

Approssimazione di dati

Lucia Gastaldi

DICATAM - Sez. di Matematica,
<http://lucia-gastaldi.unibs.it>

Indice

- 1 Approssimazione di dati
 - Approssimazione di dati

- 2 Minimi quadrati lineari
 - Regressione lineare
 - Minimi quadrati lineari: caso generale

Legge di Ohm

La **legge di Ohm** afferma che la differenza di potenziale V ai capi di un conduttore elettrico e l'intensità di corrente elettrica I che lo attraversa sono legati dalla relazione

$$V = R \cdot I$$

essendo R la resistenza elettrica del conduttore.

Supponiamo di avere un apparecchio per misurare la differenza di potenziale agli estremi di un filo conduttore e un amperometro per misurare l'intensità di corrente corrispondente.

I risultati sperimentali (contenuti nel file `datiOhm.m`) sono:

$I = [-1, 6, 13, 16, 17, 25, 28, 29, 33, 36, 39, 43, 44]$;

$V = [0, 1, 1.3, 1.6, 1.9, 2.3, 2.6, 2.9, 3.2, 3.5, 3.9, 4.2, 4.5]$;

$V = V / 4.5 * 7$;

Si vuole rappresentare la legge che regola la relazione tra I e V .

Legge di Ohm (segue)

- Rappresentare i dati sperimentali marcandoli con un pallino.
- Interpolare i dati sperimentali usando un polinomio interpolatore di grado opportuno.
- Interpolare i dati usando le spline.
- Trovare la retta di regressione lineare.

Approssimazione nel senso dei minimi quadrati lineari

Siano dati (x_i, y_i) per $i = 0, \dots, n$ con n abbastanza grande.

L'interpolazione in questo caso non dà risultati significativi.

Cerchiamo un'approssimazione della nostra quantità fisica nella forma

$$\sum_{j=0}^m a_j \varphi_j(x) \quad \text{essendo } \varphi_j \text{ funzioni assegnate.}$$

Imponendo il passaggio per tutti i punti si ottiene il sistema

$$\sum_{j=0}^m a_j \varphi_j(x_i) = y_i, \quad \text{per } i = 0, \dots, n$$

di $n + 1$ equazioni in $m + 1$ incognite con $m \ll n$, che difficilmente ha soluzione.

Per $i = 0, \dots, n$ e $j = 0, \dots, m$, poniamo $A_{ij} = \varphi_j(x_i)$, $b_i = y_i$ e $x_j = a_j$;
il sistema $Ax = b$ è sovradeterminato.

Minimi quadrati lineari

Minimi quadrati lineari

Trovare x che renda minima la seguente quantità:

$$\min_{x \in \mathbb{R}^{m+1}} \|Ax - b\|^2 = \min_{x \in \mathbb{R}^{m+1}} \sum_{i=0}^n \left(\left(\sum_{j=0}^n A_{ij} x_j \right) - b_i \right)^2 .$$

Regressione lineare I

Dati i punti (x_i, y_i) per $i = 0, \dots, n$, vogliamo calcolare la retta $y = a_0x + a_1$ che meglio li approssima.

Il sistema diventa: $a_0x_i + a_1 = y_i$ per $i = 0, \dots, n$ e abbiamo

$$A = \begin{pmatrix} x_0 & 1 \\ x_1 & 1 \\ \dots & \dots \\ x_n & 1 \end{pmatrix} \quad y = \begin{pmatrix} y_0 \\ y_1 \\ \dots \\ y_n \end{pmatrix} \quad x = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix}$$

Si deve cercare il punto di minimo della funzione:

$$\Phi(a_0, a_1) = \sum_{i=0}^n (a_0x_i + a_1 - y_i)^2$$

Annullando le derivate parziali si ottiene:

$$\frac{\partial \Phi}{\partial a_0}(a_0, a_1) = \sum_{i=0}^n 2(a_0x_i + a_1 - y_i)x_i = 0$$

$$\frac{\partial \Phi}{\partial a_1}(a_0, a_1) = \sum_{i=0}^n 2(a_0x_i + a_1 - y_i) = 0$$

Si risolve quindi il sistema

$$\begin{pmatrix} \sum_{i=0}^n x_i^2 & \sum_{i=0}^n x_i \\ \sum_{i=0}^n x_i & n+1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^n x_i y_i \\ \sum_{i=0}^n y_i \end{pmatrix}$$

ossia

$$A^T A \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = A^T y$$

Minimi quadrati lineari: caso generale

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|^2$$

Si ha:

$$\begin{aligned}\|Ax - b\|^2 &= (Ax - b)^T (Ax - b) \\ &= x^T A^T Ax - x^T A^T b - b^T Ax + b^T b \\ &= x^T A^T Ax - 2b^T Ax + b^T b\end{aligned}$$

