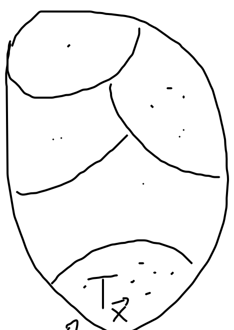
Codice di Shannon-Fano (asintoticamente ottimo; non è ottimo per n finito)
Codice di Fano (può dare talvolta risultati migliori per n fissato)
Codice di Huffman (è ottimo per ogni n finito)
} Basati su una descrizione statistica della sorgente $\mathcal{S}$

Possiamo avere codici efficienti <u>senza</u> conoscere la d.p. di $\mathcal{S}$?

$\mathcal{A} = \{x_1, \dots, x_K\}$ Una stringa $\vec{x} = (x_{i_1}, x_{i_2}, \dots, x_{i_n})$ ha tipo esatto $(n_1, n_2, \dots, n_K)$

dove $n_i = \#$ occorrenze di x_i in $\vec{x}$ (es.: $\mathcal{A} = \{a, b\}$, $\vec{x} = aba$, tipo $(2,1)$)

$\mathcal{A}^n$ ($|\mathcal{A}^n| = K^n$)



Il tipo esatto di $\vec{x}$

$\mathcal{A}^n$ è partizionato in classi di equivalenza date dai tipi esatti.

Il numero di classi di equivalenza è denotato con Γ_n. $\Gamma_n \leq (n+1)^K$

$\rightarrow \Gamma_n \leq (n+1)^K$ in quanto $0 \leq n_i \leq n$ (per ogni $i = 1 \dots K$). (Nota: ovviamente $\sum_{i=1}^{K} n_i = n$)

Denoto con $T_{\vec{x}}$ il tipo esatto della n-pla $\vec{x}$.

$$|T_{\vec{x}}| = \binom{n}{n_1, n_2, \dots, n_K} = \frac{n!}{n_1! \, n_2! \cdots n_K!}$$

Coefficiente multinomiale

Codifica multinomiale: per codificare una n-pla $\vec{x}$, si usa la parola a lungh. variabile

$$\varphi(\vec{x}) = \underbrace{\varphi_p(T_{\vec{x}})}_{\substack{\text{prefisso} \\ \text{(dipende solo} \\ \text{dal tipo)}}} * \underbrace{\varphi_s(\vec{x})}_{\substack{\text{suffisso} \\ \text{(dipende da } \vec{x})}}$$

- $\varphi_p(T_{\vec{x}})$ è il numero d'ordine (lessicografico) di $T_{\vec{x}}$ (identifica il tipo)
- $\varphi_s(\vec{x})$ è il numero d'ordine di $\vec{x}$ all'interno di $T_{\vec{x}}$ (identifica $\vec{x}$ all'interno del suo tipo)

Chiamiamo: $\ell_p = |\varphi_p|$ la lunghezza del prefisso (costante)

$\ell_s(\vec{x}) = |\varphi_s(\vec{x})|$ la lunghezza del suffisso (variabile)

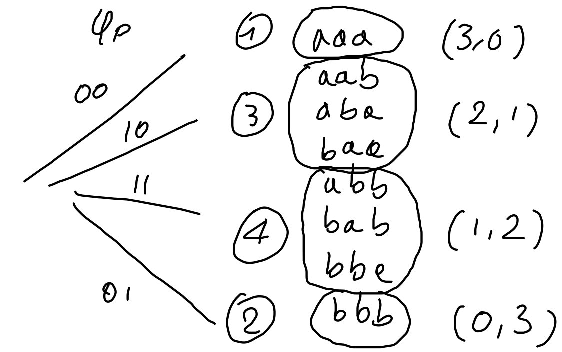
→ è una codifica B-LV

Deve essere $D^{\ell_p} \geq \Gamma_n \Rightarrow \ell_p = \lceil \log_D \Gamma_n \rceil$

$D^{\ell_s(\vec{x})} \geq |T_{\vec{x}}| \Rightarrow \ell_s(\vec{x}) = \lceil \log_D |T_{\vec{x}}| \rceil$

Esempio: $A = \{a, b\}$, $B = \{0, 1\}$ $(D=2;\ K=2)$; $n = 3$.

$\Gamma_3 = 4$ perché ho 4 tipi esatti ; $\lceil \log_2 4 \rceil = 2$



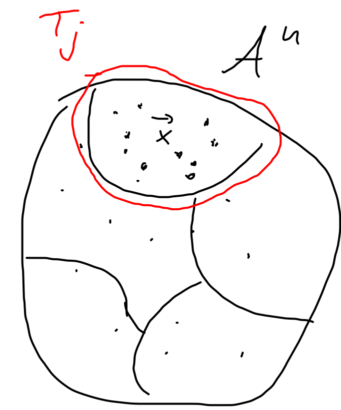
Z		Tipo	n-ple	$l_S(\vec{x})$	φ_P	$\varphi_S(\vec{x})$	$\varphi(\vec{x})$
						(stringa vuota)	
1		(3,0)	aaa	$\lceil \log_2 1 \rceil = 0$	00	—	00
2	→	(0,3)	bbb	$\lceil \log_2 1 \rceil = 0$	01	—	01
3	→	(2,1)	aab	$\lceil \log_2 3 \rceil = 2$	10	00	1000
		=	aba	=	10	01	1001
		=	baa	=	10	10	1010
4		(1,2)	bba	$\lceil \log_2 3 \rceil = 2$	11	00	1100
		=	bab	=	11	01	1101
		=	abb	=	11	10	1110

