Teoria dell'Informazione/RTI **Anno Accademico 2022–2023**

Lezione 7

Ugo Vaccaro

Nella lezione scorsa abbiamo visto il seguente risultato

Lemma 1 *Sia $\mathcal{A}$ un arbitrario algoritmo per la ricerca in $S = \{1, \ldots, n\}$, che procede attraverso la formulazione di domande del tipo "è $x \leq j$?", per opportuni $j \in S$, e sia $\mathsf{c} : S \to \{0,1\}^*$ la codifica ottenuta a partire dall'algoritmo $\mathcal{A}$. Allora*

1. *c è una codifica alfabetica degli elementi in S.*
2. *c è anche una codifica prefisso degli elementi in S*

Facciamo vedere che vale anche il vicersa, ovvero che **ogni** codifica $\mathsf{c} : \{1, \ldots, n\} \to \{0,1\}^*$ prefisso ed alfabetica degli elementi in $\{1, \ldots, n\}$ permette di costruire un algoritmo per la ricerca in $\{1, \ldots, n\}$, che procede attraverso la formulazione di domande del tipo "è $x \leq j$?".

Lemma 2 *Sia $f : \{1, \ldots, n\} \to \{0,1\}^*$ una codifica prefisso ed alfabetica. Allora f definisce un algoritmo $\mathcal{A}$ per la ricerca in $S = \{1, \ldots, n\}$, che procede attraverso la formulazione di domande del tipo "è $x \leq j$?", per opportuni $j \in S$.*

Dimostrazione. Sia

$$A_0^1 = \{i \in S : \text{il primo bit di } f(i) \text{ è uguale a } 0\},$$

$$A_1^1 = \{i \in S : \text{il primo bit di } f(i) \text{ è uguale a } 1\}.$$

In virtù del fatto che la codifica f è alfabetica, vale che $\forall a \in A_0^1$ e $\forall b \in A_1^1$ si ha $a < b$. Denotiamo con m_1 il valore massimo dell'insieme A_0^1. L'algoritmo $\mathcal{A}$ formulerà come prima domanda la domanda "è $x \leq m_1$?". Se la risposta è SI, allora sappiamo che il valore incognito x si trova nell'insieme A_0^1 (dove $\mathcal{A}$ continuerà la ricerca), se la risposta è NO, allora sappiamo che il valore incognito x si trova nell'insieme A_1^1 (dove $\mathcal{A}$ continuerà la ricerca). Supponiamo che la risposta sia stata SI. Allora, detti

$$A_0^2 = \{i \in A_0^1 : \text{il secondo bit di } f(i) \text{ è uguale a } 0\},$$

$$A_1^2 = \{i \in A_0^1 : \text{il secondo bit di } f(i) \text{ è uguale a } 1\}$$

e m_2 il valore massimo dell'insieme A_0^2, l'algoritmo $\mathcal{A}$ formulerà come seconda domanda la domanda "è $x \leq m_2$?". Se la risposta è SI, allora sappiamo che il valore incognito x si trova nell'insieme A_0^2 (dove $\mathcal{A}$ continuerà la ricerca), ciò sempre a causa della proprietà che f è una codifica alfabetica , se la risposta è NO, allora sappiamo che il valore incognito x si trova nell'insieme A_1^2 (dove $\mathcal{A}$ continuerà la ricerca).

Procedendo in questo modo, l'algoritmo $\mathcal{A}$ ad ogni domanda "scopre" un bit della codifica $f(x)$ del valore incognito $x \in S$. Pertanto, dopo un numero di domande pari alla lunghezza della codifica $f(x)$, l'algoritmo $\mathcal{A}$ conosce tutto $f(x)$ e può quindi risalire al valore di x. □

In virtù dei due risultati appena visti, o parlare di algoritmi che determinano il valore di un'incognita $x \in \{1, \ldots, n\}$ mediante domande del tipo "è $x \leq j$?", per opportuni $j \in \{1, \ldots, n\}$, o parlare di codifiche $f : \{1, \ldots, n\} \to \{0,1\}^*$ prefisso ed alfabetiche, è esattamente la stessa cosa.

Vediamo un esempio. Sia data la codifica prefisso ed alfabetica $f : \{1,2,3,4,5,6,7,8\} \to \{0,1\}^*$ data da $f(1) = 0, f(2) = 10, f(3) = 110, f(4) = 1110, f(5) = 11110, f(6) = 111110, f(7) = 1111110, f(8) = 1111111$. In tal caso $A_0^1 = \{1\}$, $A_1^1 = \{2,3,4,5,6,7,8\}$, e $m_1 = 1$. La prima domanda formulata dall'algoritmo sarà "è $x \leq 1$?". Se la risposta è SI, allora sappiamo che $x = 1$, se la risposta è NO iteriamo la ricerca in A_1^1, procedendo allo stesso modo, e così via.

