Teoria dell'Informazione

Simone Alessandro Casciaro

18 Ottobre 2024

Lezione 6: Codici di Huffman

> Esercizio
>
> In questa lezione, partiamo da un esercizio
> $\mathbb{X} = \{s_1, s_2, s_3, s_4, s_5, s_6\}$
> $\mathbb{P} = \{0.05, 0.45, 0.12, 0.09, 0.16, 0.13\}$
> Costruiamo un codice istantaneo.
> È sufficiente costruire un codice a virgola, ad esempio:
> $c_1(s_1) = 1$
> $c_1(s_2) = 01$
> $c_1(s_3) = 001$
> $c_1(s_4) = 0001$
> $c_1(s_5) = 00001$
> $c_1(s_6) = 00000$
> Questo codice non è ideale, in quanto lo abbiamo costruito ignorando le probabilità con cui ogni simbolo appare nella sorgente.
> Infatti, $\mathbb{E}(l_{c_1}) = 1 * 0.5 + 2 * 0.45 + 3 * 0.12 + 4 * 0.09 + 5 * 0.16 + 5 * 0.13 = \boxed{3.12}$
>
> Se costruiamo un codice a virgola ordinando i simboli per probabilità, abbiamo:
> $c_2(s_1) = 00000$
> $c_2(s_2) = 1$
> $c_2(s_3) = 0001$
> $c_2(s_4) = 00001$
> $c_2(s_5) = 01$
> $c_2(s_6) = 001$
> E abbiamo $\mathbb{E}(l_{c_2}) = 5 * 0.5 + 1 * 0.45 + 4 * 0.12 + 5 * 0.09 + 2 * 0.16 + 3 *$

> $0.13 = \boxed{2.34}$

Algoritmo di Huffman

L'algoritmo di Huffman ci permette di trovare il codice istantaneo che minimizza il valore atteso delle lunghezze $\mathbb{E}(l_c)$

Cioè vogliamo trovare il codice tale che $\begin{cases} \min\limits_{l_1,...,l_n} \sum\limits_{i=1}^{m} p_i l_i \\ \sum\limits_{i=1}^{m} d^{-l_1} \leq 1 \end{cases}$

Come funziona l'algoritmo?

1. Si prendono i simboli della sorgente $\mathbb{X}$ e si ordinano in base alle probabilità in maniera decrescente.
2. Si crea un nuovo modello fittizio della sorgente $\mathbb{X}'$ in cui i d simboli meno probabili sono sostituiti da un unico simbolo $\hat{x}$ con probabilità pari alla somma delle probabilità dei simboli sostituiti: $p(\hat{x}) = \sum\limits_{i=m-d+1}^{m} p(x_i)$
3. Si itera fino a che la sorgente ha al massimo d simboli.

> Esempio con il codice di inizio lezione
> Abbiamo $\mathbb{X}_1$ e supponiamo di avere $d = 2$:
> $p(s_2) = 0.45$
> $p(s_5) = 0.16$
> $p(s_6) = 0.13$
> $p(s_3) = 0.12$
> $p(s_4) = 0.09$
> $p(s_1) = 0.05$
> I due simboli meno probabili sono s_4 e s_1
>
> Costruiamo $\mathbb{X}_2$ sostituendo questi due simboli, poi ordiniamo nuovamente
> $p(s_2) = 0.45$
> $p(s_5) = 0.16$
> $p(s_{4+1}) = 0.14$
> $p(s_6) = 0.13$

> $p(s_3) = 0.12$
> e iterativamente finché abbiamo al massimo 2 simboli
>
> Costruiamo $\mathbb{X}_3$
> $p(s_2) = 0.45$
> $p(s_{6+3}) = 0.25$
> $p(s_5) = 0.16$
> $p(s_{4+1}) = 0.14$
>
> Costruiamo $\mathbb{X}_4$
> $p(s_2) = 0.45$
> $p(s_{5+4+1}) = 0.30$
> $p(s_{6+3}) = 0.25$
>
> Costruiamo $\mathbb{X}_5$
> $p(s_{5+4+1+6+3}) = 0.55$
> $p(s_2) = 0.45$

