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

Lezione 19

Ugo Vaccaro

In questa lezione vedremo un'applicazione della Teoria dei Codici a correzione d'errore al seguente problema, già descritto nella Lezione 13. É data una popolazione di n individui $\{1, 2, \ldots, n\} = [n]$, ed al più d di essi sono "positivi". Sia $P \subset [n]$, $|P| \leq d$ tale insieme incognito di elementi positivi. Possiamo tentare di scoprire l'insieme incognito $P \subset [n]$ degli individui positivi modo seguente: scegliamo un insieme $A \subseteq [n]$, ed effettuiamo un test $T(A)$ su di esso che ci restituisce la seguente risposta (esito):

$$T(A) = \begin{cases} 1 & \text{se } A \cap P \neq \emptyset \\ 0 & \text{altrimenti.} \end{cases} \tag{1}$$

L'obiettivo è di identificare l'insieme incognito P effettuando "pochi" test. Come abbiamo già menzionato, il problema in questione astrae vari problematiche pratiche che sorgono in diversi ambiti, dal test di malattie ai test genetici, dal testing industriale alla comunicazione in canali a multiaccesso.

In generale, un algoritmo per la risoluzione del problema deve specificare insiemi $A_1, A_2, \ldots, A_t \subseteq [n]$ da testare, in maniera tale che dalla conoscenza degli esiti $T(A_1), T(A_2), \ldots, T(A_t)$ si possa risalire all'insieme incognito P. Vi sono due possibili classi di algoritmi che potremmo usare: algoritmi adattivi e algoritmi non adattivi. Negli algoritmi adattivi l'insieme A_{i+1} da testare al passo $i+1$ può essere scelto sulla base della conoscenza dei risultati $T(A_1), T(A_2), \ldots, T(A_i)$. Negli algoritmi non adattivi, gli insiemi $A_1, A_2, \ldots, A_t$ vengono scelti tutti allo steso tempo (all'inizio dell'algoritmo) ed i test $T(A_1), T(A_2), \ldots, T(A_t)$ possono essere quindi eseguiti contemporaneamente. Pur essendo gli algoritmi adattivi più vantaggiosi (in quanto possono sfruttare maggiore conoscenza), in pratica si usano quasi solo esclusivamente algoritmi non adattivi. Ciò è dovuto al fatto che la esecuzione dei test $T(A_i)$ può richiedere molto tempo, e nelle situazioni reali non vi può essere la possibilità di aspettare tale tempo per conoscere l'esito $T(A_i)$ prima di scegliere il prossimo test A_{i+1}.

Nella Lezione 13 provammo che *ogni* algoritmo che scopre l'insieme incognito P, deve necessariamente effettuare un numero di test t pari almeno a

$$t \geq \log_2 \sum_{k=0}^{d} \binom{n}{d} \geq d \log_2 \frac{n}{d}. \tag{2}$$

Ovviamente, un numero di test pari a n risolve sempre il problema, basterà testare gli insiemi $A_1 = \{1\}, A_2 = \{2\}, \ldots, A_n = \{n\}$. Cerchiamo di capire se si può fare meglio.

Per ogni insieme $S \subseteq [n]$, associamo ad esso il suo vettore caratteristico $\mathbf{x}(S) = (x_1(S), \ldots, x_n(S)) \in \{0,1\}^n$ così definito

$$\forall i = 1, \ldots, n \qquad x_i(S) = \begin{cases} 1 & \text{se } i \in S \\ 0 & \text{altrimenti.} \end{cases}$$

Dati t generici test $A_1, A_2, \ldots, A_t \subseteq [n]$, consideriamo i loro associati vettori caratteristici $\mathbf{x}(A_1), \mathbf{x}(A_2), \ldots, \mathbf{x}(A_t)$, e costruiamo la matrice binaria M di dimensione $t \times n$ in cui la i-esima riga è proprio il vettore $\mathbf{x}(A_i)$. Ad esempio, se i test fossero $A_1 = \{1\}, A_2 = \{2\}, \ldots, A_n = \{n\}$, ovviamente avremmo che M coinciderebbe con la matrice identità I_n. Vediamo un esempio meno banale con la seguente matrice. Qui $n = 12$ e $t = 6$.