Ritorniamo al nostro problema originale, che consisteva nello stimare il minimo numero medio di domande del tipo "è $x \leq j$?" per determinare il valore di un elemento incognito x in $\{1,\ldots,n\}$, nota la distribuzione di probabilità $\mathbf{p} = (p_1,\ldots,p_n)$ che descrive la probabilità p_i con cui x assume valore i, per $i = 1,\ldots,n$. Abbiamo realizzato che questo problema è perfettamente equivalente a determinare la minima lunghezza media $\sum_{i=1}^n p_i\ell_i$ di una codifica prefisso alfabetico $f : \{1,\ldots,n\} \to \{0,1\}^*$, dove ℓ_i è la lunghezza della codifica $f(i)$ di $i \in \{1,\ldots,n\}$. Vale il seguente risultato

Teorema 1 *Sia X una variabile casuale che assume valori i, con probabilità p_i, per $i = 1,\ldots,n$, e sia $\boldsymbol{p} = (p_1,\ldots,p_n)$. Per ogni codifica $f : \{1,\ldots,n\} \to \{0,1\}^*$ prefisso ed alfabetica vale che*

$$\sum_{i=1}^n p_i\ell_i \geq H(\mathbf{p}),$$

dove ℓ_i è la lunghezza della codifica $f(i)$ di $i \in \{1,\ldots,n\}$, e $H(\mathbf{p}) = -\sum_{i=1}^n p_i \log p_i$ è l'entropia di $\mathbf{p}$.

Inoltre, esiste una codifica $f : \{1,\ldots,n\} \to \{0,1\}^*$ prefisso ed alfabetica per cui

$$\sum_{i=1}^n p_i\ell_i < H(\mathbf{p}) + 2.$$

Dimostrazione. La diseguaglianza $\sum_{i=1}^n p_i\ell_i \geq H(\mathbf{p})$ è stata già provata nel Teorema 2 della Lezione 3, in quanto ogni codifica prefisso alfabetica è ovviamente anche Unicamente Decifrabile.

Per provare la seconda parte del Teorema, procediamo nel modo seguente. Per ogni $j = 1,\ldots,n$ definiamo i numeri r_j e ℓ_j nel modo seguente

$$r_j = \sum_{i=1}^{j-1} p_i + \frac{p_j}{2} \qquad \ell_j = \lceil -\log p_j \rceil + 1.$$

La parola codice $f(j)$ associata a $j \in \{1,\ldots,n\}$ sarà costituita dai primi ℓ_j bit della espansione binaria di r_j. In altri termini, esprimiamo il numero r_j mediante potenze di $1/2$ nel solito modo

$$r_j = \frac{1}{2}b_1 + \frac{1}{2^2}b_2 + \frac{1}{2^3}b_3 + \ldots,$$

dove $b_i \in \{0,1\}$, per $i \geq 1$, e poniamo $f(j) = b_1 \ldots b_{\ell_j}$. Ad esempio, se $r_j = 0.145$ e $\ell_j = 8$, avremo

$$r_j = \frac{1}{2}0 + \frac{1}{2^2}0 + \frac{1}{2^3} + \frac{1}{2^4}0 + \frac{1}{2^5}0 + \frac{1}{2^6} + \frac{1}{2^7}0 + \frac{1}{2^8} + r = 0.144531 + r,\ 0 < r < \frac{1}{2^8},$$

e quindi $f(j) = 00100101$.

Vale ovviamente che $r_1 < r_2 < \ldots < r_n$. É altresì ovvio che le espansioni binarie di numeri crescenti sono ordinate rispetto all'ordinamento lessicografico, per cui $f(1) \prec f(2) \prec \ldots \prec f(n)$. Quindi la codifica $f : \{1,\ldots,n\} \to \{0,1\}^*$ è alfabetica. Proviamo che f gode della proprietà prefisso. Supponiamo che ciò non sia, ovvero per qualche $i,j \in \{1,\ldots,n\}$, vale che $f(i)$ è prefisso di $f(j)$. Ciò vuol dire che r_i e r_j hanno gli stessi ℓ_i bit all'inizio della

loro espansione binaria. In altri termini, ogni potenza $2^{-1}, 2^{-2}, \ldots, 2^{-\ell_i}$ o compare nella espansione, in termini di potenze negative di 2, di entrambi r_i e r_j, o in nessuna delle due. Di conseguenza

$$|r_i - r_j| < \frac{1}{2^{\ell_i}} \leq \frac{1}{2^{-\log p_i + 1}} = \frac{p_i}{2}. \tag{1}$$

D'altra parte, per definizione di r_i e r_j vale che

$$|r_i - r_j| = \left| \sum_{k=1}^{i-1} p_k + \frac{p_i}{2} - \sum_{k=1}^{j-1} p_k - \frac{p_j}{2} \right| > \frac{p_i}{2} + \frac{p_j}{2} > \frac{p_i}{2},$$

in flagrante contraddizione con (1).

