

Approssimazione polinomiale ¹

A. Sommariva ²

Keywords: Approssimazione polinomiale ai minimi quadrati. Curve fitting.

Revisione: 4 giugno 2019

1. Approssimazione polinomiale

Dato un campionamento $\{(x_i, y_i)\}$, $y_i = f(x_i)$, $i = 1, 2, \dots, N$, dove $x_i \neq x_j$ se $i \neq j$, nell'approssimazione ai minimi quadrati invece di interpolare si cerca un polinomio \mathcal{L}_m di grado $m \leq N$, in cui tipicamente $m \ll N$, tale che la somma degli scarti quadratici

$$\sqrt{\sum_{i=1}^N (y_i - \mathcal{L}_m(x_i))^2}$$

è minima, ovvero che risolva il problema di ottimizzazione in $m + 1$ variabili $\mathbf{a} = (a_0, \dots, a_m)$

$$\min_{\alpha \in \mathbb{R}^{m+1}} \sum_{i=1}^N (y_i - (a_0 + a_1 x_i + \dots + a_m x_i^m))^2. \quad (1)$$

Si può dimostrare che tale problema ha una e una sola soluzione \mathbf{a} , ovvero esiste un unico polinomio \mathcal{L}_m minimizzante la somma degli scarti quadratici (1).

Nota 1.1 (Richiede la conoscenza di algebra lineare). *La dimostrazione usa elementi di algebra lineare. Ne forniamo una traccia*

Se

$$V = (v_{i,j})_{i=1, \dots, N, j=0, \dots, m} = (x_i^j)$$

è una matrice di Vandermonde rettangolare di dimensione $N \times (m+1)$, allora si mostra che $V^T V$ è una matrice di dimensione $(m+1) \times (m+1)$, simmetrica e definita positiva (e quindi non singolare).

Inoltre si vede che $\mathbf{a} = (a_i)$ è soluzione ai minimi quadrati se e solo se risolve il sistema $V^T V \mathbf{a} = V^T \mathbf{y}$ per $\mathbf{y} = (y_i)$, ed essendo $V^T V$ non singolare, necessariamente la soluzione \mathbf{a} è unica.

Teorema 1.2. *Se*

- $h = \max_i \Delta x_i \leq \theta(b - a)/m^2$,

- $\theta \in (0, 1)$,

allora per ogni $f \in C^k([a, b])$, $k > 1$, esiste una costante c_k tale che

$$\max_{x \in [a, b]} |\mathcal{L}_m(x) - f(x)| \leq c_k m^{1-k}.$$

Commento 1.3. Questo teorema asserisce che la soluzione ai minimi quadrati \mathcal{L}_m di grado m , su un set di nodi sufficientemente fitti, darà uniformemente una buona approssimazione della funzione f da approssimare, tendenzialmente migliore se la funzione è molto regolare.

Non dice molto del valore della costante c_k , e quindi il termine $c_k m^{1-k}$ non è direttamente valutabile.

Esempio. Approssimare ai minimi quadrati le funzioni

1. $f_1(x) = x^{30} \in C^\infty([0, 1])$,
2. $f_2(x) = x^{7/2} \in C^3([0, 1])$,
3. $f_3(x) = \exp(x) \in C^\infty([0, 1])$.

in opportuni nodi equispaziati.

Come noto dal teorema precedente, possiamo aspettarci che se $x_k = a + kh$, $k = 1, \dots, n$ con h sufficientemente piccolo, allora il massimo errore

$$\max_{x \in [a, b]} |\mathcal{L}_m(x) - f_k(x)|, \quad k = 1, 2, 3$$

tenderà a 0 al crescere di m .

Al variare di m , sceglieremo $h = 0.5(b - a)/m^2$. Se n è il numero di subintervalli di $[a, b]$, visto che $nh = (b - a)$, otteniamo

$$n = (b - a)/h = 2m^2.$$

Il calcolo della soluzione ai minimi quadrati viene eseguito utilizzando la funzione Matlab `polyfit`, e per avere una stima dell'errore commesso

$$E_m(f_k) = \max_{x \in [a, b]} |\mathcal{L}_m(x) - f_k(x)|$$

valutiamo

$$E_m^*(f_k) = \max_{x \in \Delta_{10000}} |\mathcal{L}_m(x) - f(x)| \approx E_m(f_k),$$

dove $\Delta_{10000} = \{x_k = -5 + k/1000, \quad k = 0, \dots, 10000\}$.

