

Vettori e matrici in Matlab

Federico Poloni <fpoloni@di.unipi.it>

16 marzo 2015

1 Vettori e matrici

MATLAB è pensato per lavorare con vettori e matrici; pertanto, ha una sintassi specifica e parecchi comandi dedicati, che rendono molto più semplice lavorare con i vettori rispetto a un linguaggio generico come il C.

1.1 Creare vettori e matrici

```
>> A=[1 2 3; 4 5 6]
```

```
A =  
 1 2 3  
 4 5 6
```

```
>> zeros(3,2)
```

```
ans =  
 0 0  
 0 0  
 0 0
```

```
>> ones(3,2)
```

```
ans =  
 1 1  
 1 1  
 1 1
```

```
>> eye(3)
```

```
ans =  
 1 0 0  
 0 1 0  
 0 0 1
```

```
>> randn(2,3)
```

```
ans =  
 0.567178 -0.126397 -0.090664  
-0.678601 0.504481 0.754911
```

1.2 Il range operator :

Con la sintassi `a:t:b` creiamo un vettore (riga) che contiene gli elementi `a`, `a+t`, `a+2t...` fino a `b` (o fino all'ultimo che sia minore o uguale a `b`). Se `t=1`, può essere omesso.

```
>> 1:0.5:4
ans =

    1.0000  1.5000  2.0000  2.5000  3.0000  3.5000  4.0000

>> 1:10
ans =

    1  2  3  4  5  6  7  8  9  10

>> 1:2:10
ans =

    1  3  5  7  9
```

Dove avete già usato l'operatore `:`?

1.3 Accedere agli elementi

```
>> A=ones(2,3)
A =

    1  1  1
    1  1  1

>> A(1,2)=2
A =

    1  2  1
    1  1  1

>> A(1,2)
ans = 2
>> A(5,10)
error: invalid row index = 5
error: invalid column index = 10
>> A(5,10)=7
A =

    1  2  1  0  0  0  0  0  0  0
```

```
1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 7
```

Se cerco di *leggere* un elemento che non esiste (perché la matrice è troppo piccola), ottengo un errore. Se cerco di *scrivere* un elemento che non esiste, la matrice viene automaticamente ingrandita.

1.4 Operazioni su vettori

```
>> a=1:3
a =

    1    2    3

>> b=4:6
b =

    4    5    6

>> a+b
ans =

    5    7    9

>> sin(a)
ans =

    0.84147    0.90930    0.14112

>> 2*a+1
ans =

    3    5    7

>> a.*b %operazioni elemento per elemento
ans =

    4   10   18

>> c=a' %matrice trasposta
c =

    1
    2
```

```
3
>> C=a'*b %prodotto matrice-matrice
C =

    4  5  6
    8 10 12
   12 15 18

>> length(a) %lunghezza di un vettore
ans = 3

>> size(C) %dimensioni di una matrice - (righe, colonne)
ans =

    3  3
```

2 Ottenere aiuto

Con il comando `help nomefunzione` potete ottenere aiuto su un comando di Matlab. Per esempio:

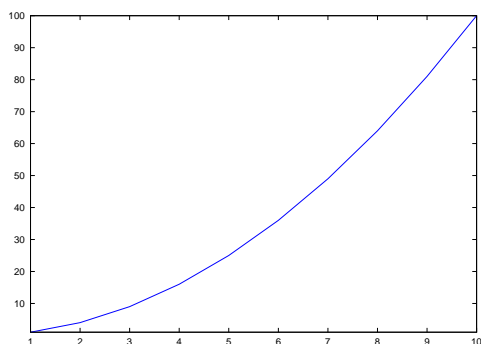
3 Grafici

Il comando `plot(x,y)` prende come argomenti due vettori della stessa lunghezza `x` e `y` e disegna sul piano cartesiano i punti `x(i),y(i)` collegandoli con una linea.

```
>> r=1:10
r =

    1  2  3  4  5  6  7  8  9 10

>> plot(r,r.^2)
```



Il seguente comando disegna un cerchio.

```
>> t=0:.001:2*pi;  
>> plot(cos(t),sin(t))
```

4 Sottomatrici e determinanti

Utilizzando l'operatore `:`, in MATLAB/Octave è possibile selezionare un'intera sottomatrice di una matrice:

```
>> A=[1 2 3; 4 5 6; 7 8 9]  
A =  
  
    1  2  3  
    4  5  6  
    7  8  9  
  
>> A(1:2,2:3)  
ans =  
  
    2  3  
    5  6  
  
>> A(2:end,1:end-2)  
ans =  
  
    4  
    7  
  
>> A(1:end,1:end)  
ans =  
  
    1  2  3  
    4  5  6  
    7  8  9  
  
>> A(1,:)   
ans =  
  
    1  2  3
```

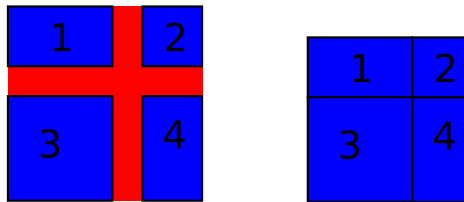
La sintassi `a:b` seleziona tutte le righe/colonne comprese tra `a` e `b` (estremi inclusi). Il valore `end` viene sostituito con il massimo indice disponibile. Il solo `:` è un'abbreviazione per `1:end`.

