Réseaux, information et communication

Entropie d'une source aléatoire
1er théorème de Shannon : codage parfait
Compression sans pertes

Yves Roggeman
FACULTÉ DES SCIENCES - DÉPARTEMENT D'INFORMATIQUE
Boulevard du Triomphe - CP212
B-1050 Bruxelles (Belgium)
Tél. : +32-2-650 5598
E-mail : `yves.roggeman@ulb.ac.be`

Année 2015–2016

ULB

UNIVERSITÉ LIBRE DE BRUXELLES
yves.roggeman@ulb.ac.be

Compression

- **Idée : Extension de la source**
  - **Travailler par blocs de *n* symboles**
  - **Utiliser le code optimal (Huffman) sur ces blocs**
  - **$\Rightarrow$ Longueur <u>par symbole</u> décroît $\Rightarrow$ limite ($n \to \infty$) ?**
- **Exemple : 1 → 0.844 → 0.823 → ... (→ .8113)**

4/4	**0**	**1**						
Proba	3/4	1/4						
Code	0	1						
27/32	**00**	**01**	**10**	**11**				
Proba	9/16	3/16	3/16	1/16				
Code	0	11	100	101				
158/192	**000**	**001**	**010**	**100**	**011**	**101**	**110**	**111**
Proba	27/64	9/64	9/64	9/64	3/64	3/64	3/64	1/64
Code	1	001	010	011	00000	00001	00010	00011

ULB

UNIVERSITÉ LIBRE DE BRUXELLES

yves.roggeman@ulb.ac.be

(Quantité d') Information

- **Information de s_i de proba p_i : $I(p_i)$**
  - **$I(p) \geq 0$ et $I(p) \not\equiv 0$**
  - **$I(p_1.p_2) = I(p_1)+I(p_2)$ ← ≈ arbitraire, mais...**
  - **$I(p)$ continue en p**
- **Théorème de Shannon**
  - **$I(p) = -k.\ln(p) = -\log_b(p) = \log_b(1/p)$ [$p > 0$] avec $k > 0$ ou $b > 1$**
  - **Démonstration... [$I(p^t) = t\, I(p)$]**
  - **Propriétés : $I(1) = 0$ et monotone**

Entropie d'une source

- **Définition : « information moyenne »**
  - $H(S) = H(p_1, p_2, \dots p_q) = H(\mathbb{P}) = \mathscr{E}_{\mathbb{P}}[I(p_i)]$
  - $H(S) = -\sum p_i \log_b p_i$ où $0.\log(0) = 0.\log(1/0) = 0$
- **Propriétés**
  - $H(S) \geq 0$; H(S) continue, symétrique…

    \+ « Cohérence » :

$$H(p_1 \dots p_q) = H((p_1+p_2) p_3 \dots p_q) + (p_1+p_2) \cdot H\left(\frac{p_1}{p_1+p_2}, \frac{p_2}{p_1+p_2}\right)$$

- **Normalisation : définition du « bit »**
  - Poser b = r = |C| ; 2 en binaire : $I(p) = -\log_2 p$
  - Ainsi bit = H(½, ½) = 1 = pile ou face (idem si q = r)

yves.roggeman@ulb.ac.be

Entropie minimale et maximale

- **Min : $H(S) = 0 \Leftrightarrow \exists j : p_j = 1$**
  - **Donc autres $p_i = 0$, car $\sum p_i = 1$**
  - **Démonstration : $-p_i \log p_i \geq 0$ ...**
- **Max : $H(S) = \log_r q \Leftrightarrow \forall i : p_i = 1/q$**
  - **Lemme : $\log x \leq x-1$**
  - **Démonstration :**
    - $\Leftarrow$: Direct
    - $\Rightarrow$: $\log_r q - H(S) = \ldots \geq 0$ et égalité si...

