

METODI STATISTICI PER LA BIOINGEGNERIA

Laboratorio 1

A.A. 2025-2026

Enrico Longato



Le ore di laboratorio sono materia d'esame

L'esame prevede una prova pratica di MATLAB che peserà per circa un terzo sul voto totale dell'esame.

In pratica:

- Avrete 60 minuti per svolgere 4 esercizi MATLAB, da ~ 3 punti ciascuno, al calcolatore.
- Ciascun esercizio sarà composto da una parte di produzione del codice "da zero" e da una o più richieste di commento sul risultato ottenuto.
- Dunque, è importante sia **saper scrivere codice a partire da un file vuoto**, sia comprendere e **saper commentare quanto codificato**.



Contenuto della prova MATLAB

La prova MATLAB conterà di 4 esercizi delle seguenti tipologie:

1. Regressione lineare (con la formula risolutiva codificata esplicitamente!).
2. Applicazione di un test statistico tra quelli visti a lezione.
3. Pulizia, pre-processing o trasformazione dati.
4. Esercizio su uno degli altri argomenti del corso: survival analysis, clustering, analisi dei residui, gestione della collinearità, regressione regolarizzata.

Alcune note ulteriori

- A livello di codice, la soluzione degli esercizi sarà sempre esattamente sovrapponibile a quanto visto a lezione (le uniche, minime, varianti saranno nella pulizia dati che è difficilmente "copiaincollabile").
- All'esame saranno previsti tre tipi di risposte:
 - i. Risposta numerica secca: gli esercizi saranno ben congegnati in modo che la risposta numerica sia sempre e solo in seguito a un calcolo "atomico" e che sia corretta se e solo se il (mini)procedimento è corretto.
 - ii. Risposta discorsiva breve: lapidario commento ai risultati, tipicamente nella forma "sì/no e perché".
 - iii. Grafico su MATLAB: deve essere corretto in tutte le sue parti come da richiesta d'esame.



ISCRIZIONE CORSO SU STEM ELEARNING

<https://stem.elearning.unipd.it/course/view.php?id=13945>



N.B. accedere con modalità SSO (= "attraverso UNIWEB") con la mail @studenti.unipd.it

Nella pagina elearning del corso troverete, per ciascuna esercitazione:

- Una copia delle slide mostrate a lezione.
- I file da scaricare per svolgere l'esercitazione in aula (se necessari).
- **Dopo la lezione**, uno o più file .m con la soluzione proposta.



Laboratorio 1: Contenuti e obiettivi

1. Esercitazione "alla lavagna"

- Basi dell'ambiente di sviluppo MATLAB (v. dispense su elearning)
- Indicizzazione di matrici
- Operazioni tra vettori

2. Esercizi da svolgere in autonomia (per superare la "paura del file bianco")

- Autovettori, autovalori, inversa di matrice
- Massimo e minimo
- Generazione di sequenze (vettori equispaziati, in scala lineare, in scala logaritmica)

3. Ripasso di teoria (**parte integrante del programma d'esame di teoria!**)

- Norma di un vettore
- Autovettori e autovalori
- Inversa di matrice



Prima di svolgere l'esercitazione (oppure al bisogno), utilizzare il comando **help di MATLAB seguito dal nome delle seguenti function, utili allo svolgimento degli esercizi.**

Esercizio 1 (svolto)

- find, size, sub2ind, ind2sub

Esercizio 2 (svolto)

- norm

Esercizio 3 (proposto)

- rank, eig, inv, min, max

Esercizio 4 (proposto)

- linspace, logspace, horzcat

```
>> help find
find - Find indices and values of nonzero elements
This MATLAB function returns a vector containing the linear indices of
each nonzero element in array X.

Syntax
k = find(X)
k = find(X,n)
k = find(X,n,direction)

[row,col] = find(___)
[row,col,v] = find(___)

Input Arguments
X - Input array
    scalar | vector | matrix | multidimensional array
n - Number of nonzeros to find
    positive integer scalar
direction - Search direction
    'first' (default) | 'last'

Output Arguments
k - Indices to nonzero elements
    vector
row - Row subscripts
    vector
col - Column subscripts
    vector
v - Nonzero elements of X
    vector
```



ESERCIZIO 1 (svolto)

- Creare una matrice 3x4 con i seguenti elementi e inserirla in una variabile chiamata `mat2D`