Possiamo anche assegnare un valore a una sottomatrice selezionata in questo modo:

```
>> A(1:2,1:2)=eye(2)
A =
  1 0 3
  0 1 6
  7 8 9
```

Ovviamente le dimensioni devono essere compatibili: non posso selezionare una sottomatrice 2×2 e assegnarle il valore `eye(3)`!

La seguente function ritorna la *matrice minore* di (i, j) in A , cioè la matrice che si ottiene eliminando la i -esima riga e la j -esima colonna di A .



```
function B=minor(A,i,j)
  X=A(1:i-1,1:j-1);
  Y=A(1:i-1,j+1:end);
  Z=A(i+1:end,1:j-1);
  W=A(i+1:end,j+1:end);
  B=[X Y; Z W];
endfunction
```

Abbiamo già visto che se X, Y, Z, W sono numeri, la riga di codice `B=[X Y; Z W]` crea la matrice

$$\begin{bmatrix} X & Y \\ Z & W \end{bmatrix};$$

ora vediamo che la stessa sintassi funziona anche se X, Y, Z, W sono matrici di dimensioni “compatibili” e crea la matrice formata accostando i quattro blocchi.

Esercizio 1. Creare una `function d=mydet(A)` che calcoli il determinante di una matrice quadrata A utilizzando la formula di Laplace sulla prima riga, cioè

$$\det(A) = A_{11} \det A^{(11)} - A_{12} \det A^{(12)} + A_{13} \det A^{(13)} - \dots + (-1)^{n+1} A_{1n} \det A^{(1n)}$$

dove A_{ij} è l’elemento di A nella posizione (i, j) e $A^{(ij)}$ è la matrice minore di A rispetto a (i, j) . Hint: la funzione può essere *ricorsiva*, cioè chiamare sé stessa al suo interno. Occhio a definire un caso base!

Poi testarla su alcune matrici, confrontandola con la funzione `det` di Octave, per esempio le matrici di Vandermonde `V=vander(1:2)`, `V=vander(1:3)`, `V=vander(1:4)`...

5 Tempi di calcolo

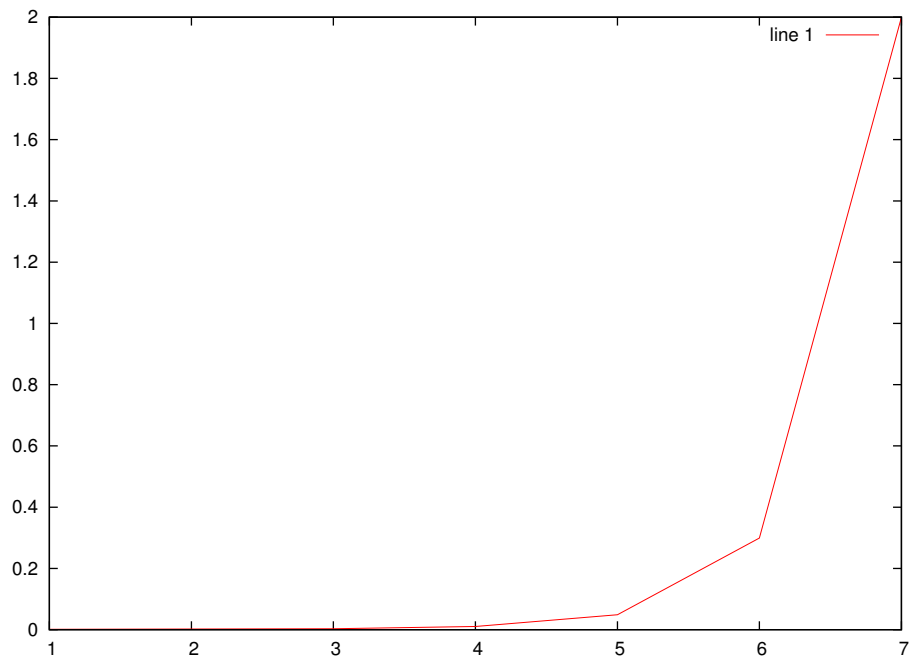
Le funzioni `tic` e `toc` misurano il tempo necessario ad eseguire più istruzioni. La prima fa partire il cronometro, la seconda lo ferma e restituisce il valore ottenuto. Per esempio, le istruzioni

```
tic;  
x=10;  
myexp2(x,500);  
t=toc;
```

salvano in `t` il tempo necessario ad eseguire le due righe centrali.

La seguente funzione disegna un grafico del tempo impiegato per calcolare i determinanti delle matrici `vander(1:n)` con `n=1:7`.

```
function plottimes();  
n=7;  
tempi=zeros(n,1); %prepara un vettore vuoto con i tempi  
for i=1:n  
    A=vander(1:i); %la matrice viene generata prima del 'tic':perche'  
    tic;  
    d=mydet(A);  
    tempi(i)=toc;  
endfor  
plot(1:n,tempi);  
endfunction
```



I tempi di calcolo crescono molto velocemente. Già per $n = 9$ la funzione è praticamente inutilizzabile. Difatti, con questo algoritmo, per calcolare un determinante di dimensione n dobbiamo calcolarne n di dimensione $n - 1$; quindi vale

$$\text{tempo}(n) \approx n \cdot \text{tempo}(n - 1)$$

da cui $\text{tempo}(n) \approx n! \cdot \text{tempo}(1)$. Il nostro algoritmo quindi non è adatto a calcolare il determinante in modo efficiente.