Sistemi di equazioni lineari

Note Title 2021-10-18

Prodotto matrice-vettore: $A \in \mathbb{C}^{m \times n}$ (m righe, n colonne)

$v \in \mathbb{R}^n$ $\quad w = Av \in \mathbb{R}^m$

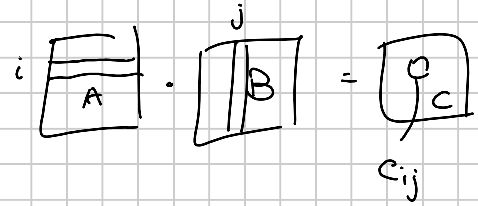
$$w_i = \sum_{j=1}^{n} A_{ij} v_j \qquad A = \left[a_1 | a_2 | \cdots \; a_n \right] \quad a_i \in \mathbb{R}^m \text{ colonne della matrice } A$$

$$\vec{w} = \underset{\text{vettori}}{\vec{a}_1} \underset{\text{scalari}}{v_1} + \vec{a}_2 v_2 + \dots + \vec{a}_n v_n$$

combinazione lineare degli a_i con coefficienti $v_1, v_2 \dots v_n$

Prodotto matrice-matrice:

$$AB = C \qquad C_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj} \quad \forall i,j$$



$$A \in \mathbb{C}^{m \times n} \quad B \in \mathbb{C}^{n \times p} \rightarrow C \in \mathbb{C}^{m \times p}$$

dimensioni interne devono coincidere perché sia ben definito

⚠ L'ordine conta: $AB \neq BA$
(magari uno non è neppure ben definito)

(estende la definizione sopra, se consideriamo v come una matrice $n \times 1$)

Proprietà algebriche:

$$(A+B)C = AC + BC \qquad (AB)C = A(BC)$$

⚠ Non vale annullamento del prodotto: ad es, $\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = O_{2 \times 2}$

Possiamo moltiplicare matrici/vettori per scalari: $3 \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} -3 & 3 \\ -3 & 3 \end{bmatrix}$

Potete "spostare" scalari dentro i prodotti: $A(\underset{\text{scalare}}{\alpha} B) = \alpha(AB) = (AB)\alpha$
(cambiare ordine di fattori scalari)

Sistemi lineari: dati $\underset{\text{matrici}}{A}$, b trovare x t.c. $Ax = b$ (b, x: vettori)

Sistemi quadrati, $A \in \mathbb{C}^{n \times n}$ $\quad x \in \mathbb{C}^n$, $b \in \mathbb{C}^n$

Ammettono una e una sola soluzione $\Leftrightarrow$ le colonne di A sono lin.

indipendenti, e quindi sono una base di $\mathbb{C}^n \Leftrightarrow A$ è <u>invertibile</u>

Ci concentriamo su sistemi con A invertibile.

La soluzione è data dal prodotto di b per un'altra matrice:

$x = A^{-1} b$ $\quad$ A^{-1} è (l'unica) matrice tale che $AA^{-1} = A^{-1}A = I = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$.

matrice identità

⚠ <u>non</u> possiamo scrivere $\underset{n\times 1 \neq n \times n}{x = bA^{-1}}$ né $x = \frac{b}{A}$ ← no matrici a denominatore

Matlab ha $\mathrm{inv}(A)$, ma $\underbrace{\mathrm{inv}(A) * b}$ non è il metodo migliore per risolvere sist. lineari.

Calcolare $\mathrm{inv}(A) \Leftrightarrow$ risolvere n sistemi lineari $A^{-1}e_1$, $A^{-1}e_2$, ... $A^{-1}e_n$

$\Updownarrow$

$Ax_i = e_i$

$e_i = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ ← 1 in posizione i

Altri cattivi algoritmi: quelli basati su determinanti → più costosi + problemi di overflow / underflow

<u>es</u>: $\det \underbrace{\begin{bmatrix} 10 & & & 0 \\ & 10 & & \\ & & 10 & \\ 0 & & \ddots & 10 \end{bmatrix}}_{400\times 400} = 10^{400}$ → al di fuori del "range" $[10^{-308}, 10^{308}]$

$\det \underbrace{\begin{bmatrix} 0.1 & & & \\ & 0.1 & & \\ & & \ddots & \\ & & & 0.1 \end{bmatrix}}_{400\times 400} = 10^{-400}$

Nelle prossime lezioni: algoritmi migliori

In Matlab: `>> x = A \ b`

barra rovesciata, backslash

mnemonica: A \ b ricorda $\frac{b}{A}$, la A sta sotto.

(Matlab ha anche $C / A = C * A^{-1}$)

Algoritmi diretti per la soluzione di sistemi lineari

Partiamo da casi semplici.

