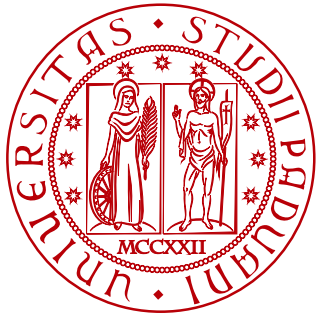


UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Vettori e Matrici

Corso di Calcolo Numerico

29 Aprile 2019

Richiami

In MATLAB, ogni variabile ha una struttura di tipo vettoriale o array. Un array è un insieme di valori ordinati, cioè memorizza più dati all'interno di una struttura identificata da un singolo nome di variabile.

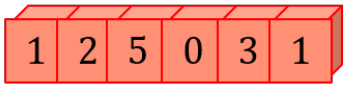
- Un array ha due dimensioni: la prima dimensione rappresenta il numero di righe, la seconda il numero di colonne
- ARRAY ad un indice $->$ VETTORE
- ARRAY a due indici $->$ MATRICE
- Scalare: ARRAY 1 x 1
  Vettore riga: ARRAY 1 x n
  Vettore colonna: ARRAY n x 1
  Matrice: ARRAY n x m

Richiami

Vettore

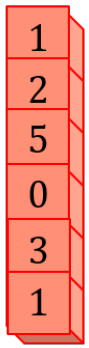
Riga [1x6]

$x = [1 \;\; 2 \;\; 5 \;\; 0 \;\; 3 \;\; 1]$



Colonna [6x1]

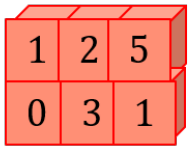
$x = [1; 2; 5; 0; 3; 1]$



Matrice

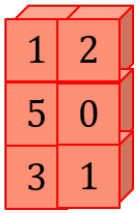
2 x 3

$A = [1 \;\; 2 \;\; 5 \; ; \; 0 \;\; 3 \;\; 1]$



3 x 2

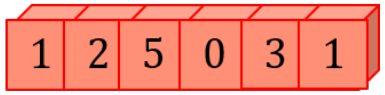
$A = [1 \;\; 2; \;\; 5 \;\; 0; \;\; 3 \;\; 1]$



Richiami

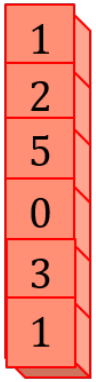
Per passare da vettori riga a vettori colonna si utilizza il simbolo di apice, operazione che dal punto di vista dell'algebra lineare corrisponde alla trasposizione.

$$x = [1 \;\; 2 \;\; 5 \;\; 0 \;\; 3 \;\; 1]$$



Riga [1x6]

$$y = [1 \;\; 2 \;\; 5 \;\; 0 \;\; 3 \;\; 1]'$$



Colonna [6x1]

Funzioni utili di MATLAB

`length`	`length(x)` restituisce la lunghezza del vettore `x`
`size`	`size(A)` restituisce un vettore con numero di righe e colonne di `A`
`'`	serve per ottenere la trasposta di una matrice o vettore
`who`	indica quali variabili sono presenti nell'ambiente di lavoro
`whos`	restituisce informazioni sulla dimensione, spazio occupato e tipo di variabile

Funzioni utili di MATLAB

Funzioni MATLAB che consentono di costruire particolari matrici e vettori.
Si consulti l'help per una descrizione dettagliata.

`linspace`	vettore riga di elementi equispaziati
`zeros`	matrice contenente solo elementi uguali a zero
`ones`	matrice contenente solo elementi uguali a uno
`eye`	matrice identità
`diag`	matrice diagonale
`magic`	matrice a valori interi con somme uguali su righe e colonne
`tril e triu`	estraggono la parte triangolare inferiore e superiore
`inv`	restituisce matrice inversa (applicazione a matrici quadrate)
`det`	restituisce il determinante di matrici quadrate
`eig`	per calcolare autovalori e autovettori di matrici quadrate

Pre-allocazione di memoria

Per ottenere codice più veloce, è possibile effettuare la pre-allocazione di vettori e matrici, ossia l'allocazione in memoria dei vettori e delle matrici prima del loro utilizzo (usando la funzione `zeros` o `ones`)

```
% Codice non ottimizzato
n = 1000;
tic
for i=1:n % Ciclo per la costruzione della matrice A
    for j=1:n
        A(i,j) = i + j;
    end
end
time1 = toc;
% Codice ottimizzato
tic
B = zeros(n); % Pre - allocazione delle variabile B in cui salvare i risultati
for i=1:n % Ciclo per costruire la matrice B precedentemente allocata
    for j=1:n
        B(i,j) = i + j;
    end
end
time2 = toc;
```

