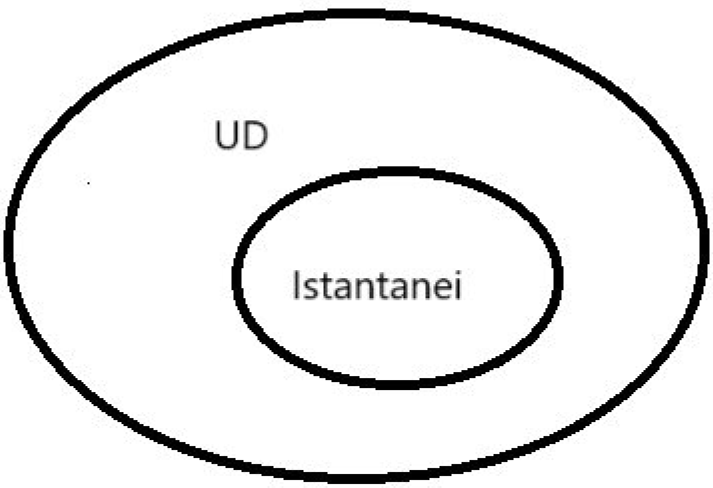
Teoria dell'Informazione

Simone Alessandro Casciaro
22 Ottobre 2024

Lezione 7: Disuguaglianza di Kraft-McMillan

Ripasso



I codici Istantanei sono tutti Univocamente Decodificabili. Tuttavia, ci chiediamo se esiste un codice Univocamente Decodificabile e non Istantaneo che, magari, migliora il valore atteso delle lunghezze.

Il nostro obiettivo è trovare un codice tale che

$$\begin{cases} \min\limits_{l_1,...,l_m} \sum_{i=1}^{m} p_i l_i & \min \mathbb{E}(l_c) \\ \sum_{i=1}^{m} d^{-l_i} \leq 1 & \text{Disuguaglianza di Kraft} \end{cases}$$

Si ricordano le seguenti nozioni:

$$
\begin{aligned}
c : \mathbb{X} \to D^+ \qquad & c \text{ non singolare} \\
C_n : \mathbb{X}^n \to D^+ \qquad & C_n \text{ non singolare} \\
C_n(x_1, \dots, x_n) = c(x_1) \dots c(x_n) & \\
l_{C_n}(x_1, \dots, x_n) = \sum_{i=1}^{n} l_c(x_i) \qquad & n \geq 1 \\
l_{max} = \max_{i=1,\dots,m} l_c(x_i) &
\end{aligned}
$$

Teorema di Kraft-McMillan

Siano $l_1, \dots, l_m$ le lunghezze di un codice $d-$ario Univocamente Decodificabile per una sorgente $< \mathbb{X}, \mathbb{P} >$ di m simboli $\iff \sum_{i=1}^{m} d^{-l_i} \leq 1$

Dimostrazione

$\boxed{\Longleftarrow}$

Per la disuguaglianza di Kraft, $\sum_{i=1}^{m} d^{-l_i} \leq 1 \implies \exists c$ istantaneo con lunghezze $l_1, \dots, l_m$.

Ma un codice istantaneo è sempre univocamente decodificabile.

$\boxed{\Longrightarrow}$

Considero un codice c univocamente decodificabile e calcolo $\left(\sum_{x \in \mathbb{X}} d^{-l_c(x)} \right)^n$

> Se $\boxed{n = 2}$
>
> $$\left(\sum_{i=1}^{m} a_i \right)^2 = \sum_{i=1}^{m} a_i * \sum_{j=1}^{m} a_j = \sum_{i=1}^{m} \sum_{j=1}^{m} a_i a_j$$

Nel caso generale,

$$
\begin{aligned}
\left(\sum_{x\in\mathbb{X}} d^{-l_c(x)}\right)^n &= \sum_{x_1\in\mathbb{X}}\cdots\sum_{x_n\in\mathbb{X}} d^{-l_c(x_1)}\dots d^{-l_c(x_n)} \\
&= \sum_{x_1\in\mathbb{X}}\cdots\sum_{x_n\in\mathbb{X}} d^{-\sum\limits_{i=1}^{n} l_c(x_i)} \\
&= \sum_{(x_1,\dots,x_n)\in\mathbb{X}^n} d^{-l_{C_n}(x_1,\dots,x_n)} \qquad \text{Passiamo alla sorgente } <\mathbb{X}^n,\mathbb{P}_n>
\end{aligned} \tag{1}
$$

Analizzando l_{C_n}, scopriamo che:

$n \leq l_{C_n}(x_1,\dots,x_n)$ perché se ad ogni x_i associo un solo simbolo nella codifica, abbiamo $\sum\limits_{i=1}^{n} 1 = n\backslash$

$l_{C_n}(x_1,\dots,x_n) \leq n * l_{max}$ perché se ad ogni x_i associo la lunghezza massima della codifica, abbiamo $\sum\limits_{i=1}^{n} l_{max} = n * l_{max}$

