les longueurs nk des mots constitutifs de ce code doivent satisfaire à l'inégalité (I), d'après Mac Millan;
- D'après Kraft, il est alors possible de construire un code irréductible à partir de ces nk.
- → tout code simplement séparable peut donc être remplacé par un code irréductible avec le même $\overline{n}$.

Détermination de $\overline{n}$ optimal

On montre que :

1. la longueur moyenne $\overline{n}$ de tout code séparable vérifie :

$$\overline{n} \geq \frac{H(X)}{\log D}$$

2. il existe une limite inférieure pour $\overline{n}$, mais $\overline{n}$ n'atteint effectivement sa limite inférieure que dans certains cas particuliers.

Détermination de $\overline{n}$ optimal

3. il existe aussi une limite supérieure pour $\overline{n}$ optimal puisqu'il existe au moins un code irréductible avec $\overline{n}$ vérifiant :

$$\overline{n} < \frac{H(X)}{\log D} + 1$$

- Donc le $\overline{n}$ optimal satisfait la double inégalité :

$$\frac{H(X)}{\log D} \leq \overline{n} < \frac{H(X)}{\log D} + 1 \quad \text{(II)}$$

Détermination de $\overline{n}$ optimal

- Cas binaire (D=2) : on a alors :

$$H(X) \leq \overline{n} < H(X) + 1$$

- Contre exemple :
  Cas d'un code non séparable et $\overline{n}$ < H(X)

$$\begin{cases} x_1 \rightarrow 1 \quad et \quad p_1 = 0.4 \\ x_2 \rightarrow 0 \quad et \quad p_2 = 0.4 \\ x_3 \rightarrow 100 \quad et \quad p_3 = 0.2 \end{cases}$$

- On trouve alors :
  H(X) = 0.529 +0.529 + 0.464 = 1.522
  et $\overline{n}$ = 0.4 + 0.4 + 0.6= 1.4

Méthode de l'arbre(suite)

- Construire petit à petit un arbre qui correspond au code
- Exploiter l'idée suivante :
  Choisir les nk vérifiant :

$$\frac{-\log p_k}{\log D} \leq n_k < \frac{-\log p_k}{\log D} + 1$$

Exemple : Construction de code optimal

- cas particulier : D=2 , les p(xk) sont des puissances négatives de 2.
- 8 messages [X] = (x1,....,x8) P = (1/4, 1/4, 1/8, 1/8, 1/16,....,1/16).
- Calculons I(xk) = - log p(xk) pour chaque valeur de k.
- Dans ce cas on trouve des entiers : I(x1) = I(x2) = 2, I(x3) = I(x4) = 3, I(x5) = ...= I(x8) = 4.

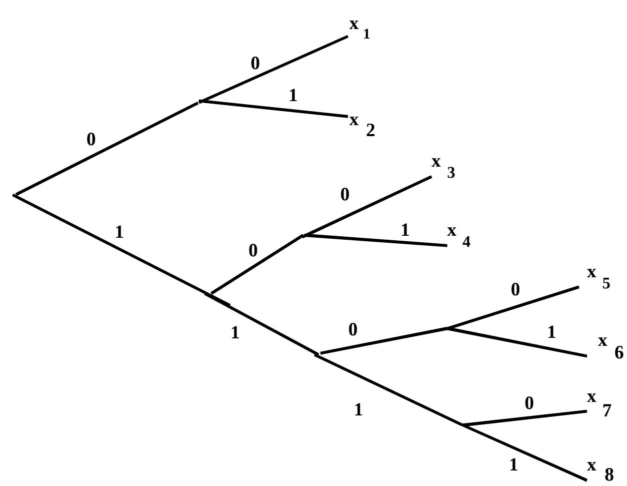
Exemple (suite)

- Si choix nk = I(xk) → $\overline{n}$ = H(X) : borne inférieure atteinte
- De plus p(xk) = 2^{-nk}

$$\rightarrow \sum_{k=1}^{8} 2^{-n_k} = 1$$

→ arbre correspondant complet.

