Approssimazione nel senso dei minimi quadrati

Problema 1

Nella seguente tabella sono riportati i tempi di CPU (in sec) rilevati, necessari all'ordinamento di un vettore di lunghezza n.

n	10000	20000	30000	40000	50000
CPU(sec)	0.31	0.95	2.45	4.10	6.46

Determinare la parabola che meglio approssima i dati nel senso dei minimi quadrati e stimare quanto tempo di CPU serve per ordinare 1000 vettori, ognuno di 55218 elementi.

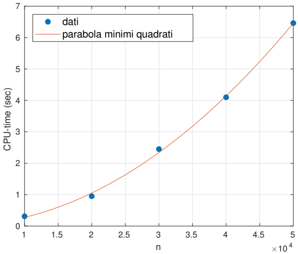
Svolgimento

La soluzione nel senso dei minimi quadrati è la parabola $p_2(n) = a_1 n^2 + a_2 n + a_3$ che minimizza la funzione (funzione obiettivo)

$$\Phi(\mathbf{a}) = \sum_{i=1}^{5} (\underbrace{a_1 n_i^2 + a_2 n_i + a_3}_{p_2(n_i)} - T_i)^2.$$

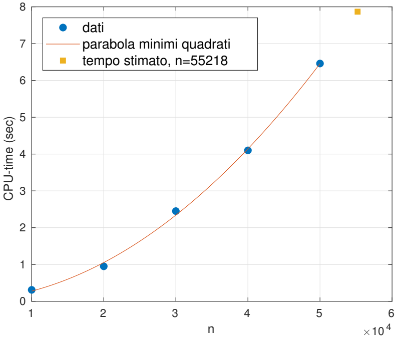
Le incognite sono i coefficienti $a_1,\ a_2,\ a_3$,

```
n=[... ];
T=[... ];
a=polyfit(n,T,2);
n1=linspace(n(1),n(end),100))';
T1=polyval(a,n1);
plot(n,T,'o',n1,T1);
grid on
```



Per stimare il tempo di CPU necessario per ordinare un vettore di $n = 55218$ elementi:

```
ns=55218;
ys=polyval(a,ns)
```



Quindi la risposta finale sarà

```
tempo=1000*ys
```

Minimi quadrati per il Machine Learning

Il metodo dei minimi quadrati è largamente utilizzato come algoritmo per il Machine Learning (apprendimento automatico), quando si hanno

- problemi di tipo *supervised learning*,
- in cui l'output (cioè il valore da stimare) è un numero reale
- e si vuole sostenere un basso costo computazionale.

Dati gli insiemi $X = \{x_1, x_2, \ldots, x_m\}$ e $Y = \{y_1, y_2, \ldots, y_m\}$
(training set= dati che devono insegnare all'algoritmo)
si cerca un modello (una funzione) $f : X \to Y$ che esprima la correlazione tra i due insiemi di dati:

$$y_i = f(x_i) + e_i, \qquad i = 1, \ldots, m$$

in modo da poter predire l'ouput (incognito) $\hat{y} = f(\hat{x})$ in corrispondenza di un nuovo dato $\hat{x}$.

Un problema con dati 2d

Si vuole costruire un modello lineare (un piano) per stimare il prezzo al metro quadrato delle abitazioni di una zona circostante alla città, conoscendone l'età e la distanza dal centro città.
Si conoscono i seguenti dati:

X		Y
età (anni)	distanza (km)	prezzo (euro/m^2)
10	3	2300
5	9	2000
30	4	1800
25	10	1700
45	6	1550
6	12	1800
15	7	2200
35	16	1450

Stimare il prezzo di una abitazione di 23 anni distante 5 km dal centro città.

Svolgimento

Poniamo x_1 =età, x_2 =distanza, y =prezzo.
Cerchiamo un piano

$$p(x_1, x_2) = a_1 x_1 + a_2 x_2 + a_3 : \qquad p((x_1)_i, (x_2)_i) \sim y_i, \quad i = 1, \ldots, m$$

nel senso dei minimi quadrati.
Cioè cerchiamo

$$\mathbf{a}^* = \underset{\mathbf{a}=[a_1,a_2,a_3]'}{\text{argmin}} \; \Phi(\mathbf{a})$$

con

$$\Phi(\mathbf{a}) = \sum_{i=1}^{m} (\underbrace{a_1 (x_1)_i + a_2 (x_2)_i + a_3}_{p((x_1)_i,(x_2)_i)} - y_i)^2.$$

`polyfit` funziona solo per problemi in 1d, mentre qui i dati X sono 2d.
Allora ricorriamo all'interpretazione algebrica del problema di minimo:

costruiamo la matrice X ed il vettore Y:

$$X = \begin{bmatrix} (x_1)_1 & (x_2)_1 & 1 \\ \vdots & \vdots & \vdots \\ (x_1)_m & (x_2)_m & 1 \end{bmatrix}, \qquad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}.$$

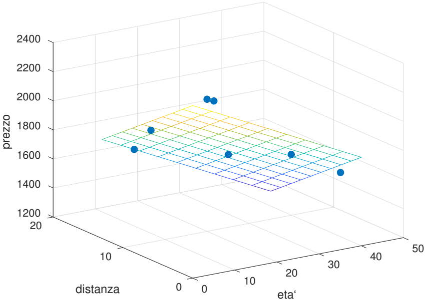
$$\mathbf{a}^* = \underset{\mathbf{a}=[a_1,a_2,a_3]'}{\operatorname{argmin}} \quad \Phi(\mathbf{a}) \quad \Leftrightarrow \quad X^T X \mathbf{a}^* = X^T Y.$$

Comando matlab:

```
a=X\Y
```

Calcoliamo $\mathbf{a}^*$:

```
eta=   [     ];
dist=  [     ];
prezzo=[     ];
N=length(eta);
X=[eta,dist,ones(N,1)];
Y=prezzo;
as=X\Y;
```



e disegnamo:

```
figure(1);clf
p1=plot3(eta,dist,prezzo,'o'); hold on
x1v=linspace(min(eta),max(eta),10);
x2v=linspace(min(dist),max(dist),10);
[x1,x2]=meshgrid(x1v,x2v);
z=as(1)*x1+as(2)*x2+as(3);
mesh(x1,x2,z); grid on
xlabel('eta`'); ylabel('distanza'); zlabel('prezzo')
xs=23; ys=5; ps=as(1)*xs+as(2)*ys+as(3)
```