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

Il significato della matrice è ovvio. Essa corrisponderebbe ad un algoritmo che esegue i seguenti test: $A_1 = \{2,4,5,9,10,12\}, A_2 = \{1,4,6,8,10,12\}, A_3 = \{1,3,5,7,8,11\}, A_4 = \{2,4,5,7,9,11\}, A_5 = \{2,3,5,6,9,12\}, A_6 = \{1,3,6,8,10,12\}$. Supponiamo ora che si abbiano i seguenti risultati ai test: $T(A_1) = 1, T(A_2) = 1, T(A_3) = 0, T(A_4) = 1, T(A_5) = 0, T(A_6) = 1$, che possiamo per comodità organizzare nel seguente vettore colonna

$$\begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \tag{3}$$

Supponiamo infine che si sappia che $d \leq 2$ ed cerchiamo di individuare i positivi P. Sicuramente $1 \notin P$ in quanto $1 \in A_3$ ma $T(A_3) = 0$. Analogamente, $2 \notin P$ in quanto $2 \in A_5$ ma $T(A_5) = 0$, $3 \notin P$ in quanto $3 \in A_5$ ma $T(A_5) = 0$. Abbiamo invece che $4 \in P$ in quanto $T(A) = 1$ per tutti gli A per cui $4 \in A$. Analogamente possiamo dedurre che $5 \notin P$ in quanto $5 \in A_5$ ma $T(A_5) = 0$, $6 \notin P$ in quanto $6 \in A_5$ ma $T(A_5) = 0$, $7 \notin P$ in quanto $7 \in A_3$ ma $T(A_3) = 0$, $2 \notin P$ in quanto $8 \in A_3$ ma $T(A_3) = 0$, $9 \notin P$ in quanto $9 \in A_5$ ma $T(A_5) = 0$. Abbiamo invece che $10 \in P$ in quanto $T(A) = 1$ per tutti gli A per cui $10 \in A$. Infine $11 \notin P$ in quanto $11 \in A_3$ ma $T(A_3) = 0$ e $12 \notin P$ in quanto $12 \in A_5$ ma $T(A_5) = 0$. Ne concludiamo che l'unica coppia che può essere positiva, compatibilmente con gli esiti dei test dati dal vettore (3), è $\{4, 10\}$. Analogamente si può verificare per altri possibili positivi.

Cerchiamo ora di comprendere di quali proprietà deve godere una generica matrice binaria M, rappresentante gli insiemi $A_1, A_2, \ldots, A_t$ da testare, affinchè essa ci permetta di scoprire gli elementi postivi $P \subseteq [n]$. Denotiamo con $M[i,j]$ il generico elemento nella riga i-esima e colonna j-esima della matrice M. Dalla (1), il generico i-esimo test $T(A_i)$ darà un risultato pari a 1 se e solo se $\exists j \in [n]$ tale che $j \in P \cap A_i$. Detto altrimenti, $T(A_i) = 1$ se e solo se per almeno un $j \in P$ vale che $M[i,j] = 1$. Ovvero, vale che

$$T(A_i) = \bigvee_{j \in P} M[i,j] \tag{4}$$

dove con $\bigvee$ intendiamo l'operatore logico OR. Se ora indichiamo con $M^1, \ldots, M^n$ le colonne della matrice M, avremo che il vettore colonna $\mathbf{x}$ di dimensione $t \times 1$ che rappresenta i risultati di tutti i test (come in (3)) sarà pari a

$$\mathbf{x} = \bigvee_{j \in P} M^j, \tag{5}$$

dove qui l'OR viene calcolato, per ciascuna componente i-esima di $\mathbf{x}$, $i = 1, \ldots, t$, in accordo alla (4). Di conseguenza, la proprietà che la matrice M deve godere per poter risalire all'insieme incognito P (qualunque esso sia), noto il vettore degli esiti $\mathbf{x}$ dato da (5), è la seguente:

$$\forall S_1, S_2 \subseteq [n], \text{ con } |S_1|, |S_2| \leq d, \quad S_1 \neq S_2 \Rightarrow \bigvee_{j \in S_1} M^j \neq \bigvee_{j \in S_2} M^j. \tag{6}$$

Infatti, se ogni possibile distinto insieme di positivi produce un differente vettore di esiti, da tale vettore sarà possibile risalire all'insieme di positivi. Matrici che godono della proprietà (6) vengono dette metrici d-separabili.