4. Si assegnano le d codifiche ai d simboli rimasti nell'ultima sorgente creata e si prosegue a ritroso iterativamente usando le concatenazioni

> Su $\mathbb{X}_5$
> $c(s_{5+4+1+6+3}) = 0$
> $c(s_2) = 1$
> ma $s_{5+4+1+6+3}$ era composto da s_{5+4+1} e s_{6+3}, quindi associo loro le due codifiche con una concatenazione
>
> Su $\mathbb{X}_4$
> $c(s_{5+4+1}) = 00$
> $c(s_{6+3}) = 01$
> $c(s_2) = 1$
> e si prosegue fino a $\mathbb{X}_1$
>
> Su $\mathbb{X}_3$
> $c(s_5) = 000$
> $c(s_{4+1}) = 001$

$c(s_{6+3}) = 01$
$c(s_2) = 1$

Su $\mathbb{X}_2$
$c(s_5) = 000$
$c(s_{4+1}) = 001$
$c(s_6) = 010$
$c(s_3) = 011$
$c(s_2) = 1$

Su $\mathbb{X}_1$
$c(s_5) = 000$
$c(s_4) = 0010$
$c(s_1) = 0011$
$c(s_6) = 010$
$c(s_3) = 011$
$c(s_2) = 1$
Calcolando $\mathbb{E}(l_c) = 4 * 0.05 + 1 * 0.45 + 3 * 0.12 + 4 * 0.09 + 3 * 0.16 + 3 * 0.13 = \boxed{2.24}$

Un'implementazione Python del Codice di Huffman è la seguente

```python
def Huffman(X: list[str], P: list[float], d: int = 2) -> list:
    # Algoritmo che trova il codice istantaneo che minimizza il valore atteso delle lunghezze di c
    if len(X) != len(P):
        raise Exception('X and P must have the same lenght')

    # Inseriamo i Dummies
    nDummies = (1-len(X))%(d-1)

    sorgente = [(x, p) for x, p in zip(X, P)]
    for _ in range(nDummies):
        sorgente.append(('Dummy', 0))

    # Iteriamo finché la sorgente ha al massimo d simboli
    while len(sorgente) > d:
        # Si ordinano le probabilità
        sorgente = sorted(sorgente, key = lambda x:x[1], reverse=True)
        new_sorgente = []

        for i in range(len(sorgente)-d):
            new_sorgente.append(sorgente[i])
        # Si tolgono i d simboli meno probabili e si aggiunge un unico simbolo che li sostituisce
        sum_p = 0
        sum_x = []
        for i in range(d):
            sum_p += sorgente[len(sorgente)-1-i][1]
            sum_x.append(sorgente[len(sorgente)-1-i][0])
        new_sorgente.append((sum_x, sum_p))
        sorgente = new_sorgente.copy()

    # Creiamo la codifica
    codifica = []
    for i, simbolo in enumerate(sorgente):
        codifica.append((simbolo[0], str(i)))

    # Risrotoloiamo le codifiche
    while len(codifica) < len(X) + nDummies:
        for i, simbolo in enumerate(codifica):
            if type(simbolo[0]) == list:
                for j, minisimbolo in enumerate(simbolo[0]):
```

```
                    codifica.append((minisimbolo, simbolo[1]+str(j)))
                codifica.remove(simbolo)

    # Togliamo i Dummies
    for simbolo, codice in codifica:
        if simbolo == 'Dummy':
            codifica.remove((simbolo, codice))
    return codifica
```