Per trovare il minimo si annulla il gradiente, ottenendo il sistema lineare

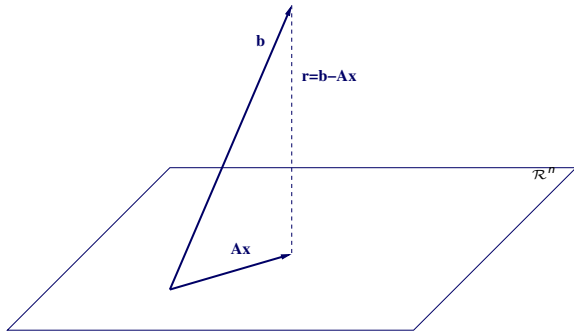
$$A^T Ax = A^T b$$

detto **sistema delle equazioni normali**.

Sistema delle equazioni normali

$$A^T A x = A^T b$$

- La matrice $A^T A$ è una matrice quadrata $n \times n$.
- La matrice $A^T A$ è **simmetrica** e **definita positiva** se A è a rango pieno. In questo caso la soluzione è unica.
- La matrice $A^+ = (A^T A)^{-1} A^T$ è detta **matrice pseudo-inversa** di A .



Minimi quadrati lineari in Matlab

Per calcolare il polinomio di grado m che approssima i punti (x_i, y_i) per $i = 0, \dots, n$ nel senso dei minimi quadrati lineari si usa la function `polyfit` con il comando

```
p=polyfit(x,y,m)
```

Identificazione di parametri

Il file `dati.m`, contiene le *osservate* y_i di una certa quantità, prese per valori di t equispaziati.

- Trovare la retta di regressione lineare e plottare in una stesso grafico la retta ed i punti assegnati.
- In una seconda figura plottare i residui $y(t_i) - y_i$. Osservare che c'è un valore dei residui molto più grande degli altri. Chiamiamo il valore corrispondente *valore anomalo*.
- Eliminare l'osservata corrispondente al valore del residuo massimo. Ricalcolare la retta di regressione lineare e ripetere i grafici del caso precedente.
- Osservare l'andamento del residuo.
- Fittare i dati (escluso il valore anomalo) con un modello della forma seguente:

$$M(t) = \beta_1 + \beta_2 t + \beta_3 \sin t.$$

- Plottare il modello con una linea continua, insieme ai dati (marcati con o) e al valore anomalo (marcato con *).

find

Il comando `find` serve per trovare le componenti di un vettore diverse da zero.

Esempio

Posto `x=[0 1 2 3 0 2 1]`, il comando `I=find(x)` fornisce il risultato `I=[2 3 4 6 7]`.

Ricordando che una variabile logica è vera se ha valore diverso da zero, il comando `J=find(x==1)` fornisce gli indici delle componenti che sono uguali a uno, quindi `J=[2 7]`.

Eliminazione di una componente

- Per eliminare la componente di posto `i` dal un vettore **riga** `v`:
`v1=[v(1:i-1),v(i+1:end)]`.
- Per eliminare la componente di posto `i` dal un vettore **colonna** `w`: `w1=[w(1:i-1);w(i+1:end)]`.

Costruzione di una matrice per colonne

Per fittare i dati con il modello $M(t) = \beta_1 + \beta_2 t + \beta_3 \sin(t)$, si deve calcolare il valore di $M(t_i)$ per ogni i e poi si cerca $\underline{\beta} = (\beta_1, \beta_2, \beta_3)$ in modo tale che $\underline{\beta}$ fornisca il valore minimo di

$$\sum_{i=1}^n (M(t_i) - y_i)^2 = \sum_{i=1}^n (\beta_1 + \beta_2 t_i + \beta_3 \sin(t_i) - y_i)^2.$$

Questo è equivalente a risolvere il sistema sovradeterminato $A\underline{\beta} = \underline{y}$ essendo

$$A = \begin{bmatrix} 1 & t_1 & \sin(t_1) \\ 1 & t_2 & \sin(t_2) \\ \dots & \dots & \dots \\ 1 & t_n & \sin(t_n) \end{bmatrix} \quad \underline{y} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix}$$

Se $\mathbf{t}$ è il vettore colonna che contiene i valori della variabile t , la matrice può essere costruita per colonne con il comando

```
A=[ones(size(t)),t,sin(t)]
```