Claudia Landi
Matricola: 7149573
Anno Accademico: 2023-2024

Sicurezza dei Dati e Privacy - Relazione terzo set di esercizi

1 Esercizi di programmazione

1.1 Esercizio 3.2

La soluzione dell'esercizio è stata implementata nel file *huffman.py*. Il metodo che si occupa di eseguire i passi dell'algoritmo di Huffman è *huffman_algorithm* che prende in input due liste: la prima rappresenta l'alfabeto, la seconda le probabilità di ciascuna lettera. Come prima cosa viene fatto un controllo sulle lunghezze delle due liste, per verificare che corrispondano, dopodiché viene creato un dizionario che associa a ciascuna lettera la sua probabilità, così da poter gestire meglio i passaggi successivi. Il primo e il secondo passo dell'algoritmo di Huffman sono stati implementati rispettivamente nelle funzioni *huffman_first_step* e *huffman_second_step*. La prima prende in input il dizionario delle probabilità delle lettere e in un ciclo seleziona i due elementi con minore probabilità, li raggruppa in una tupla e vi associa nel dizionario la somma delle due rispetive probabilità, inoltre gli elementi raggruppati vengono eliminati dal dizionario, così da non essere considerati nuovamente nelle iterazioni successive del ciclo. Queste operazioni sono ripetute finché nel dizionario non rimane un solo elemento, che avrà come chiave la tupla contenente tutte le lettere dell'alfabeto raggruppate gerarchicamente in base alle probabilità, e come valore la probabilità totale, ovvero 1. A questo punto la tupla viene estratta dal dizionario e passata alla funzione *huffman_second_step*, che si occupa di creare una codeword per ciascun simbolo dell'alfabeto. Per farlo associa 0 all'elemento di sinistra della tupla e 1 all'elemento di destra, dopodiché, se gli elementi erano a

loro volta delle tuple, ripete ricorsivamente i passaggi sulle componenti delle sotto-tuple, concatenando ogni volta 0 o 1 alla codeword già associata all'elemento. Viene infine restituito un dizionario che associa a ciascuna lettera una codeword, il codice così ottenuto è ottimo.
La procedura per la decodifica di una stringa binaria è stata implementata nel metodo *huffman_ decoding*, che prende in input una stringa binaria e un dizionario che associa a ciascuna lettera dell'alfabeto una codeword. La funzione scorre la stringa in ingresso finché non trova una sottostringa che corrisponde ad una codeword del dizionario e a quel punto la sostituisce con la lettera dell'alfabeto corrispondente. Infine, terminata la decodifica, la stringa tradotta viene restituita dal metodo.

2 Esercizi di approfondimento

2.1 Esercizio 2.2

a) Un esempio di strategia nel caso di 5 monete è rappresentato dall'albero in figura 1.