Operazioni tra Array

```
>> x = 1:5;
>> y = [50 10 30 40 20];
```

Moltiplicazione di un array per uno scalare

```
>> 2*x
ans =
2 4 6 8 10
```

Somma e differenza tra array

```
>> x+y
ans =
51 12 33 44 25
>> y-x
ans =
49 8 27 36 15
```

Operazioni tra Array

Moltiplicazione puntuale, divisione ed elevamento a potenza puntuale

```
>> x.*y
ans =
50 20 90 160 100
>> y./x
ans =
50 5 10 10 4
>> y.^x
ans =
50 100 27000 2560000 3200000
```

Operazioni tra Array

Prodotto vettore-vettore

Dati due vettori (colonna) **x** e **y** di lunghezza n, il prodotto scalare si scrive come:

$$s = x^T y = \sum_{i=1}^{n} x_i y_i$$

```
s = 0.0;
for i=1:n
    s = s + x(i)*y(i);
end
```

Operazioni tra Array

Prodotto matrice-vettore

Data una matrice quadrata A di dimensioni $m \times n$ e un vettore colonna **x** di lunghezza n, il prodotto matrice-vettore $y = Ax$ si scrive in componenti come:

$$y_i = \sum_{j=1}^{n} A_{i,j} x_j \text{ per } i = 1, 2, ..., m$$

```
A = load('matrice.dat');
x = load('vettore.dat');
[m,n]=size(A);
y = zeros(m,1);
for i=1:m
    for j=1:n
        y(i) = y(i) + A(i,j)*x(j);
    end
end
```

Operazioni tra Array

Prodotto matrice-matrice

Data una matrice quadrata A di dimensioni $m \times p$ e B di dimensioni $p \times n$, il prodotto matrice-matrice, $C = AB$, si scrive in componenti come:

$$C_{i,j} = \sum_{k=1}^{p} A_{i,k} B_{k,j} \text{ per } i = 1, 2, ..., m \text{ e } j = 1, 2, ..., n$$

```
[m,p]=size(A);
[~,n]=size(B);
C = zeros(m,n);
for i=1:m
    for j=1:n
        for k=1:p
            C(i,j)=C(i,j)+A(i,k)*B(k,j);
        end
    end
end
```

In pratica ...

In MATLAB, lo stesso risultato si ottiene direttamente utilizzando l'operatore di moltiplicazione `*`:

Prodotto vettore-vettore

```
s = x*y
```

Nota Bene: `x` vettore riga e `y` vettore colonna

Prodotto matrice-vettore

```
y = A*x
```

Prodotto matrice-matrice

```
C = A*B
```

Elevamento a potenza

Attenzione alla differenza tra gli operatori `.^` e `^`.
L'elevamento a potenza `^` si applica solo a matrici quadrate:

```
>> B = A^p;
```

$$A = \begin{pmatrix} 1 & 4 & -1 \\ 3 & 2 & 0 \\ 0 & 0 & 8 \end{pmatrix}$$

$$B = A^p = \underbrace{\begin{pmatrix} 1 & 4 & -1 \\ 3 & 2 & 0 \\ 0 & 0 & 8 \end{pmatrix} * \ldots * \begin{pmatrix} 1 & 4 & -1 \\ 3 & 2 & 0 \\ 0 & 0 & 8 \end{pmatrix}}_{\text{p volte}}$$

L'operatore :

L'operatore : permette di estrarre sotto-matrici da una matrice nota.

Esempio 1

```
>> A = [1 2 3 4; 5 6 7 8; 9 10 11 12];
>> C = A(:,1);
>> R = A(1,:);
>> C23 = A(:,2:3);
>> C14 = A(:,[1 4]);
```

Esempio 2

```
>> vert = [0 1 1 0; 0 0 1 1];
>> x = [vert(1,:)  vert(1,1)];
>> y = [vert(2,:)  vert(2,1)];
>> plot(x,y,'r');
```

Autovalori e Autovettori

Esempio

Trovare autovalori e autovettori della matrice A:

$$A = \begin{pmatrix} 5 & 8 & 9 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

`>> lambda = eig(A);`	restituisce in `lambda` gli autovalori di A
`>> [V,L] = eig(A);`	restituisce in `V` la matrice le cui colonne sono gli AUTOVETTORI di A e in `L` una matrice diagonale i cui elementi diagonali sono i corrispondenti AUTOVALORI di A.