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

- Trovare l'indice dell'elemento cerchiato (criterio "`== 2`") utilizzando **`find`**
- Tradurre l'indice lineare in formato riga/colonna utilizzando **`ind2sub`**
- Ricavare viceversa l'indice lineare a partire dalla coppia riga/colonna (**`sub2ind`**)
- Vettorizzare la matrice ottenuta, creare una nuova matrice delle dimensioni originali e inserire al suo interno gli elementi della matrice vettorizzata



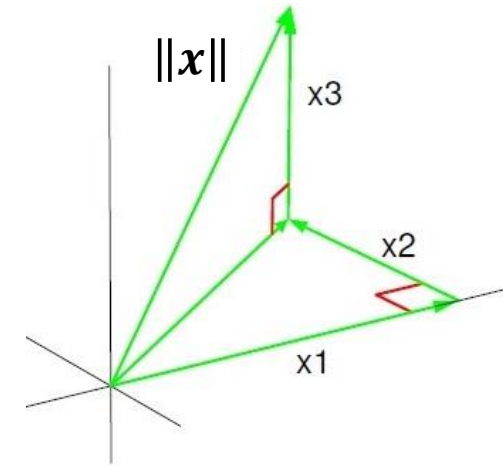
Ripasso: norme e distanze

Norma euclidea

Sia $\mathbf{x} = (x_1, x_2, \dots, x_N)$ un vettore appartenente a $\mathbb{R}^N$.

Si definisce **norma euclidea** di $\mathbf{x}$ lo scalare

$$\|\mathbf{x}\| := \sqrt{x_1^2 + x_2^2 + \dots + x_N^2}$$



Norma p

In generale, la **norma p** di un vettore $\mathbf{x} \in \mathbb{R}^N$ è uno scalare

$$\|\mathbf{x}\|_p := \sqrt[p]{|x_1|^p + |x_2|^p + \dots + |x_N|^p}$$

Notare i valori assoluti delle componenti, che servono per le norme dispari; per esempio, la norma 1 è la somma dei valori assoluti delle componenti.

Distanza euclidea

Siano $\mathbf{x}$ e $\mathbf{y}$ due vettori appartenenti a $\mathbb{R}^N$.

La distanza euclidea tra $\mathbf{x}$ e $\mathbf{y}$ è la norma euclidea della loro differenza, ovvero lo scalare

$$\text{dist}(\mathbf{x}, \mathbf{y}) := \|\mathbf{x} - \mathbf{y}\| = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_N - y_N)^2}$$



Ripasso: dimensioni del prodotto matriciale

Regole mnemoniche / di controllo per il prodotto tra matrici

1. Il prodotto tra matrici non è commutativo

$$AB \neq BA$$

2. Il prodotto tra matrici è definito (e, dunque, eseguibile) soltanto se il numero di colonne della prima matrice è uguale al numero di righe della seconda.

$$A_{3 \times 7} B_{7 \times 15} = W_{3 \times 15}$$

3. Una matrice che è il risultato di una sequenza di prodotti matriciali ha tante righe quante la prima matrice della moltiplicazione e tante colonne quante l'ultima.

$$A_{3 \times 7} B_{7 \times 15} C_{15 \times 6} D_{6 \times 4} = Z_{3 \times 4}$$



ESERCIZIO 2 (svolto)

- Creare due vettori riga (v1 e v2):

$$v1 = [4 \quad 2 \quad 6] \text{ e } v2 = [3 \quad 15 \quad 7]$$

- Calcolare $A = v1^T v2$
- Calcolare $A1 = v1 v2^T$
- Calcolare la norma euclidea del vettore v1 seguendo la formula:

$$\|x\| = \sqrt{\sum_{i=1}^n x_i^2}$$

- Calcolare la norma del vettore v1 con la funzione **norm** e verificarne i risultati rispetto al punto precedente
- Estrarre dalla matrice A una sottomatrice B composta dalle ultime 2 colonne e dalla prima e ultima riga di A

$$\begin{bmatrix} 12 & 60 & 28 \\ 6 & 30 & 14 \\ 18 & 90 & 42 \end{bmatrix}$$



Ripasso: autovettori e autovalori (1 di 2)

Definizioni

Sia $A \in \mathbb{R}^{N \times N}$ una matrice quadrata di N righe e N colonne.