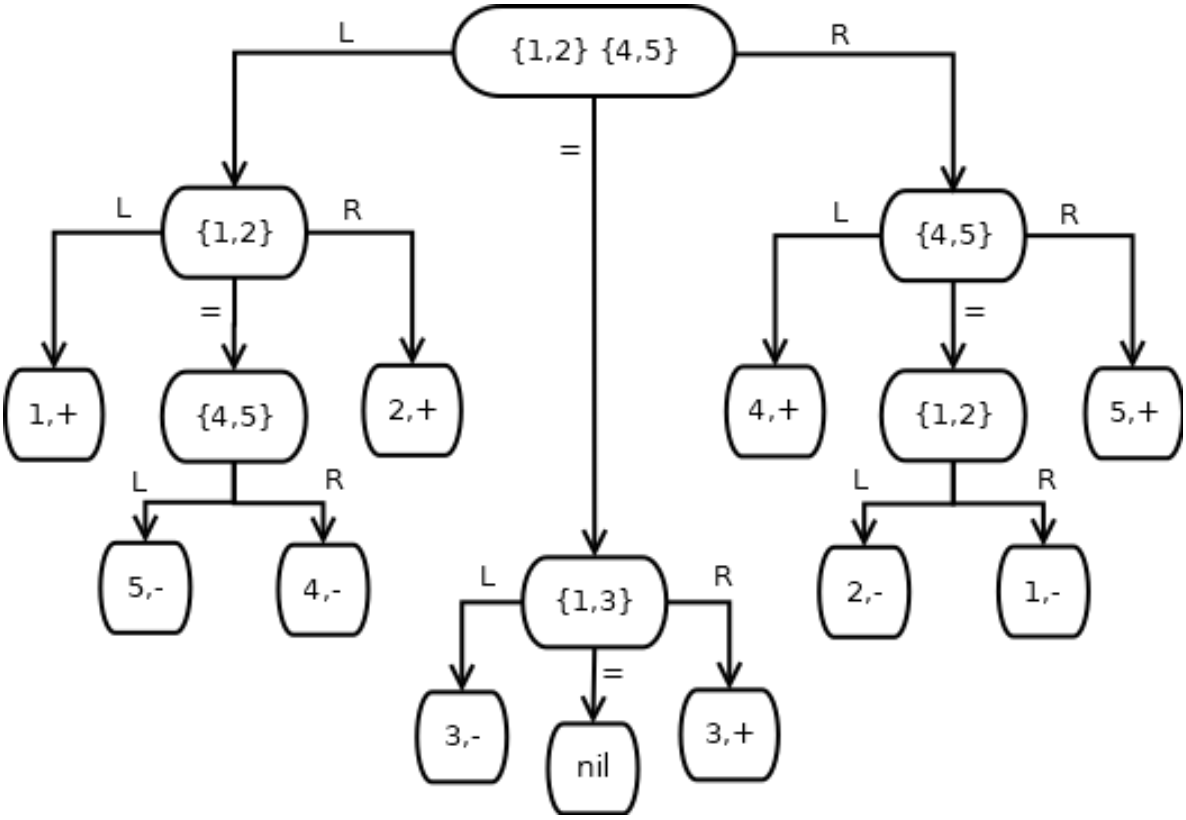

Figura 1: Rappresentazione ad albero di una strategia con 5 monete.

La strategia consiste nel pesare inizialmente due coppie di monete, lasciando la quinta fuori. Se il peso si sbilancia in una direzione, si continuano ad effettuare le pesate con le due monete che si trovano dal lato più pesante. Se una delle due risulta a sua volta più pesante, allora è stata individuata la moneta falsa, quella appunto che pesa di più, se invece il peso delle due monete è lo stesso, vuol dire che quella falsa si trova nella coppia che nella prima misurazione pesava meno. In quest'ultimo caso si confrontano le due monete che insieme pesavano meno, la più leggera sarà quella falsa. Se invece nella prima pesata le due coppie di monete hanno lo stesso peso, vuol dire che la moneta falsificata è quella rimasta fuori, oppure che non esiste (nil). Per verificarlo si confronta la moneta rimanente con una di quelle già controllate, se pesa di più o di meno, vuol dire che la rimanente era quella falsa, se invece pesano uguali, nessuna moneta è stata falsificata.

b) La lunghezza massima della strategia descritta corrisponde all'altezza dell'albero che la rappresenta, pertanto è uguale a 3. La lunghezza media corrisponde invece all'altezza media dell'albero, che deve essere determinata considerando la probabilità delle foglie:

$$l_{media} = \sum_{x \in X} p(x) l(x)$$

dove $p(x)$ è la probabilità di ciascuna foglia, ovvero $\frac{1}{11}$ e $l(x)$ è la lunghezza del cammino che porta dalla radice alla foglia x, pertanto si ha:

$$l_{media} = \frac{3}{11} \cdot 4 + \frac{2}{11} \cdot 7 = \frac{26}{11}$$

c) In una strategia qualsiasi, dato n numero di monete, il limite inferiore al numero massimo di pesate è dato dal limite inferiore dell'altezza dell'albero ternario, indicata con k. In un albero ternario di altezza k, si hanno al più 3^k foglie. Considerando i $2n + 1$ possibili valori di X, si ha che $3^k \geq 2n + 1$ e quindi:

$$k \geq \lceil \log_3(2n + 1) \rceil$$

corrisponde al limite inferiore al numero massimo di pesate.