Symb	Moy	A	B	C	D	E	F
Proba		12/32	6/32	5/32	4/32	3/32	2/32
Info	2.347	1.415	2.415	2.678	3.000	3.415	4.000
Sh-F	2.438	2	2	3-2	3	3-4	3-4
Huff	2.406	1	3	3	3	4	4

Extension de la source

- **Définition (cf. compression)**
  - Étant donné une source S, regrouper en bloc de n symboles : S^n
  - $\mathbb{P}[s_{i1}s_{i2}\ldots s_{in}] = \mathbb{P}[s_{i1}]\mathbb{P}[s_{i2}]\ldots\mathbb{P}[s_{in}] = p_{i1}\, p_{i2} \ldots p_{in}$
- **Théorème 1**
  - $H(S^n) = n\, H(S)$
  - Démonstration...
  - Donc : par symbole, ça ne change pas...
- **Théorème 2 : c'est une borne inférieure**
  - $\forall\, K : L(K) \geq H(S)$
  - Démonstration...

1^{er} théorème de Shannon : codage parfait de la source

- *Shannon's Noiseless Coding Theorem*
- Lien entre les 2 théorèmes précédents
  - **$H(S) \leq L_{min}(S) \leq H(S)+1$**
    - $= H(S)+1$ seulement si $p_0 = 1$ & $p_1 = 0$
  - $$\lim_{k \to \infty} \frac{L_{min}\left(S^k\right)}{k} = H\left(S\right)$$
  - **Démonstration…**
- Compression maximale :
  - **Message de *n* symboles (de S) en *n* H(S) bits**

UNIVERSITÉ LIBRE DE BRUXELLES
yves.roggeman@ulb.ac.be

Codes optimaux

- **Huffman**
  - $\exists\, n_0, \forall\, n : n > n_0 \Rightarrow L(S^n) = n\, H(S)$
  - **Pas de meilleure compression si *n* grand**
  - **Mais il faut connaître les p_i !**
- **Shannon-Fano ?**
  - **Il existe un code avec $\lceil I(p_i) \rceil = \ell_i$**
  - **Shannon-Fano fait mieux : $\ell_i - 1 \leq I(p_i) \leq \ell_i + 1$**
  - **Huffman fait encore mieux**
    - Réduit ℓ_i des plus probables

ULB

UNIVERSITÉ LIBRE DE BRUXELLES

yves.roggeman@ulb.ac.be

Compression sans perte

- **Comment estimer les p_i à la volée ?**
- **Codes par dictionnaire**
  - **Liv-Zempel (LZ77) : fenêtre glissante**
    - Symbole = référence à occurrence précédente
    - « Deflate » → NTFS
    - Aussi LZMA (chaîne de Markov)
    - LZ78 (breveté) : dict. global → ZIP...
  - **~-Welch (LZW 84, breveté) → GIF, TIFF, PNG...**
    - Dict. reconstruit à la volée au décodage

yves.roggeman@ulb.ac.be

Algorithme LZW

- **Encodage**
  - **Initialiser**

    Dictionnaire = {caractères}

    w = $\varnothing$
  - **Algorithme :**

    si w+c $\in$ Dictionnaire

        w = w+c

    sinon

        w+c $\rightarrow$ Dictionnaire

        output(index(w))

        w = c
  - **Code : accroître quand |Dictionnaire| $\geq 2^k$**
- **Décodage : idem**

ULB
UNIVERSITÉ LIBRE DE BRUXELLES
yves.roggeman@ulb.ac.be

Codes adaptatifs

- **Faller-Gallager-Knuth (FGK 85), Vitter (87)**
  - **Arbre pondéré et labellisé (№ de création) + feuille « Vide » (poids nul), Bloc = {même poids}**
  - **Algorithme Λ de Vitter (linéaire) : Encode ≡ Décode**

```
Chercher c
output(code(c))
Si c = Vide
      output('c') // en « clair »
      éclater Vide en 2 nouveaux fils : d = 'c' et g = Vide
      poids[père] = poids[c] = 1
      c = père[c]
Ttq c ≠ racine
      si c ≠ 1er(Bloc) ≠ père : swap(c,1er)
      ++ poids[c]
      c = père[c]
```