I risultati sono rappresentati in tabella.

m	$E_m^*(f_1)$	$E_m^*(f_2)$	$E_m^*(f_3)$	$n + 1$
1	$6.9e - 01$	$3.0e - 01$	$1.4e - 01$	3
2	$4.0e - 01$	$5.2e - 02$	$9.6e - 03$	9
3	$3.4e - 01$	$4.3e - 03$	$7.3e - 04$	19
4	$2.8e - 01$	$3.2e - 04$	$4.1e - 05$	33
5	$2.1e - 01$	$6.2e - 05$	$1.9e - 06$	51
6	$1.4e - 01$	$1.7e - 05$	$7.3e - 08$	73
7	$8.6e - 02$	$6.1e - 06$	$2.5e - 09$	99
8	$4.9e - 02$	$2.5e - 06$	$7.2e - 11$	129
9	$2.6e - 02$	$1.1e - 06$	$1.9e - 12$	163
10	$1.3e - 02$	$5.6e - 07$	$4.5e - 14$	201
15	$1.1e - 04$	$3.7e - 08$	$2.0e - 15$	451
20	$9.4e - 08$	$5.4e - 09$	$1.8e - 15$	801

Tabella 1: Nella tabella indichiamo il grado $m = 1, \dots, 10, 15, 20$, gli errori compiuti dalle approssimazioni ai minimi quadrati $E_m^*(f_1)$, $E_m^*(f_2)$, $E_m^*(f_3)$ e il numero di punti di campionamento.

2. Minimi quadrati e curve fitting

Nota 2.1. Nelle scienze applicate capita frequentemente che gli esperimenti producano set di dati $\{(x_i, y_i)\}_{i=1, \dots, N}$ dove le ascisse sono distinte. Uno degli scopi è di determinare una formula del tipo $y = f(x)$ che relazioni le variabili.

Usualmente è a disposizione una classe di funzioni \mathcal{F} e bisogna determinare f all'interno di questa famiglia. Tale problema è noto come *curve fitting*.

In questa sezione consideriamo il caso in cui $\mathcal{F} = \mathbb{P}_m$, ovvero i polinomi di grado al più m e determiniamo f utilizzando la soluzione ai minimi quadrati.

Nota 2.2. Inoltre i dati y_i sono spesso inesatti, dovuti agli errori sperimentali compiuti. Questa tecnica permette in qualche senso di evidenziare una curva che meglio li approssima

2.1. Regressione lineare

Definizione 2.3 (Regressione lineare). Se $\mathcal{F} = \mathbb{P}_1$, ovvero calcoliamo la soluzione ai minimi quadrati \mathcal{L}_1 , si parla di *regressione lineare*.

Esempio. Si considerino i dati della seguente tabella (cf. [1, p.79])

x_i	1	3	4	6	7
y_i	-2.1	-0.9	-0.6	0.6	0.9

Si disegni il grafico della soluzione al problema di regressione lineare.

I risultati sono evidenziati in figura, dove la retta di regressione è quella che meglio approssima i dati nel senso dei minimi quadrati.

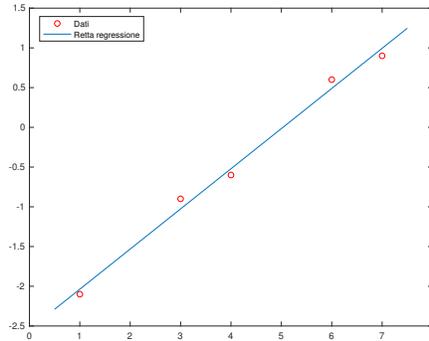


Figura 1: Dati e retta di regressione lineare.

2.2. Approssimazione polinomiale e dati inesatti

In alcuni problemi, i dati sono soggetti a rumore e uno vuole in qualche senso togliere il rumore.