d) Si osserva che ogni strategia corrisponde ad un codice istantaneo ternario sull'alfabeto di tre simboli {L, R, =}, pertanto è possibile generalizzare

i risultati visti per i codici binari al caso ternario, considerando la base 3 invece che la base 2. La lunghezza media della strategia coincide con la lunghezza media del codice $L(C)$, per il quale si ha:

$$L(C) \geq H_3(\mathbf{p})$$

dove $\mathbf{p}$ rappresenta il vettore delle probabilità di ciascuna $x \in X$ e $H_3(\cdot)$ rappresenta l'entropia calcolata con i logaritmi in base 3 invece che in base 2 come nel caso dei codici binari. Si osserva che $\mathbf{p}$ è un vettore di probabilità uniformi $\left(\frac{1}{2n+1}, \frac{1}{2n+1}, \ldots, \frac{1}{2n+1}\right)$ quindi sostituendo si ottiene:

$$H_3(\mathbf{p}) = H_3\left(\frac{1}{2n+1}, \frac{1}{2n+1}, \ldots, \frac{1}{2n+1}\right) =$$

$$= \sum_{i=1}^{2n+1} \frac{1}{2n+1} \log_3(2n+1) = \log_3(2n+1)$$

il limite inferiore per il numero medio di pesate è dato quindi da $L(C) \geq \log_3(2n+1)$.

2.2 Esercizio 2.3

Si considera un canale di comunicazione in cui ogni 5 minuti viene liberato un piccione, che trasporta una lettera (simbolo) di 8 bit. Per il canale si hanno quindi un alfabeto di ingresso e uno di uscita di $2^8 = 256$ elementi.

a) Nel caso in cui tutti i piccioni raggiungano la destinazione, il canale è debolmente simmetrico, infatti la sua matrice corrisponde alla matrice identità e vale dunque che:

1. tutte le righe sono uguali a meno di permutazioni,
2. la somma di tutti gli elementi sulla colonna è $c = 1$.

Se il canale è debolmente asimmetrico si ha che la sua capacità C è data da:

$$C = \log_2 |Y| - H(r)$$

dove Y è l'alfabeto di uscita del canale e $H(r)$ è l'entropia di una riga qualsiasi della matrice del canale, che in questo caso è 0, dato che il messaggio consegnato è sicuramente quello corretto. Pertanto si avrà

che la capacità del canale è:

$$C = \log_2 |Y| = \log_2(256) = 8$$

Per determinare infine la capacità bit/ora del canale, si considera che, partendo un piccione ogni 5 minuti, in un'ora arriveranno a destinazione 12 piccioni, si avrà quindi una capacita di $12 \cdot 8 = 96$ bit/ora.

b) Considerando invece il caso in cui una frazione α dei piccioni venga abbattuta, sostituendo ogni piccione abbattuto con uno che trasporta una lettera (simbolo di 8 bit) scelto a caso, si avrà una probabilità di $1 - \alpha$ che il piccione raggiunga la destinazione.
Si nota che anche se un piccione viene abbattuto (con probabilità α) c'è comunque una probabilità di $\frac{1}{2^8}$ che il piccione con cui viene sostituito contenga la stessa lettera originale, quindi la probabilità totale che la lettera corretta arrivi a destinazione è $(1 - \alpha) + \frac{\alpha}{2^8}$, la probabilità che invece venga consegnata una lettera sbagliata in particolare (una delle altre 255) è di $\frac{\alpha}{2^8}$. La matrice del canale è quindi una matrice 256×256 della forma seguente:

$$M = \begin{bmatrix} (1-\alpha) + \frac{\alpha}{2^8} & \frac{\alpha}{2^8} & \frac{\alpha}{2^8} & \dots & \frac{\alpha}{2^8} \\ \frac{\alpha}{2^8} & (1-\alpha) + \frac{\alpha}{2^8} & \frac{\alpha}{2^8} & \dots & \frac{\alpha}{2^8} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\alpha}{2^8} & \frac{\alpha}{2^8} & \frac{\alpha}{2^8} & \dots & (1-\alpha) + \frac{\alpha}{2^8} \end{bmatrix}$$

