Regressione Lineare

Guglielmo Bartelloni

31 marzo 2022

30 Marzo 2022

Possiamo avere rumore nei dati, ovvero i dati possono venire da misure sperimentali affette da errore.

1 Approssimazione Polinomiale nel senso dei minimi quadrati

Supponiamo di avere un campione di dati, (x_i, f_i), per $i = 0, ..., N$ in cui $f_i = \sum_{i=0}^{m} a_k x_i^k$ con $m << N$ (Abbiamo pochi dati che rispettano le condizioni di interpolazione).

(Assumiamo che almeno m+1 delle ascisse x_i siano tra loro distinte)

Quello che dobbiamo fare e' quindi quello di calcolare i coefficienti $a_0, ..., a_m$ che meglio approssimano i dati del problema.

Un criterio per questo, consiste nel chiamare $y_i = \sum_{k=0}^{m} a_k x_i^k$ il valore **previsto** dal modello polinomiale e, definendo i 2 vettori:

$$f = [f_0, f_1, ..., f_N]^T.$$

$$Y = [y_0, y_1, ..., y_N]^T.$$

Minimizzare la norma della loro differenza:

$$||f - Y||.$$

Questa norma, per quello che andremo a vedere, e' scelta come norma 2. Il motivo di questo deriva dal fatto che:

Y e' una matrice con in ogni riga

Immagine appunti

Osservazione 1. Se almeno m+1 delle ascisse x_i sono distinte tra loro, le corrispondenti righe definiscono una matrice di Vandermonde non singolare.

Pertanto, $rank(V) = m + 1$, ovvero la matrice $V \in R^{N+1xm+1}$ ha **rango massimo**.

Quindi desideriamo calcolare il vettore a come soluzione del sistema lineare sovradeterminato:

$$Va = f \ (**).$$

Pertanto, se in (*) utilizziamo la norma 2, questo equivale a risolvere (**) nel senso dei minimi quadrati, mediante la fattorizzazione QR della matrice V.

I coefficienti, che sono le componenti del vettore a, definiscono il **polinomio di approssimazione ai minimi quadrati** dei dati (x_i, f_i) per $i = 0, ..., N$

Osservazione 2. Se m=N si ottiene il polinomio interpolante, in quanto si soddisfano esattamente le condizioni:

$$\sum_{k=0}^{m} a_k x_i^k = f_i \quad i = 0, ..., m.$$

La function **polyfit** di Matlab implementa l'approssimazione polinomiale ai minimi quadrati.

Syntax

```
p = polyfit(x,y,n)
[p,S] = polyfit(x,y,n)
[p,S,mu] = polyfit(x,y,n)
```

Description

`p = polyfit(x,y,n)` returns the coefficients for a polynomial `p(x)` of degree n that is a best fit (in a least-squares sense) for the data in y. The coefficients in p are in descending powers, and the length of p is n+1 example

$$p(x) = p_1 x^n + p_2 x^{n-1} + ... + p_n x + p_{n+1}.$$

`[p,S] = polyfit(x,y,n)` also returns a structure S that can be used as an input to `polyval` to obtain error estimates.

`[p,S,mu] = polyfit(x,y,n)` also returns mu, which is a two-element vector with centering and scaling values. `mu(1)` is `mean(x)`, and `mu(2)` is `std(x)`. Using these values, `polyfit` centers x at zero and scales it to have unit standard deviation, example

$$\hat{x} = \frac{x - \bar{x}}{\sigma_x}.$$

This centering and scaling transformation improves the numerical properties of both the polynomial and the fitting algorithm.