Esempio. Consideriamo quali dati $(x_i, y_i)_{i=1, \dots, 40}$ in cui $x_i = -1 + i \cdot \frac{1}{20}$ e $y_i = \sin(10 \cdot x_i) + 10^3 \mu_i$ dove μ_i è un numero random in $[-0.5, 0.5]$. Determinare il polinomio di miglior approssimazione.

In qualche modo simuliamo dei dati $y_i^* = \sin(x_i)$ cui abbiamo aggiunto il rumore $10^3 \mu_i$.

- L'intento è approssimare la funzione $f^*(x) = \sin(10x)$ togliendo il rumore.
- Non ha senso calcolare l'interpolante polinomiale, perché si riprodurrebbero i rumori e non la funzione cercata.

In figura, mostriamo i risultati ottenuti utilizzando le approssimanti \mathcal{L}_m ai minimi quadrati di grado rispettivamente $m = 5, m = 10, m = 15, m = 30$.

In virtù dei 6 cambi di convessità / concavità della funzione f^* , è chiaro che un polinomio di grado basso non fornisca risultati apprezzabili.

Ad esempio un polinomio di grado 5 può avere al più 4 cambi di convessità / concavità, e non può essere adatto allo scopo.

Un polinomio \mathcal{L}_m di grado m troppo alto approssimerebbe la funzione con rumore e non f^* .

Nel nostro esempio, $m = 15$ sembra essere un buon compromesso. Si noti come a grado $m = 30$ comincino a sorgere oscillazioni agli estremi. Inoltre per problemi dovuti a `polyfit`, il risultato non è quello corretto teoricamente.

La tabella che segue, mostra gli errori

$$E_m(f^*) = \max_{x \in [-1, 1]} |f^*(x) - \mathcal{L}_m(x)|, \quad E_m(f) = \sum_{i=1}^{40} |f(x_i) - \mathcal{L}_m(x_i)|^2,$$

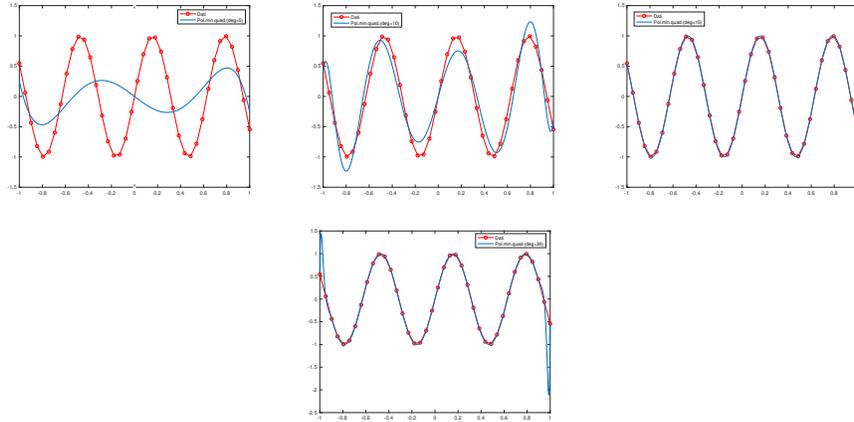


Figura 2: Dati (in rosso) e approssimanti ai minimi quadrati di grado $m = 5, m = 10, m = 15, m = 30$ (in blu).

evidenziando un miglioramento di entrambe fino a grado $m = 20$ e un peggioramento nell'approssimare la funzione senza rumore f^* pur approssimando sempre meglio i dati (nel senso dei minimi quadrati).

m	$E_m(f^*)$	$E_m(f)$
5	$1.2e + 00$	$4.0e + 00$
10	$3.3e - 01$	$1.2e + 00$
15	$4.0e - 03$	$5.1e - 03$
20	$5.7e - 03$	$2.3e - 03$
25	$3.6e - 02$	$2.2e - 03$
30	$1.7e + 00$	$1.8e - 03$

Tabella 2: Grado m ed errori $E_m(f^*) = \max_{x \in [-1,1]} |f^*(x) - \mathcal{L}_m(x)|$, $E_m(f) = \sum_{i=1}^{40} |y_i - \mathcal{L}_m(x_i)|^2$.

3. Bibliografia

- [1] A. Quarteroni e F. Saleri, Elementi di calcolo numerico, Progetto Leonardo, 1999.