A diagonale
$$A = \begin{bmatrix} A_{11} & 0 & 0 & 0 & \cdots & 0 \\ 0 & A_{22} & 0 & & & \vdots \\ 0 & 0 & A_{33} & 0 & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & & & 0 & A_{nn} \end{bmatrix} = \begin{bmatrix} A_{11} & & & \\ & A_{22} & & \\ & & A_{33} & \\ & & & \ddots & \\ & & & & A_{nn} \end{bmatrix}.$$

$$\Rightarrow \begin{bmatrix} A_{11} & & & \\ & A_{22} & & \\ & & \ddots & \\ & & & A_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} A_{11}x_1 \\ A_{22}x_2 \\ \vdots \\ A_{nn}x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$x_i = \frac{b_i}{A_{ii}} \qquad i = 1, 2, \ldots n$$

n operazioni aritmetiche (una per ogni $i = 1, 2, \ldots n$)

Matrici triangolari

A triangolare inferiore se

$$A = \begin{bmatrix} A_{11} & 0 & 0 & \cdots & 0 \\ A_{21} & A_{22} & 0 & & \vdots \\ A_{31} & A_{32} & A_{33} & \ddots & 0 \\ \vdots & & & \ddots & 0 \\ A_{n1} & & & & A_{nn} \end{bmatrix} \qquad (A_{ij} = 0 \text{ se } i < j)$$

Algoritmo: sostituzione in avanti

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} A_{11} & & & \\ A_{21} & A_{22} & & \\ A_{31} & A_{32} & A_{33} & \\ \vdots & & \ddots & \\ A_{n1} & \cdots & & A_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} A_{11}x_1 \\ A_{21}x_1 + A_{22}x_2 \\ A_{31}x_1 + A_{32}x_2 + A_{33}x_3 \\ \vdots \\ A_{n1}x_1 + A_{n2}x_2 + \cdots + A_{nn}x_n \end{bmatrix}$$

Dalla prima riga, $x_1 = \dfrac{b_1}{A_{11}}$

Dalla seconda, $b_2 = A_{21}x_1 + A_{22}x_2 \qquad x_2 = \dfrac{b_2 - A_{21}x_1}{A_{22}} \qquad x_3 = \dfrac{b_3 - A_{31}x_1 - A_{32}x_2}{A_{33}}$

$\vdots$

i-esima riga: $x_i = \dfrac{b_i - \sum_{j=1}^{i-1} A_{ij} x_j}{A_{ii}}$ ← già calcolati ai passi precedenti

"sostituzione in avanti" $(1, 2, 3, \ldots n)$

Costo (in termini di operazioni aritmetiche):

1 operazione per x_1
3 operazioni per x_2
5 operaz. per x_3
$\vdots$
$2n-1$ operaz. x_n

$$1+3+5+7\ldots+2n-1 = n^2$$

$1 \quad 4 \quad 9 \quad 16$

per induzione, se volete dimostrarlo

(Calcoliamo la soluzione esatta, quindi l'errore analitico è zero)

Matrici triangolari superiori

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1n} \\ 0 & A_{22} & A_{23} & \cdots & A_{2n} \\ 0 & 0 & \ddots & & \\ \vdots & \vdots & & & \\ 0 & 0 & \cdots & 0 \; A_{n-1,n-1} & A_{n-1,n} \\ 0 & 0 & \cdots & 0 \quad 0 & A_{n,n} \end{bmatrix} \qquad A_{ij} = 0 \text{ se } i > j$$

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & \cdots & & A_{1n} \\ & A_{22} & \cdots & & A_{2n} \\ & & \ddots & & \\ & & & A_{n-1,n-1} & A_{n-1,n} \\ & & & & A_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 + \cdots + A_{1n}x_n \\ A_{22}x_2 + A_{23}x_3 + \cdots + A_{2n}x_n \\ \vdots \\ A_{n-1,n-1}x_{n-1} + A_{n-1,n}x_n \\ A_{n,n}x_n \end{bmatrix}$$

"sostituzione all'indietro": parto a calcolare x_n:

$$x_n = \frac{b_n}{A_{nn}}$$

$$x_{n-1} = \frac{b_{n-1} - A_{n-1,n}x_n}{A_{n-1,n-1}}$$

$$\vdots$$

$$x_2 = \frac{b_2 - A_{23}x_3 - \cdots - A_{2n}x_n}{A_{22}}$$

$$x_1 = \frac{b_1 - A_{12}x_2 - \cdots - A_{1n}x_n}{A_{11}}$$

tutti noti dei passi precedenti

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} A_{ij}x_j}{A_{i,i}}$$

complessità $1+3+5+\cdots 2n-1 = n^2$

Oss: divido sempre per elementi diagonali, A_{ii}.

Se A è diagonale, triang. inf. o triang. sup, allora è invertibile se e solo se $A_{ii} \neq 0$ per ogni $i = 1, 2, \ldots n$

(dim. per esempio notando che $\det(A) = A_{11}A_{22}A_{33} \cdots A_{nn}$).

Osservazione: se una matrice triang. inf/sup ha molti

elementi nulli (anche dal lato dove non sono tutti zeri),
posso ignorarli nelle sommatorie

$\rightsquigarrow$ il costo è $\leq 2 \cdot nnz(A)$, numero di elementi non-zero di A

Abbiamo studiato l'errore di algoritmi del tipo "calcola $y = f(x)$"
$y \in \mathbb{R}$, $x \in \mathbb{R}$.

Questo problema: calcola $x = f(A, b)$. $\qquad x_3 = f(A, s)$

matrice

vettori

Generalizziamo i concetti di "errore", "distanza", "valore assoluto" a vettori e matrici

<u>Norma vettoriale</u>: una funzione $f: \mathbb{C}^n \to \mathbb{R}$ si dice norma vettoriale se soddisfa queste proprietà

1) $f(v) \geq 0$ per ogni $v \in \mathbb{C}^n$, e $f(v) = 0$ se e solo se $v = 0$
2) $f(\alpha v) = |\alpha| \cdot f(v)$ per ogni $\alpha \in \mathbb{C}$, $v \in \mathbb{C}^n$
3) $f(v + w) \leq f(v) + f(w)$ per ogni $v, w \in \mathbb{C}^n$

generalizza la disuguaglianza triangolare

$\|v\|$