Exemple (suite)



Exemple (suite)

- **Codage :**

x1 ---> 00	**x5 ---> 1100**
x2 ---> 01	**x6 ---> 1101**
x3 ---> 100	**x7 ---> 1110**
x4 ---> 101	**x8 ---> 1111**

ALGORITHMES DE SHANNON-FANO ET DE HUFFMAN

Pour ces deux algorithmes les entrées sorties sont comme suit :

Entrée : Un ensemble de messages avec les probabilités d'apparition correspondante.

Sortie : mots de code correspondant aux messages.

METHODE DE SHANNON-FANO

La méthode consiste en un classement des messages suivant l'ordre décroissant des probabilités.

On partage ensuite les messages en deux groupes de probabilités aussi voisine que possible l'une de l'autre ---> S0 et S1.

Refaire la même chose sur chaque groupe obtenu jusqu'à ce que l'on obtienne des singletons.

Exemple

- Soit une source avec 5 messages x1, … ,x5 avec les probabilités suivantes :

  p1 = 0,35 p2= 0,22 p3= 0,18
  p4= 0,15 p5= 0,10

- D'où l'entropie de cette source est H = 2,1987 bit/message.

Exemple (suite)

					codage
p1	0,35	S_0 (0,57)	S_{00} (0,35)		00
p2	0,22		S_{01} (0,22)		01
p3	0,18		S_{10} (0,18)		10
p4	0,15	S_1 (0,43)	S_{11} (0,25)	S_{110} (0,15)	110
p5	0,10			S_{111} (0,10)	111

La Longueur moyenne est $\overline{n}$ = 2,25 e.b.
et l'efficacité de ce codage est E = 0,977 bit/ e.b.

Algorithme de Huffmann

1. Mettre dans une première colonne les différents messages en suivant l'ordre décroissant pour leurs probabilités

2. Remplir la seconde colonne par les probabilités des messages.

3. Pour remplir la 3ème colonne, on fait la somme des deux probabilités les plus faibles de la 2ème colonne (un nouveau symbole est ainsi crée ayant comme probabilité la somme des probabilités). On classe de nouveau les probabilités.

Algorithme de Huffmann (suite)

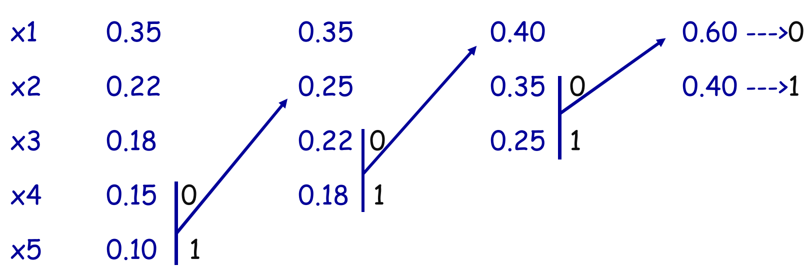
4. On remplit ainsi toutes les colonnes jusqu'à ce que l'on obtienne une colonne avec deux probabilités seulement.

   On associe ensuite aux deux probabilités les plus faibles de chaque colonne, les valeurs 0 et 1.

   Le mot de code associe à un message est obtenu en partant du message et en suivant les symboles qu'il a généré. Les mots de code sont à écrire de la droite vers la gauche.

Algorithme de Huffmann

- Exemple : la même source



Algorithme de Huffmann

- Les mots de code :

x1	00
x2	10
x3	11
x4	010
x5	011

- La longueur moyenne est $\bar{n}$ = 2,25 e.b.
- L'efficacité de ce codage est E= 0,977 bit/e.b.

Premier théorème fondamental de Shannon

- Dans le cas d'une source sans mémoire, il est possible de réduire la longueur moyenne des messages.
- Plus précisément, on peut montrer que la limite inférieure de la formule (II) peut toujours être atteinte.
- C-à-d qu'on montre qu'il est effectivement possible de trouver un codage optimal (un codage dont l'efficacité est 1).

Premier théorème fondamental de Shannon

- Démonstration utilisant l'extension d'ordre K d'une source qu'on se donne.

Soit une source sans mémoire [X] = (x1,........xN). Considérons maintenant la source $X^2 = X \otimes X$, c.à.d la source capable de transmettre N^2 messages. X^2 constitue l'extension à l'ordre 2 de la source d'origine.