La proprietà di d-separabilità permette, quindi, di determinare chi è l'insieme degli elementi positivi, conoscendo il vettore degli esiti dei test. Tuttavia, tale "decodifica" non è agevole, in quanto non vi altra possibilità che provare per ogni $S \subseteq [n]$, con $|S| \leq d$, qual è quell'unico S compatibile con il vettore dei test ottenuti. Purtroppo, il numero di tali insiemi S è pari a $\binom{n}{d} \approx n^d$. Useremo quindi matrici che godono di una proprietà più forte della d-separabilità, che ci permetteranno di risalire agli elementi positivi, una volta noti gli esiti dei test, in maniera più agevole.

Diremo che una matrice binaria M di dimensione $t \times n$ è d-disgiunta se e solo se

$$\forall S \subseteq [n], \text{ con } |S| \leq d, \text{ e } \forall j \in [n] \setminus S \quad \exists i \in [t] \text{ tale che } M[i,j] = 1 \text{ e } \forall k \in S \text{ vale che } M[i,k] = 0. \quad (7)$$

In altri termini, la proprietà (7) dice la cosa seguente: per ogni colonna M^j della matrice M esiste una riga i in cui, all'intersezione della colonna M^j e della riga i vi è un 1, mentre per tutte altre d colonne arbitrarie l'intersezione con la stessa riga i ha solo valori 0.

Proviamo che se M è d-disgiunta allora M è anche d-separabile. In effetti, proveremo il contrappositivo, ovvero proveremo che se M *non* è d-separabile allora non è neanche d-disgiunta. Supponiamo quindi che M non sia d-separabile, ovvero

$$\exists S_1, S_2 \subseteq [n], \text{ con } |S_1|, |S_2| \leq d, \quad S_1 \neq S_2 \text{ e } \bigvee_{j \in S_1} M^j = \bigvee_{j \in S_2} M^j. \quad (8)$$

Poichè $S_1 \neq S_2$, possiamo senz'altro trovare un k tale che $k \in S_1$ e $k \notin S_2$. Ovviamente, ogni 1 che compare nella colonna M^k compare anche, nella stessa posizione, anche in $\bigvee_{j \in S_1} M^j$. Di conseguenza, dalla (8), ogni 1 che compare nella colonna M^k compare anche, nella stessa posizione, in $\bigvee_{j \in S_2} M^j$. Di conseguenza, *non* esiste alcuna posizione in cui M^k ha valore 1 e *tutte* le colonne M^j, con $j \in S_2$, hanno valore zero. Dalla (7), ciò implica che M non è d-disgiunta.

Proviamo ora il seguente risultato.

Lemma 1 *Sia M una matrice binaria, di dimensione $t \times n$, tale che*

1. *ogni colonna di M contiene almeno w 1*
2. *ogni coppia di colonne ha al più z 1 in comune, nelle stesse posizioni.*

Allora, M è d-disgiunta, per $d = \lfloor (w-1)/z \rfloor$.

Dimostrazione. Per ogni colonna M^j di M, sia m^j l'insieme degli indici $i \in [t]$ tale che la colonna M^j ha un uno nella riga i. Per provare che M è d disgiunta, occorrerà provare che per ogni $S \subset [n]$, $|S| \leq d$ e per ogni $j \in [n] \setminus S$ vale che

$$m^j \not\subseteq \bigcup_{k \in S} m^k,$$

ovvero, equivalentemente

$$m^j \not\subseteq \bigcup_{k \in S} (m^k \cap m^j),$$

il che, ovviamente, vale se e solo se vale

$$|m^j \setminus \bigcup_{k \in S} (m^k \cap m^j)| > 0.$$

Proviamo quindi quest'ultima affermazione. Si ha

$$\begin{aligned}|m^j \setminus \bigcup_{k\in S}(m^k \cap m^j)| =& |m^j| - |\bigcup_{k\in S}(m^k \cap m^j)| \\ \geq& |m^j| - \sum_{k\in S}|m^k \cap m^j| \\ \geq& w - |S|z \\ \geq& w - dz \\ \geq& w - \frac{w-1}{z}z \\ =& 1\end{aligned}$$

□

Le matrici d-disgiunte sono da preferirsi rispetto alle matrici che godono della più debole proprietà di d separabilità in quanto permettono di scoprire chi sono gli eventuali d elelementi positivi in manierà più agevole, ovvero senza dover testare, *per ogni possibile* $S \subseteq [n]$, con $|S| \leq d$, qual è quell'unico S compatibile con il vettore dei test ottenuti. Infatti, sia P l'insieme dei positivi, $|P| \leq d$, e sia $\mathbf{x} = \bigvee_{j\in P} M^j$ il vettore contenente i risultati dei test. Presa una generica colonna M^i, dalla proprietà (7) è immediato che verificare che l'insieme m^i (come definito nel Lemma precedente) è incluso nell'insieme degli indici $i \in [t]$ tale che la colonna $\mathbf{x}$ ha un 1 nella riga i, *se e solo se* $i \in P$, ovvero se e solo se i è un positivo. Basterà quindi effettuare questo controllo, iterativamente, per $M^1, M^2, \ldots, M^n$ per scoprire tutti gli elementi di P.