Codici di Huffman

Sia c' un codice di Huffman $d-$ario per la sorgente $< \mathbb{X}', \mathbb{P}' >$ con $\mathbb{X}' = \{x_1, \dots, x_{m-d+1}\}$ e $\mathbb{P}' = \{p_1, \dots, p_{m-d+1}\}$ tali che
$p(x_i) = p_i \quad \forall i = 1, \dots, m-d+1$ e
$p_i \geq p_j \quad \forall i < j$ (cioè le probabilità sono ordinate in ordine decrescente)
Si costruisce la sorgente $\mathbb{X} = \{x_1, \dots, x_{m+1}\}$ togliendo un simbolo x_k ma aggiungendo d elementi, in modo tale che $|\mathbb{X}| = m$. Si devono rispettare due proprietà:

1. $0 \leq p(x_{m+1}) \leq \dots \leq p(x_{m-d+2}) < p(x_{m-d+1})$ Le probabilità devono rimanere ordinate
2. $\sum_{i=2}^{d+1} p(x_{m-d+i}) = p(x_k)$ Le probabilità dei simboli aggiunti devono essere, sommate, uguali a quella del simbolo tolto

Allora, costruendo il codice c in questo modo:

$$c(x_i) = \begin{cases} c'(x_i) & i \leq m-d+1 \\ c'(x_k)a & i > m-d+1 \quad a = 0, \dots, d-1 \end{cases}$$

$c(x)$ è un codice di Huffman per la sorgente $\mathbb{X}$

Teorema di Huffman

Ipotesi

Sia una sorgente $< \mathbb{X}, \mathbb{P} >$, $d > 1$ e sia c codice di Huffman

Tesi

$$c_2 \text{ codice istantaneo } \implies \mathbb{E}(l_c) \leq \mathbb{E}(l_{c_2})$$

In pratica, il codice di Huffman minimizza il valore atteso delle lunghezze rispetto a tutti gli altri codici istantanei per la sorgente $\mathbb{X}$

Dimostrazione

Nella nostra dimostrazione, assumeremo sempre $d = 2$ per semplicità nei conti, ma essa è estendibile anche al caso $d > 2$
La dimostrazione è data per induzione, dunque

$\boxed{\text{Caso Base}}$
$|\mathbb{X}| \leq d$ (nel costro caso, quindi, $|\mathbb{X}| = 2$)
Abbiamo i simboli s_1 e s_2 associate alle loro probabilità p_1 e p_2.
Indipendentemente dalle probabilità, si può associare ad ogni simbolo una codifica formata da un solo elemento di D senza rendere il codice ambiguo.
Ad esempio, $c(s_1) = 0$ e $c(s_2) = 1$ oppure viceversa.

$\boxed{\text{Caso Induttivo}}$
$|\mathbb{X}| = m$
Assumiamo che per $|\mathbb{X}| = m - 1$ il teorema di Huffman valga.
Prendiamo due elementi $u, v \in \mathbb{X}$ tali che

$$p(u) \text{ e } p(v) \text{ siano minime} \tag{1}$$

Definiamo una nuova sorgente $< \mathbb{X}', \mathbb{P}' >$ sostituendo i simboli $u, v \in \mathbb{X}$ con un simbolo $z \in \mathbb{X}'$ e con probabilità
$p'(x) = \begin{cases} p(x) & x \neq z \\ p(u) + p(v) & x = z \end{cases}$

$$\text{Il codice } c' \grave{}\text{e un codice di Huffman ottimale per la sorgente } \mathbb{X}' \tag{2}$$

per ipotesi induttiva.

Costruiamo il codice c sulla base di c' in questo modo

$$c(x) = \begin{cases} c'(x) & x \notin \{u, v\} \\ c'(z)0 & x = u \\ c'(z)1 & x = v \end{cases}$$

$$c \text{ è un codice di Huffman} \tag{3}$$

per definizione di codice di Huffman.

