CALCOLO MATRICIALE: APPLICAZIONI IN INFORMATICA

Analizziamo ora due semplici esempi di applicazioni informatiche del calcolo matriciale: i codici di Hamming per la correzione degli errori di trasmissione e un sistema di cifratura a matrici. Ambedue richiedono l'uso di **matrici intere** (cioè matrici i cui elementi sono numeri interi) e l'estensione alle matrici della funzione modulo studiata in algebra.

Data una matrice A (intera) con elementi $a_{ij,}$ si definisce $B = A \bmod n$ la matrice, ancora ovviamente intera, i cui elementi b_{ij}sono dati da:

$$b_{ij} = (A \bmod n)_{ij} = a_{ij} \bmod n$$

Naturalmente si ha:

$$(A \times B) \bmod n = (A \bmod n \times B \bmod n) \bmod n$$

Esempio: se $A = \begin{pmatrix} 4 & 5 \\ 7 & 6 \end{pmatrix}$ allora $A \bmod 3 = \begin{pmatrix} 1 & 2 \\ 1 & 0 \end{pmatrix}$. Inoltre, essendo:

$$\begin{pmatrix} 4 & 5 \\ 7 & 6 \end{pmatrix}^{-1} = \begin{pmatrix} -\frac{6}{11} & \frac{5}{11} \\ \frac{7}{11} & -\frac{4}{11} \end{pmatrix}$$

otteniamo:

$$A^{-1} \bmod 3 = \begin{pmatrix} -\frac{6}{11} & \frac{5}{11} \\ \frac{7}{11} & -\frac{4}{11} \end{pmatrix} \bmod 3 = \begin{pmatrix} 0 & 1 \\ 2 & 1 \end{pmatrix}$$

Allora, essendo:

$$(A \times A^{-1}) \bmod 3 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \bmod 3 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

possiamo verificare il risultato:

$$\begin{pmatrix} 4 & 5 \\ 7 & 6 \end{pmatrix} \bmod 3 \times \begin{pmatrix} -\frac{6}{11} & \frac{5}{11} \\ \frac{7}{11} & -\frac{4}{11} \end{pmatrix} \bmod 3 = \begin{pmatrix} 1 & 2 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 2 & 1 \end{pmatrix} \bmod 3 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Notiamo che per quanto riguarda l'invertibilità $\bmod n$, una matrice deve avere determinante diverso da zero *modulo* n. Ad esempio $\begin{pmatrix} 1 & -3 \\ 6 & 7 \end{pmatrix}$ è invertibile $\begin{pmatrix} 1 & -3 \\ 6 & 7 \end{pmatrix}^{-1} = \begin{pmatrix} \frac{7}{25} & \frac{3}{25} \\ -\frac{6}{25} & \frac{1}{25} \end{pmatrix}$, ma non è invertibile $\bmod 5$, infatti $\begin{pmatrix} 1 & -3 \\ 6 & 7 \end{pmatrix} \bmod 5 = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}$ che ha determinante nullo.

I codici di Hamming:

Un codice di Hamming di tipo (n, m) - ad esempio $(n = 4, m = 7)$ - per la correzione degli errori opera su una parola (detta messaggio) di 4 bit e genera un codice a 7 bit. Definiamo una matrice H a 4 righe e 7 colonne come segue:

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Dato un messaggio m, ad esempio $\begin{bmatrix} 1 & 0 & 1 & 1 \end{bmatrix}$ il corrispondente codice è dato da $mH \bmod 2 = c$:

$$\begin{bmatrix} 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \bmod 2 = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Per la struttura della matrice H il messaggio appare sempre codificato nei bit 3, 5, 6, and 7. I 3 bit restanti sono i bit di controllo che correggono gli eventuali errori. Il codice c è poi trasmesso e ricevuto come r . Il ricevente esegue un'altra moltiplicazione con la matrice P:

$$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

P è tale che $HP \bmod 2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ e inoltre **gli elementi di ogni riga forniscono la rappresentazione binaria del numero della riga stessa.**

Se non ci sono errori di trasmissione, $c = r$ e quindi, siccome $cP \bmod 2 = mHP \bmod 2 = 0$, si deve ottenere $rP \bmod 2 = 0$.