Calcoliamo infine la lunghezza media della codifica f. Abbiamo

$$\begin{aligned}
\sum_{i=1}^{n} p_i \ell_i &= \sum_{i=1}^{n} p_i \lceil -\log p_i \rceil + 1 \\
&< -\sum_{i=1}^{n} p_i (\log p_i + 1) + 1 \\
&= H(\mathbf{p}) + \sum_{i=1}^{n} p_i + \sum_{i=1}^{n} p_i \\
&= H(\mathbf{p}) + 2.
\end{aligned}$$

□

Il Teorema appena visto ci permette di concludere, quindi, che il (minimo) numero medio di domande del tipo "è $x \leq j$?"che sono necessarie (e sufficienti) per determinare i valori di una variabile casuale distribuita in accordo a $\mathbf{p}$, e che assume valori in $\{1, \ldots, n\}$, è compreso tra $H(\mathbf{p})$ e $H(\mathbf{p})+2$. Cosa cambia se possiamo effettuare domande di tipo più generale, ad esempio di tipo "è x in A?", dove A è un sottoinsieme arbitrario di $\{1, \ldots, n\}$? É chiaro che le domande del tipo "è $x \leq j$?"sono un caso particolare di quest'ultimo, in quanto esse corrispondono a chiedere "è x in A?", dove A è costretto ad essere un sottoinsieme della forma $\{1, \ldots, j\}$. Per domande generali di tipo "è x in A?" si può dimostrare un risultato analogo al Teorema 1. In particolare, si può dedurre che il (minimo) numero medio di domande del tipo "è x in A?", dove A può essere un sottoinsieme arbitrario di $\{1, \ldots, n\}$, che sono necessarie (e sufficienti) per determinare i valori di una variabile casuale distribuita in accordo a $\mathbf{p}$, e che assume valori in $\{1, \ldots, n\}$, è compreso tra $H(\mathbf{p})$ e $H(\mathbf{p}) + 1$.

Il risultato prima visto ci offre un'ulteriore importante interpretazione dell'entropia $H(X)$ di una variabile casuale. Esso rappresenta il minimo numero di test del tipo "è $X \leq j$" per poter determinare il valore assunto dalla v.c. X, ovvero è il minimo numero di bits che dobbiamo ricevere per poter determinare il valore incognito assunto da X. Pertanto, $H(X)$ può essere anche interpretata come una misura "dell'incertezza" che abbiamo su quali valori la v.c. X può assumere. Poichè molti problemi algoritmici possono essere riformulati come il problema di determinare il valore assunto da una v.c. X, la questione prima vista ha una rilevanza generale. A mò di esempio, consideriamo il classico problema algoritmico di ordinare una generica sequenza di n numeri $x_1, \ldots, x_n$ dal più grande al più piccolo, per esempio. Il problema può essere visto, equivalentemente, come quello di determinare, tra tutte le possibili $n!$ permutazioni $\pi_1, \ldots \pi_{n!}$ degli n numeri $x_1, \ldots, x_n$, quella permutazione che dispone i numeri dal più piccolo al più grande. Possiamo quindi immaginare di avere una variabile casuale

$$X = \begin{pmatrix} \pi_1 & \pi_2 & \cdots & \pi_{n!} \\ P(\pi_1) & P(\pi_2) & \cdots & P(\pi_{n!}) \end{pmatrix} \tag{2}$$

che assume come valori le possibili $n!$ permutazioni $\pi_1, \ldots \pi_{n!}$ degli n numeri $x_1, \ldots, x_n$, e noi vogliamo scoprire la permutazione che dispone i numeri dal più piccolo al più grande mediante domande del tipo "è $x_i \leq x_j$?".

Ci troviamo ancora una volta di fronte a domande la cui risposta ci dà un solo bit di informazione. Pertanto, il numero medio di domande che dobbiamo effettuare per scoprire la permutazione incognita che dispone i numeri dal più piccolo al più grande è pari al numero medio di domande per scoprire il valore incognito assunto dall variabile casuale (2). Per i risultati visti in queste lezioni, tale numero è pari almeno all'entropia $H(X)$. Se $P(\pi_i) = 1/n!$, per $i = 1, \ldots, n!$, ciò vuol dire che il numero medio di domande che dobbiamo effettuare *non potrà* essere inferiore a $\log n! \geq cn \log n$, per qualche costante c.

In generale, se possiamo riformulare un problema algoritmico come quello di passare da un livello di incertezza $H(X)$ (per qualche variabile casuale X opportunamente definita) ad un livello di incertezza nullo (ovvero siamo in grado di determinare il valore assunto da X) mediante test i cui esiti ci danno al più k bits di informazione ciascheduno, allora otteniamo che il numero test da effettuare per determinare il valore della v.c. X *non può* essere inferiore a $H(X)/k$. Questa osservazione risulta essere utile nel determinare limitazioni inferiori al tempo richiesto da vari algoritmi per risolvere dati problemi.