Calcoliamo $\mathbb{E}(l_c)$

$$\begin{aligned} \mathbb{E}(l_c) &= \sum_{x \in X} l_c(x)p(x) \\ &= \sum_{x \in X'} l_{c'}(x)p'(x) - l_{c'}(z)p'(z) + l_c(u)p(u) + l_c(v)p(v) \\ &= \sum_{x \in X'} l_{c'}(x)p'(x) - l_{c'}(z)p'(z) + \big(l_{c'}(z) + 1\big)p(u) + \big(l_{c'}(z) + 1\big)p(v) \\ &= \sum_{x \in X'} l_{c'}(x)p'(x) - l_{c'}(z)p'(z) + \big(l_{c'}(z) + 1\big)\big(p(u) + p(v)\big) \\ &= \sum_{x \in X'} l_{c'}(x)p'(x) - l_{c'}(z)p'(z) + \big(l_{c'}(z) + 1\big)\big(p'(z)\big) \\ &= \sum_{x \in X'} l_{c'}(x)p'(x) - l_{c'}(z)p'(z) + l_{c'}(z)p'(z) + p'(z) \\ &= \sum_{x \in X'} l_{c'}(x)p'(x) + p'(z) \\ &= \mathbb{E}(l_{c'}) + p'(z) \end{aligned}$$

Dunque

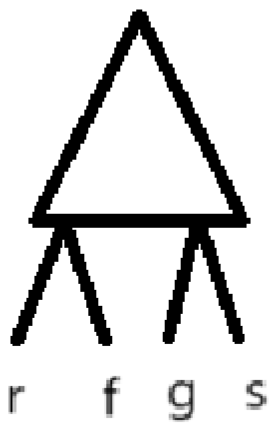
$$\mathbb{E}(l_c) = \mathbb{E}(l_{c'}) + p'(z) \tag{4}$$

Consideriamo ora, una seconda funzione di codifica c_2 per la sorgente $\mathbb{X}$. Prendiamo due elementi $r, s \in \mathbb{X}$ tali che

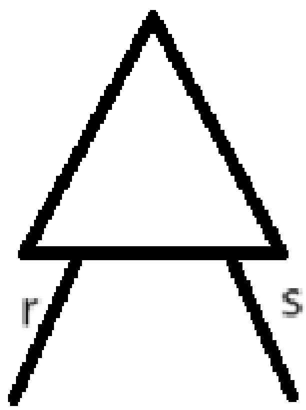
$$l_{c_2}(r) \text{ e } l_{c_2}(s) \text{ sono massime} \tag{5}$$

Si hanno 3 casi nell'albero di codifica:

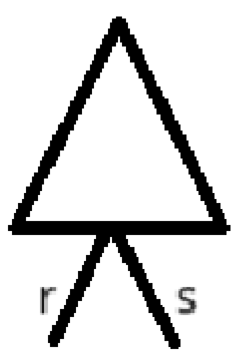
1. r e s non sono fratelli, ma hanno dei fratelli



2. r e s non sono fratelli e non hanno fratelli



3. r e s sono fratelli



Tutti e 3 i casi sono riconducibili al terzo, quindi possiamo analizzare solo questo caso senza perdere di generalità.
Costruisco il codice $\tilde{c}_2$ in questo modo

$$\tilde{c}_2(x) = \begin{cases} c_2(x) & x \notin \{r, s, u, v\} \\ c_2(u) & x = r \\ c_2(v) & x = s \\ c_2(r) & x = u \\ c_2(s) & x = v \end{cases}$$ In poche parole, stiamo invertendo le codifiche tra u, v, r, s