L'affermazione (vera sempre per costruzione): *se non ci sono errori allora ottengo il risultato nullo* **implica logicamente solo che** *se non ottengo il risultato nullo allora ci sono stati errori* (ma si potrebbe benissimo - sebbene con probabilità piccola, vedi più sotto - ottenere il risultato nullo anche in presenza di due o più errori). Il meccanismo logico del codice è tale per cui un certo risultato non può dire che certamente non ci sono errori, ma solo che, molto probabilmente, non ce ne sono.

Quindi un risultato nullo $rP \bmod 2 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$ indica che - **molto probabilmente** - non ci sono stati errori di trasmissione e il messaggio m è costituito proprio dai bit 3, 5, 6, and 7 del codice. Se invece il codice viene ricevuto con un solo errore, ad esempio $r = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$, allora si otterrebbe $rP \bmod 2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$, che rappresenta in binario il numero decimale 2, e quindi l'errore coinvolge la seconda riga della matrice P (perchè $rP \bmod 2 = (c - r)P \bmod 2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} P \bmod 2 =$ alla seconda riga di P) e quindi è errato il secondo bit del codice!. Così il codice corretto era invece $\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$. Il messaggio si legge quindi sulle colonne 3, 5, 6, e 7 del codice corretto e risulta essere: $\begin{bmatrix} 1 & 0 & 1 & 1 \end{bmatrix}$. Se l'errore è invece nel settimo bit, lo possiamo analogamente rilevare e correggere: $\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} P \bmod 2 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$ che rappresenta proprio il numero 7. *Naturalmente questo procedimento funziona solo in presenza di nessuno o un solo errore.* Due o più errori possono infatti trarre in inganno: ad esempio in $\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$ sono errati il secondo ed il settimo bit, ma $\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} P \bmod 2 = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}$ che rappresenta il numero 5. Peggio ancora, in $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ ci sono tre errori ma, ovviamente, $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} P \bmod 2 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$.

Naturalmente la probabilità di avere errori multipli su codici corti è molto bassa ed è per questo motivo che i codici di questo tipo funzionano. Nessun codice di lunghezza "accettabile" (neanche quelli più moderni e raffinati) può rilevare e correggere un numero molto elevato di errori. Tutti i codici di correzione sono "probabilistici". A titolo di curiosità: se $1/100$ *è la probabilità di avere un errore su un singolo bit, e si assume che gli errori siano indipendenti, la probabilità di avere nessuno o un solo errore in una parola di* 7 *bit è molto alta :*

$$P = (1 - 1/100)^7 + 7(1 - 1/100)^6/100 = 0.998$$

La formula si ricava così: se p *è la probabilità che si verifichi un evento (ad esempio la presenza di un errore), allora* $1-p$ *è la probabilità che non si verifichi. La probabilità* P *che si verifichi l'evento* k *volte in* n *prove indipendenti è allora* $\binom{n}{k} p^k (1-p)^{n-k}$. *Infatti* $\binom{n}{k}$ *indica il numero di modi differenti in cui si possono scegliere tra le* n *prove le* k *in cui si verifica l'evento. Inoltre*

la probabilità di ciascuna di queste scelte è, per l'indipendenza, il prodotto delle singole probabilità (k errori e $n-k$ bit corretti), cioè: $p^k(1-p)^{n-k}$.

Studiamo ora un altro esempio in cui un codice di questo tipo trova due errori e ne corregge uno. L'analisi di questo esempio ci consente anche di comprendere a fondo il funzionamento di un codice di Hamming. Consideriamo la matrice K ottenuta dalla **trasposta** della matrice P del codice precedente aggiungendo una riga tutta di 1 e una colonna iniziale come segue:

$$K = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Passiamo ora alla notazione con i vettori colonna. Siano $m = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \bmod 2$ il messaggio e $c = \begin{bmatrix} w \\ x \\ y \\ a \\ z \\ b \\ c \\ d \end{bmatrix} \bmod 2$ il codice corrispondente, dove, come sopra, il codice è determinato dalla condizione $Kc \bmod 2 = 0$:

$$Kc \bmod 2 = \begin{bmatrix} w+x+y+a+z+b+c+d \\ x+a+b+d \\ y+a+c+d \\ z+b+c+d \end{bmatrix} \bmod 2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