Autovalori e Autovettori

Esempio

Trovare autovalori e autovettori della matrice A:

$$V = \begin{pmatrix} 1.0000 & -0.9363 & -0.1741 \\ 0 & 0.3511 & -0.6963 \\ 0 & 0 & 0.6963 \end{pmatrix}$$

$$L = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Gli autovettori di A sono normalizzati.

Teorema di Gershgorin

Sia A una matrice quadrata di ordine n. Se indichiamo con U_r e U_c l'unione dei cerchi costruiti nel piano complesso aventi come centro gli elementi diagonali e come raggio la somma dei moduli degli elementi extra-diagonali posti su ciascuna riga e colonna, rispettivamente, allora tutti gli autovalori di A saranno contenuti nella regione piano $U = U_r \cap U_c$.

ESERCIZIO

Implementare in MATLAB una function per disegnare i cerchi di Gershgorin della matrice A nel piano complesso. Plottare nello stesso grafico gli autovalori di A.

Teorema di Gershgorin

```
% Gershgorin's circles of the matrix A.
d = diag(A);
cx = real(d);
cy = imag(d);
B = A - diag(d);
[~, n] = size(A);
rig = sum(abs(B'));
col = sum(abs(B));
for i=1:n
    if rig(i) < col(i)
       raggio(i) = rig(i);
    else
       raggio(i) = col(i);
    end
end
lambda = eig(A);
lx = real(lambda);
ly = imag(lambda);
t = linspace(0,2*pi,100);
hold on; grid on; axis equal;
xlabel('Re')
ylabel('Im')
for i=1:n
    x = cx(i) + raggio(i)*cos(t);
    y = cy(i) + raggio(i)*sin(t);
    plot(x,y);
end
plot(lx,ly,'o');
hold off
title('Gershgorin circles and the eigenvalues of a matrix')
```

ESERCIZI

1. Scrivere un programma che carichi da file i vettori x, y:

$$x = \begin{pmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \end{pmatrix}, \qquad y = \begin{pmatrix} 2.0 \\ 0.0 \\ 0.0 \\ -1.0 \\ 0.0 \end{pmatrix}$$

e la matrice A:

$$A = \begin{pmatrix} 1.0 & 2.0 & 3.0 & 4.0 & 5.0 \\ -1.0 & 0.0 & 2.0 & 0.0 & -1.0 \\ 2.0 & 1.0 & 5.0 & 4.0 & 0.0 \end{pmatrix}$$

Eseguire il calcolo del prodotto scalare, $x^T y$, e del prodotto matrice-vettore, $b = Ax$, implementando le functions `prodscal` e `matvet`, rispettivamente. Verificare la correttezza dell'implementazione utilizzando l'operatore di moltiplicazione `*`.

ESERCIZI

2. Costruire la matrice quadrata A di dimensione n=10 con elementi sulla diagonale principale uguali a 6 e sulla sovra e sotto-diagonale uguali a -2. Costruire poi la matrice $B = A * A$ e, dato un vettore unitario x, fare il prodotto matrice-vettore $y = B * x$
3. Creare la matrice diagonale D di dimensione n=6 con elementi diagonali 2,4,6,8,10,12 e la matrice A di dimensione n=6 con elementi tutti pari a 1. Calcolare i prodotti matrice-matrice AD e DA. Commentare il risultato.

ESERCIZI

4. Data la matrice A:

$$A = \begin{pmatrix} 1 & -2 & -2 \\ 1 & 1 & -3 \\ -2 & 2 & 1 \end{pmatrix}$$

Rappresentare i cerchi di Gershgorin e gli autovalori di A. A partire dalla matrice A, costruire la matrice B che abbia autovalori reali e positivi.

5. Data la matrice A:

$$A = \begin{pmatrix} 5 & 8 & 9 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

verificare che sia simile alla matrice B:

$$B = \begin{pmatrix} 19.20 & 21.00 & 21.75 \\ -4.75 & -4.00 & -5.25 \\ -6.75 & -8.00 & -7.25 \end{pmatrix}$$

ESERCIZI

5. Data la matrice A:

$$A = \begin{pmatrix} -1 & 5 & 5 & 6 \\ 1 & 2 & 0 & 0 \\ 9 & 6 & 6 & 5 \\ 5 & 7 & 5 & 0 \end{pmatrix}$$

   calcolarne una matrice simile B e verificare che abbiano gli stessi autovalori.

6. Determinare una matrice quadrata reale A di ordine 3 che possieda gli autovalori $\lambda_1 = 1$, $\lambda_2 = 2$ e $\lambda_3 = 3$ corrispondenti ai seguenti autovettori:
   $\mathbf{u_1} = [2, 3, -2]^T$ $\mathbf{u_2} = [9, 5, 4]^T$ $\mathbf{u_3} = [4, 4, -1]^T$