Calcoliamo $\mathbb{E}(l_{\tilde{c}_2}) - \mathbb{E}(l_{c_2})$

$$\begin{aligned}
\mathbb{E}(l_{\tilde{c}_2}) - \mathbb{E}(l_{c_2}) &= \sum_{x \in \mathbb{X}} p(x) l_{\tilde{c}_2}(x) - \sum_{x \in \mathbb{X}} p(x) l_{c_2}(x) \\
&= \sum_{x \in \mathbb{X}} p(x) \Big(l_{\tilde{c}_2}(x) - l_{c_2}(x) \Big) \\
&= \sum_{x \in \{u,v,r,s\}} p(x) \Big(l_{\tilde{c}_2}(x) - l_{c_2}(x) \Big) \\
&= p(r) l_{c_2}(u) + p(u) l_{c_2}(r) + p(s) l_{c_2}(v) + p(v) l_{c_2}(s) - p(u) l_{c_2}(u) - p(r) l_{c_2}(r) - \\
&= \Big(p(r) - p(u) \Big) \Big(l_{c_2}(u) - l_{c_2}(r) \Big) + \Big(p(s) - p(v) \Big) \Big(l_{c_2}(v) - l_{c_2}(s) \Big)
\end{aligned}$$

Grazie al punto 1, sappiamo che $p(r) \geq p(u)$ e $p(s) \geq p(v)$

Grazie al punto 5, sappiamo che $l_{c_2}(u) \leq l_{c_2}(r)$ e $l_{c_2}(v) \leq l_{c_2}(s)$

Dunque il risultato è negativo e

$$\mathbb{E}(l_{\tilde{c}_2}) \leq \mathbb{E}(l_{c_2}) \tag{6}$$

Costruiamo un'ulteriore funzione di codifica c_2' per la sorgente $< \mathbb{X}', \mathbb{P}' >$ in questo modo:

$$c_2'(x) = \begin{cases} \tilde{c}_2(x) & x \neq z \\ \omega & x = z \end{cases}$$

con $p'(z) = p(u) + p(v)$

Calcoliamo $\mathbb{E}(l_{\tilde{c}_2})$

$$
\begin{aligned}
\mathbb{E}(l_{\tilde{c}_2}) &= \sum_{x \in \mathbb{X}} p(x) l_{\tilde{c}_2}(x) \\
&= \sum_{x \in \mathbb{X} \setminus \{z\}} p'(x) l_{c_2'}(x) + p(u)\Big(l_{c_2'}(z) + 1\Big) + p(v)\Big(l_{c_2'}(z) + 1\Big) \\
&= \sum_{x \in \mathbb{X} \setminus \{z\}} p'(x) l_{c_2'}(x) + \Big(l_{c_2'}(z) + 1\Big)\big(p(u) + p(v)\big) \\
&= \sum_{x \in \mathbb{X} \setminus \{z\}} p'(x) l_{c_2'}(x) + \Big(l_{c_2'}(z) + 1\Big)\big(p'(z)\big) \\
&= \sum_{x \in \mathbb{X} \setminus \{z\}} p'(x) l_{c_2'}(x) + l_{c_2'}(z) p'(z) + p'(z) \\
&= \mathbb{E}(l_{c_2'}) + p'(z)
\end{aligned}
$$

Quindi

$$
\mathbb{E}(l_{\tilde{c}_2}) = \mathbb{E}(l_{c_2'}) + p'(z) \tag{7}
$$

Unendo i pezzi:

Grazie al punto 3, sappiamo che c è un codice di Huffman

Inoltre,

$$
\begin{aligned}
\mathbb{E}(l_c) &= \mathbb{E}(l_{c'}) + p'(z) && \text{Per il punto 4} \\
&\leq \mathbb{E}(l_{c_2'}) + p'(z) && \text{Per il punto 2} \\
&= \mathbb{E}(l_{\tilde{c}_2}) && \text{Per il punto 7} \\
&\leq \mathbb{E}(l_{c_2}) && \text{Per il punto 6}
\end{aligned}
$$

Abbiamo dimostrato che $\mathbb{E}(l_c) \leq \mathbb{E}(l_{c_2})$ per qualunque c_2 istantaneo e quindi il codice c è ottimo.