La soluzione del sistema è (attenzione, le soluzioni sono *modulo 2 ,* notate ad esempio che $-a \bmod 2 = a \bmod 2$!)

$$\begin{cases} w = (a+b+c) \bmod 2 \\ x = (a+b+d) \bmod 2 \\ y = (a+c+d) \bmod 2 \\ z = (b+c+d) \bmod 2 \end{cases}$$

la matrice H di partenza sarebbe allora:

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Infatti

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} a+b+c \\ a+b+d \\ a+c+d \\ a \\ b+c+d \\ b \\ c \\ d \end{pmatrix}$$

Trasmettiamo c e supponiamo di ricevere invece r, ci sono allora quattro casi:

a) se non ci sono errori, allora $Kr \bmod 2 = 0$.

b) se c'è un solo errore, sia $l = r - c$. Allora $Kr \bmod 2 = Kl \bmod 2 + Kc \bmod 2 = Kl \bmod 2$. La colonna l contiene un solo elemento diverso da zero (perchè supponiamo che ci sia un solo errore), per cui $Kr \bmod 2$ risulta essere la colonna di K corrispondente alla posizione del bit errato che può quindi essere rilevato e corretto perchè tutte le colonne di K sono diverse.

c) se ci sono due errori, l contiene due elementi diversi da zero e $Kr \bmod 2$ risulta essere la somma di due colonne di K, anche se non sappiamo quali. Quello che è certo, però, è che $Kr \bmod 2$ non può essere una colonna di K (perchè la somma di due colonne di K ha come prima componente 0, mentre tutte le colonne di K hanno prima componente uguale a 1) abbiamo identificato così due errori ma non possiamo correggerli.

d) se ci sono tre o più errori non possiamo più capire nulla.

Ricapitoliamo: se $Kr \bmod 2 = 0$ molto probabilmente non ci sono errori, se $Kr \bmod 2 \neq 0$ **ma è una colonna di** K molto probabilmente c'è un errore nella posizione corrispondente alla colonna trovata, se $Kr \bmod 2 \neq 0$ **ma non è una colonna di** K molto probabilmente ci sono stati due o più errori.

Esercizio: supponiamo che ci possano essere al più due errori e decodifichiamo i seguenti codici:

$$c_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}, c_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Soluzione: calcoliamo $K \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \bmod c2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$ quindi ci sono certamente due errori. $K \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \bmod 2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ = **ottava colonna di** K, quindi c'è un errore nell'ottavo bit e quindi il messaggio corretto è $m = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

Crittografia a matrici.

Le lettere dell'alfabeto sono 26, associamo *un numero modulo* 26 ad ogni lettera:

$$\left\{ \begin{array}{cccccc} A & B & C & ... & Y & Z \\ 1 & 2 & 3 & ... & 25 & 26 \end{array} \right\}$$

Prendiamo il messaggio "AUGURI"

Il messaggio diventa il vettore m

$$m = \begin{pmatrix} 1 & 21 & 7 & 21 & 18 & 9 \end{pmatrix}$$

Un cifrario matriciale (ad esempio con matrici 2×2) consiste nello spezzare il messaggio in blocchi di 2 elementi,

$$\{(1,21),(7,21),(18,9)\}$$

Prendere poi una matrice A *invertibile modulo 26:*

$$A = \begin{pmatrix} 8 & 13 \\ -5 & -8 \end{pmatrix}$$
$$A^{-1} = \begin{pmatrix} 8 & 13 \\ -5 & -8 \end{pmatrix}^{-1} \bmod 26 = \begin{pmatrix} 18 & 13 \\ 5 & 8 \end{pmatrix}$$

Il messaggio cifrato si ottiene moltiplicando, *modulo 26,* ogni coppia per A *:*

$$\{(21,9),(17,5),(1,20)\} = \text{UIQEAT}$$

Si decodifica rimoltiplicando ogni coppia per A^{-1}. Naturalmente si possono prendere matrici di dimensione maggiore. L'interesse di questi sistemi (che soffrono però del problema di avere bisogno di una chiave segreta (la matrice A e l'associazione iniziale lettere-numeri) che deve essere nota al mittente e al ricevente) è che generano cifrature in cui una stessa lettera, a seconda della sua posizione, viene crittografata in maniera diversa (aUgUri diventa uIqEat), rendendo il codice relativamente resistente alle analisi statistiche.