Quindi

$$n \leq l_{C_n}(x_1,\dots,x_n) \leq n * l_{max}$$

Consideriamo $\mathbb{X}^n$ e lo partizioniamo in k elementi a 2 a 2 disgiunti secondo la lunghezza delle codifiche dato da C_n

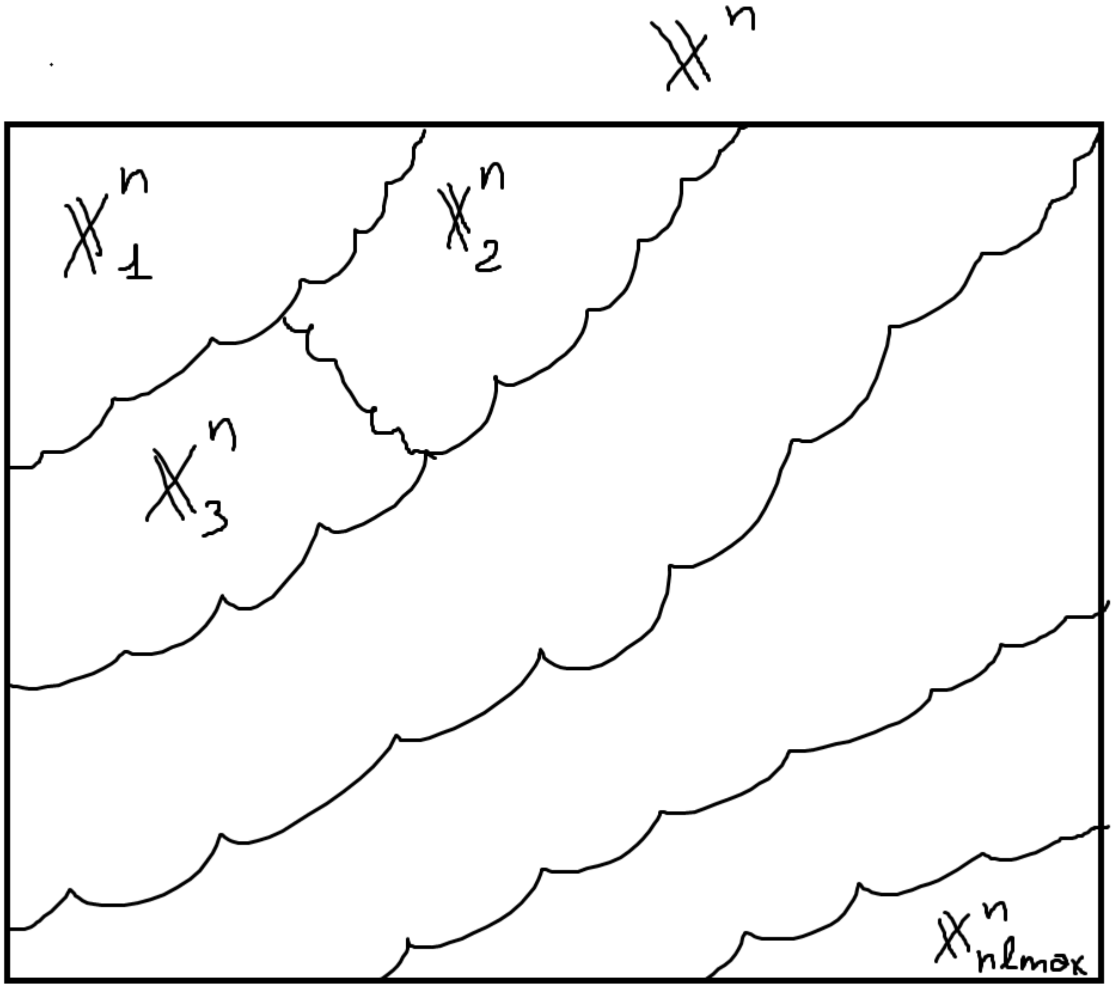
Chiamiamo ogni partizione $\mathbb{X}^n_k$

$\mathbb{X}^n_k = \{(x_1,\dots,x_n)\in\mathbb{X}^n : l_{C_n}(x_1,\dots,x_n) = k\}$

Data la premessa precedente, si ha che se $k < n$ oppure se $k > n * l_{max}$ allora $\mathbb{X}^n_k = \emptyset$

Gli insiemi $\mathbb{X}^n_k$ sono tali che:

- $\mathbb{X}^n_i \cap \mathbb{X}^n_j = \emptyset \quad \forall i \neq j$
- $\bigcap\limits_{k=n}^{n*l_{max}} \mathbb{X}^n_k = \mathbb{X}^n$
- $\mathbb{X}^n_k \subseteq \mathbb{X}^n$