Codice multinomiale non è ottimo per n finito

ma è asintoticamente ottimo $(n \to \infty)$

Cosa succede asintoticamente? $(n \to \infty)$

$$E[L^{(n)}] = \sum_{\text{tutte le n-ple } \vec{x}} p(\vec{x})(\ell_p + \ell_s(\vec{x})) \overset{\sum p(\vec{x}) = 1}{=} \ell_p + \sum_{\vec{x}} p(\vec{x})\, \ell_s(\vec{x})$$



$$= \lceil \log_D \Gamma_n \rceil + \sum_{j=1}^{\Gamma_n} \sum_{\vec{x} \in T_j} p(\vec{x}) \cdot \overbrace{\lceil \log_D |T_j| \rceil}^{\text{non dipende da } \vec{x}}$$

$$= \lceil \log_D \Gamma_n \rceil + \sum_{j=1}^{\Gamma_n} \lceil \log_D |T_j| \rceil \cdot \Big(\underbrace{\sum_{\vec{x} \in T_j} p(\vec{x})}_{Pr(T_j)} \Big)$$

Probabilità complessiva del tipo T_j: $Pr[\vec{X} \in T_j]$

$$= \lceil \log_D \Gamma_n \rceil + \sum_{j=1}^{\Gamma_n} Pr(T_j) \cdot \lceil \log_D |T_j| \rceil$$

$$< 1 + \log_D \Gamma_n + 1 + \sum_{j=1}^{\Gamma_n} Pr(T_j) \log_D |T_j| \quad \overset{?}{\longleftrightarrow} \quad H(S)$$

$$< 2 + K \log_D (n+1) + \sum_{j=1}^{\Gamma_n} Pr(T_j) \log_D |T_j|$$

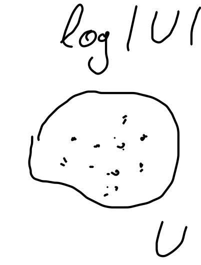
$\underbrace{\lceil x \rceil < 1 + x}$

$\Gamma_n \leq (n+1)^K$

Introduco una v.a. Z intera che rappresenta il numero d'ordine del tipo della seq. $\vec{X}$

Consideriamo l'entropia condizionata:

$$H(\vec{X} \mid Z) = \sum_{j=1}^{r_n} Pr[Z=j]\, H(\vec{X} \mid Z=j) = \sum_j Pr(T_j)\, H(\vec{X} \mid Z=j)$$



H( )

Tutte le n-ple di tipo esatto j sono equiprobabili, quindi

$$H(\vec{X} \mid Z=j) = \log_D |T_j| \quad \Rightarrow H(\vec{X} \mid Z) = \sum_j Pr(T_j) \log_D |T_j|$$

$$\rightarrow E[L^{(n)}] < 2 + K \log_D(n+1) + H(\vec{X} \mid Z)$$
$$\leq 2 + K \log_D(n+1) + H(\vec{X})$$
$$= 2 + K \log_D(n+1) + n H(S)$$

(in ipotesi di stazionarietà e assenza di memoria)

Il tasso è $\dfrac{E[L^{(n)}]}{n} < \dfrac{2}{n} + K \dfrac{\log_D(n+1)}{n} + \dfrac{\cancel{n}}{\cancel{n}} H(S) \xrightarrow{n \to \infty} H(S)$

quindi la codifica è asintoticamente ottima.

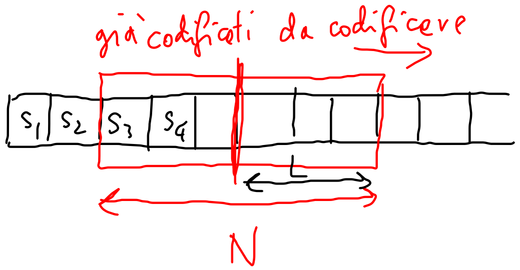
Codifica universale: <u>non</u> dipende dalla d.p. della sorgente

Codifica Ziv-Lempel (altro esempio di codice universale)

(LZ77) $\rightarrow$ numerose varianti (LZ77, LZ78, LZW, ...)
formati .zip, .gzip

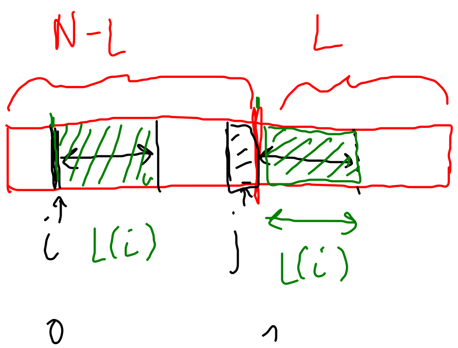
Codifica "strutturale": basata sulla struttura della concatenazione dei simboli

Stringa $s = (s_1, s_2, \dots)$ sequenza di simboli sull'alfabeto $\mathcal{A}$ $(|\mathcal{A}| = K)$



N: dimensione della finestra

$L\ (<N)$: numero di simboli massimo codificato ad ogni passo



Sia $s[i,j] = (s_i, s_{i+1}, \dots, s_j)$

Per ogni $i = 1, 2, \dots, N-L$ sia $L(i)$

il più grande intero tale che $s[i, i+L(i)-1] = s[j+1, \dots, j+L(i)]$