I valori sulla diagonale corrispondono al caso in cui il messaggio è corretto, gli altri al caso in cui il messaggio in uscita è diverso da quello che si aveva in ingresso. Anche questa matrice è debolmente simmetrica, pertanto si può nuovamente utilizzare la formula della capacità:

$$C = \log_2 |Y| - H(r)$$

Per determinare l'entropia, si utilizza la seguente proprietà di una generica distribuzione $\mathbf{p} = (p_1, \dots, p_n)$:

$$H(p_1, \dots, p_n) = H(p_1) + (1 - p_1) H\left(\frac{p_2}{1-p_1}, \dots, \frac{p_n}{1-p_1}\right)$$

Si definisce inoltre X come variabile aleatoria distribuita come $\mathbf{p}$, e Y

variabile aleatoria ausiliaria tale che:

$$Y = \begin{cases} 1 & \text{se } X = x_1 \\ 0 & \text{altrimenti} \end{cases}$$

Per dimostrare tale proprietà si può utilizzare la chain rule per scrivere l'entropia della distribuzione congiunta di X e Y come:

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$$

Si osserva che per la definizione di Y, conoscere il valore di X rende nulla l'incertezza su Y, pertanto $H(Y|X) = 0$ e quindi si ha:

$$H(X) = H(Y) + H(X|Y)$$

Per definizione:

$$H(X|Y) = \sum_y p(y)H(X|y) = p_1 \cdot H(X|Y=1) + (1-p_1) \cdot H(X|Y=0)$$

dove p_1 è la probabilità che $Y = 1$ ovvero di $X = x_1$. Dato che se $Y = 1$ si sa che $X = x_1$, l'incertezza su X dato $Y = 1$ è nulla, ovvero $H(X|Y=1) = 0$, quindi:

$$H(X|Y) = (1 - p_1) \cdot H(X|Y=0)$$

e sostituendo nella formula dell'entropia di X:

$$H(X) = H(Y) + (1 - p_1) \cdot H(X|Y=0)$$

Dalla definizione di Y si ha inoltre che $H(Y)$ corrisponde all'entropia binaria di p_1, $B(\lambda) = H(\lambda, 1-\lambda)$, mentre considerando che X è distribuita come $\mathbf{p}$, si ha $H(X|Y=0) = H\left(\frac{p_2}{1-p_1}, \ldots, \frac{p_n}{1-p_1}\right)$, concludendo la dimostrazione.
Utilizzando quanto dimostrato nel caso del canale di comunicazione preso in analisi, è possibile calcolare l'entropia della prima riga della matrice M:

$$
\begin{aligned}
H(r_1) &= B\left((1-\alpha)+\frac{\alpha}{2^8}\right) \quad + \left(\frac{\alpha\cdot 255}{2^8}\right)\cdot H\left(\frac{1}{255},\ldots,\frac{1}{255}\right) \\
&= B\left(1-\frac{\alpha}{255}\right)+\left(\alpha\frac{255}{256}\right)\log_2(255) \\
&\approx B\left(1-\alpha\frac{255}{256}\right)+\alpha\cdot 7.9631
\end{aligned}
$$

Considerando il numero di piccioni arrivati a destinazione in un'ora e utilizzando la formula della capacità per un canale debolmente simmetrico, si ottiene:

$$
bit/ora = 12\cdot C = 12\cdot\left(8-B\left(1-\alpha\frac{255}{256}\right)+\alpha\cdot 7.9631\right)
$$

dove C rappresenta la capacità del canale, data da $C=\log_2|2^8|-H(r_1)$.