Dal precedente risultato otteniamo che, al fine di costruire una matrice M d-disgiunta di dimensione $t \times n$ e con $d = \lfloor (w-1)/z \rfloor$, basterà costruire un codice binario $C \subseteq \{0,1\}^t$ con le seguenti proprietà:

1. per ogni $\mathbf{c} \in C$ vale che il numero di 1 in $\mathbf{c}$ è almeno w,
2. $\forall \mathbf{c}^1, \mathbf{c}^2 \in C$ vale che $|\{i : c^1_i = c^2_i = 1\}| \leq z$,

ed usare le parole codice di C come colonne della costruenda M.

Ricordiamo che il numero di righe t della matrice d-disgiunta M corrisponde al numero di test che occorre effettuare per poter determinare i positivi della popolazione. Ovviamente, vorremmo che t sia il più piccolo possibile.

Vediamo ora come sia possibile costruire un codice siffatto a partire dai familiari codici di Reed-Solomon visti nella lezione scorsa.

Ricordiamo che ogni parola di un tale codice ha componenti in F_q, ha numero di componenti diverse da zero in ogni parola codice pari a $n - k + 1 = q - k + 1$ (perchè? perchè ogni parola codice è la valutazione in n punti di un polinomio di grado $k-1$ e quindi al più $k-1$ di tali valutazioni possono essere pari a 0, di conseguenza almeno $n - k + 1 = q - k + 1$ sono differenti da zero) ed ha un numero di parole pari a q^k (perchè? perchè q^k sono i possibili polinomi di grado $k-1$ a coefficienti in F_q).

Effettuiamo la seguente codifice dei simboli del campo F_q in vettori binari lunghi q:

$$0 \to \begin{pmatrix}1\\0\\0\\\vdots\\0\end{pmatrix} \qquad 1 \to \begin{pmatrix}0\\1\\0\\\vdots\\0\end{pmatrix} \qquad 2 \to \begin{pmatrix}0\\0\\1\\\vdots\\0\end{pmatrix} \qquad (q-1) \to \begin{pmatrix}0\\0\\0\\\vdots\\1\end{pmatrix}$$

Con tale trasformazione, otterremo da ogni parola codice di Reed-Solomon un vettore binario lungo q^2. Il numero di uni in ciascuno di questi vettori binari sarà ovviamente pari a q. Disponiamo tutti i vettori binari così ottenuti in una matrice M di dimensione $t \times N$, dove ovviamente $t = q^2$ e $N = q^k$. Per ogni coppia di colonne $\mathbf{c}^1, \mathbf{c}^2$ di M, il numero di 1 che esse hanno nelle stesse posizioni (righe di M), ovvero la quantità $|\{i : c_i^1 = c_i^2 = 1\}|$ sarà ovviamente pari al numero di indici su cui le originali parole del codice di Reed Solomon avevano esattamente lo *stesso* valore. Poichè ogni coppia di parole codice di Reed-Solomon differisce per almeno $q - k + 1$ componenti, ne segue che $|\{i : c_i^1 = c_i^2 = 1\}| = k - 1$.

Proviamo l'affermazione, ovvero che che ogni coppia di parole codice di Reed-Solomon differisce per almeno $q - k + 1$ componenti (ovvero due parole del codice di Reed-Solomon coincisono su al più $k - 1$ componenti). Ricordiamo che le parole codice sono

$$\mathcal{C} = \{(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_q)) \ : \ f(x) = \sum_{i=0}^{k-1} f_i x^i, \ f_0, \ldots, f_{k-1} \in \mathbb{F}\}.$$

Se (per assurdo) esistessero due parole codice che differiscono per **meno** di $q - k + 1$ componenti, vorrebbe dire che esse sono uguali su **almeno** k componenti.