Riprendendo il punto 1, si ha che

$$\begin{aligned}\left(\sum_{x\in\mathbb{X}} d^{-l_c(x)}\right)^n &= \sum_{(x_1,\dots,x_n)\in\mathbb{X}^n} d^{-l_{C_n}(x_1,\dots,x_n)} \\ &= \sum_{k=1}^{nl_{max}} \sum_{(x_1,\dots,x_n)\in\mathbb{X}_k^n} d^{-l_{C_n}(x_1,\dots,x_n)} \\ &= \sum_{k=1}^{nl_{max}} \sum_{(x_1,\dots,x_n)\in\mathbb{X}_k^n} d^{-k} \\ &= \sum_{k=1}^{nl_{max}} |\mathbb{X}_k^n| d^{-k}\end{aligned} \tag{1.1}$$

Dato che C_n è una funzione non singolare (e quindi iniettiva), la cardinalità del dominio è minore di quella del codominio. Quindi $\left|\mathbb{X}_k^n\right| \leq \left|D^k\right|$ Possiamo dunque maggiorare la sommatoria precedente.

$$\begin{aligned}\left(\sum_{x\in\mathbb{X}} d^{-l_c(x)}\right)^n &= \sum_{k=n}^{nl_{max}} |\mathbb{X}_k^n| d^{-k} \\ &\leq \sum_{k=1}^{nl_{max}} |D^k| d^{-k} \\ &= \sum_{k=1}^{nl_{max}} d^k d^{-k} \\ &= \sum_{k=1}^{nl_{max}} 1 \\ &= n * l_{max}\end{aligned} \tag{1.2}$$

Abbiamo dunque dimostrato che

$$\left(\sum_{x\in\mathbb{X}} d^{-l_c(x)}\right)^n \leq n * l_{max} \tag{1.2.1}$$

Facendo uno studio di funzione e rinominando $\sum_{x\in\mathbb{X}} d^{-l_c(x)} = M$

notiamo che la disuguaglianza al punto $1.2.1$ è rispettata se e solo se $M \leq 1$, quindi abbiamo dimostrato che

$$\sum_{x \in \mathbb{X}} d^{-l_c(x)} \leq 1 \quad \square$$

Esercizi

1. Data una codifica e 3 possibili spazi di probabilità, trovare il migliore supponendo che vogliamo minimizzare il valore atteso delle lunghezze.

	C_B	p_1	p_2	p_3
x_1	000	0.2	0.4	0.1
x_2	001	0.2	0.2	0.1
x_3	01	0.2	0.2	0.2
x_4	110	0.2	0.1	0.4
x_5	111	0.2	0.1	0.2

Calcolando i 3 valori attesi, si ha che:
$\mathbb{E}(l_1) = 0.2 * 3 + 0.2 * 3 + 0.2 * 2 + 0.2 * 3 + 0.2 * 3 = 2.8$
$\mathbb{E}(l_2) = 0.4 * 3 + 0.2 * 3 + 0.2 * 2 + 0.1 * 3 + 0.1 * 3 = 2.8$
$\mathbb{E}(l_3) = 0.1 * 3 + 0.1 * 3 + 0.2 * 2 + 0.4 * 3 + 0.2 * 3 = 2.8$
Le 3 probabilità sono dunque equivalenti. Tuttavia, per nessuna delle 3 probabilità il codice è ottimale.

2. Data una sorgente $\mathbb{X} = \{a_1, \ldots, a_8, a_9, \ldots, a_{12}\}$ e $\mathbb{P}(a_i) = \begin{cases} 0.1 & i = 1, \ldots, 8 \\ 0.05 & i = 9, \ldots, 12 \end{cases}$
   Dato $d = 5$, trovare un codice istantaneo e, se possibile, ottimale.
   Usando l'algoritmo di Huffman, notiamo che il codice restituito non è ottimale, contrariamente a quanto abbiamo studiato nella lezione precedente. Infatti, il codice restituito è il seguente:

> $c(a_1) = 1$
> $c(a_2) = 2$
> $c(a_3) = 31$
> $c(a_4) = 32$

> $c(a_5) = 33$
> $c(a_6) = 34$
> $c(a_7) = 35$
> $c(a_8) = 41$
> $c(a_9) = 42$
> $c(a_{10}) = 43$
> $c(a_{11}) = 44$
> $c(a_{12}) = 45$
> Non va bene, perché il sottoalbero del simbolo 5 non viene coperto.

Si può risolvere introducendo nell'algoritmo di Huffman un numero di simboli "Dummy" con probabilità 0, per far sì che l'algoritmo possa restituire un codice istantaneo ottimale. Il numero di simboli dummy è dato dalla seguente espressione

$$n + q \equiv 1 \mod d - 1$$

dove $n = |\mathbb{X}|$ e q è il numero di simboli dummy. Con questi accorgimenti, il codice restituito è:

> $c(a_1) = 1$
> $c(a_2) = 2$
> $c(a_3) = 3$
> $c(a_4) = 41$
> $c(a_5) = 42$
> $c(a_6) = 43$
> $c(a_7) = 44$
> $c(a_8) = 45$
> $c(a_9) = 51$
> $c(a_{10}) = 52$
> $c(a_{11}) = 53$
> $c(a_{12}) = 54$