Se esistono un vettore $x \in \mathbb{R}^N$ e uno scalare λ (anche complesso) tali che

$$Ax = \lambda x$$

si dice che x è **autovettore** di A e λ il suo **autovalore** corrispondente.

Polinomio caratteristico

Trovare autovettori e autovalori è equivalente a risolvere il seguente sistema lineare (basta portare tutto a sinistra nell'equazione della definizione)

$$(A - \lambda I)x = 0$$

Gli autovalori (eventualmente complessi) sono gli zeri del determinante di $A - \lambda I$.



Ripasso: autovettori e autovalori (2 di 2)

Ricerca degli autovalori a partire dal polinomio caratteristico

Il determinante di $A - \lambda I$ è il **polinomio caratteristico** di A e si scrive

$$p(\lambda) = \det(A - \lambda I)$$

Gli autovalori di A , quindi, si ricavano risolvendo l'equazione

$$p(\lambda) = \det(A - \lambda I) = 0$$

Se $A \in \mathbb{R}^{N \times N}$, $p(\lambda)$ è un polinomio di grado N in λ , ovvero la somma delle molteplicità (= quante volte annullano il polinomio) delle sue soluzioni (reali o complesse coniugate) è N .



Ripasso: invertibilità

Definizione

Una matrice $A \in \mathbb{R}^{N \times N}$ si dice singolare se il suo determinante è nullo.

Una matrice non singolare è invertibile.

Le seguenti affermazioni (e diverse altre) sono equivalenti al concetto di invertibilità.

- Il determinante di A non è nullo, cioè $\det(A) \neq 0$.
 - Conseguenza: 0 non è un autovalore di A ; altrimenti si avrebbe che $p(0) = \det(A - 0I) = \det(A) = 0$.
- Il rango di A (= numero di righe e/o colonne linearmente indipendenti) è uguale alla dimensione di A (in questo caso N)
- Esiste una matrice B tale che $AB = BA = I$; in quel caso scriviamo che $A^{-1} := B$ è la matrice inversa di A .

NB: solo le matrici quadrate possono essere o non essere invertibili. Per le matrici rettangolari si fa riferimento al concetto di pseudo-invertibilità, che riprenderemo quando parleremo di principal component analysis (PCA).



ESERCIZIO 3 (proposto, continua alla slide successiva)

- Si creino 2 matrici 4x4

$$A = \begin{bmatrix} -8 & 2 & 7 & 3 \\ -8 & -9 & -10 & 5 \\ 9 & -6 & -10 & 3 \\ 10 & -3 & -7 & -1 \end{bmatrix} \quad \text{e} \quad B = \begin{bmatrix} 2 & 5 & 6 & 1 \\ -3 & -5 & -7 & 4 \\ 6 & -1 & 0 & 2 \\ 12 & -2 & 0 & 4 \end{bmatrix}$$

- Si calcoli il rango di entrambe le matrici (**rank**)
- Si calcolino gli autovalori di entrambe le matrici (**eig**)
- Si calcoli $D = A \text{ inv}(A)$
- Si verifichi la differenza di utilizzare l'operatore / o \ per il calcolo dell'inversa:
 $D = A \text{ inv}(A)$ è equivalente a $D = A/A$?
- Si calcoli $E = \text{inv}(A) * B$ con la dicitura standard e l'operatore \ e se ne verifichi l'equivalenza
- Si calcoli il prodotto elemento per elemento $F = A.*B$
- Verificare che è diverso dal prodotto matriciale $A*B$



ESERCIZIO 3 (proposto, continuazione)

- Trovare sia l'indice che le coppie riga/colonna di ogni elemento negativo della matrice F appena calcolata
- Cercare il massimo e il minimo della matrice A (suggerimento, vettorizzare la matrice) e trovarne l'indice lineare e le coppie/riga colonna
- Provare a ripetere l'operazione precedente non vettorizzando la matrice

ESERCIZIO 4 (proposto)

- Generare i seguenti vettori discreti
 - Intervallo j1 [0,50] con passo 1 (si usi l'operatore :)
 - Intervallo j2 [0,50] con passo 0.5 (si usi l'operatore :)
 - Intervallo j3 [501,1000] con passo 0.5, utilizzando la funzione **linspace**
 - Concatenare per righe j1 e j3 facendo uso delle parentesi quadre o della funzione **horzcat**, nominando il nuovi intervallo "spezzato" j4
 - Intervallo j5 [1,100] con 10 punti e spaziatura logaritmica (**logspace**)