Ovvero, per distinti polinomi $f(x)$ e $g(x)$ di grado al più $k - 1$ varrebbe

$$f(\alpha_{i_1}) = g(\alpha_{i_1}), \ldots, f(\alpha_{i_k}) = g(\alpha_{i_k}).$$

Ciò implica che per il polinomio $h(x) = f(x) - g(x)$ vale

$$h(\alpha_{i_1}) = 0, \ldots, h(\alpha_{i_k}) = 0$$

ma questo è impossibile in quanto $h(x) = f(x) = g(x) \neq 0$ ed ha grado al più $k - 1$.

Visto che la la quantità $|\{i : c_i^1 = c_i^2 = 1\}|$ è pari al numero di indici su cui le originali parole del codice di Reed Solomon avevano esattamente lo *stesso* valore, allora effettivamente vale che $|\{i : c_i^1 = c_i^2 = 1\}| = k - 1$.

Dal Lemma 1 ne segue che la matrice M appena costruita è una matrice d-disgiunta, per $d = \lfloor (q-1)/(k-1) \rfloor$. Dal fatto che $N = q^k$ otteniamo che $k = \log_q N$ e dalla espressione di d otteniamo che[1]

$$q = O(kd) = O(d \log_q N) = O(d \log_d N) = O\left(\frac{d}{\log_2 d} \log_2 N\right)$$

ovvero

$$t = q^2 = O\left(\left(\frac{d}{\log_2 d}\right)^2 \log_2^2 N\right) \tag{9}$$

Se confrontiamo quest'ultima espressione con la limitazione inferiore (2) possiamo vedere che la costruzione di matrici disgiunte via codici di Reed-Solomon non è poi così cattiva, dopo tutto[2].

É interessante notare che la proprietà (7) che definisce le matrici disgiunte le rende utili in molti altri contesti. Uno di questi riguarda la comunicazione su canali a multiaccesso. In un tale ambito si hanno un numero n di stazioni $s_1, s_2, \ldots, s_n$. Esse cercano di trasmettere informazioni, ad esempio via segnali radio, su di una *stessa* frequenza. Se in un dato istante $d \geq 2$ stazioni tentano simultaneamente di trasmettere, si crea ciò che si chiama una *collisione*, ed a causa della mutua interferenza l'informazione che esse tentavano di trasmettere viene persa. Il problema che si pone è quello di far ritrasmettere le d stazioni, in modo che la trasmissione di ciascheduna

[1] Ricordiamo che, date due funzioni $f, g : \mathbb{N} \to \mathbb{R}_+$, diciamo che $f(n) = O(g(n))$ se e solo se esistono costanti $c \in \mathbb{R}_+$ e $n_0 \in \mathbb{N}$ tali che $f(n) \leq cg(n)$, per ogni $n \geq n_0$.

[2] Notiamo che nella espressione (9) il valore N ha lo stesso significato del parametro n in (2)

stazione avvenga con successo, ovvero in modo tale che ogni stazione abbia un istante di tempo in cui essa e solo essa trasmetta. In generale, le stazioni non hanno la possibilità di coordinarsi tra di loro per stabilire chi deve trasmettere per prima, chi per seconda, etc., per cui se lasciate a loro stesse, potrebbero di nuovo ritrasmettere tutte allo stesso istante e replicare la collisione. Uno dei metodi in uso è quello di assegnare una colonna di una matrice d-disgiunta M di dimensione $t \times n$ a ciascuna stazione, ciò in fase di "fabbrica", per così dire. Ogni qualvolta si creerà un a collisione, ogni stazione legge la colonna che gli è stata assegnata. Se nella posizione j, $j \in [t]$, della colonna assegnata alla stazione s vi è uno 0, allora la stazione s *non* ritrasmette, se in tale posizione vi è un 1, allora la stazione s ritrasmette. Dalla proprietà (7), vi sarà almeno un istante $j \in [t]$ in cui la stazione s trasmette e tutte le altra staranno zitte, in quanto hanno valore 0 nella posizione j delle loro colonne. Quindi, dopo t istanti, tutte le d stazioni inizialmente in conflitto avranno avuto la possibilità di trasmettere con successo. Di nuovo, il parametro che qui si vuole ottimizzare è t, il numero di righe della matrice M, ovvero il numero di istanti entro il quale tutte le stazioni avranno avuto la possibilità di trasmettere con successo sul canale a multiaccesso.