

Minimi quadrati in Matlab

Alvise Sommariva

Università degli Studi di Padova
Dipartimento di Matematica Pura e Applicata

17 gennaio 2019

Approssimazione ai minimi quadrati (discreti)

Si digiti sulla command-window di Matlab

```
>> x=0:0.01:2*pi ;  
>> y=sin(2*x)+(10^(-1))*rand(size(x));  
>> plot(x,y,'r-');
```

- Dal grafico si capisce che la funzione può essere interpretata come una **perturbazione** della funzione $\sin(2x)$ nell'intervallo $[0, 2\pi]$.
- Ci interessa approssimare non tanto la funzione plottata bensì $\sin(2x)$ (che in qualche modo è la funzione senza **rumore**).
- Osserviamo che non ha senso utilizzare un interpolante polinomiale p di grado N nè una spline interpolante visto che **ricostruirebbero** la funzione **perturbata**.

Approssimazione ai minimi quadrati (discreti)

Scriviamo sulla command-window di Matlab/Octave

```
help polyfit
```

In una recente release di Matlab abbiamo

```
POLYFIT Fit polynomial to data.  
POLYFIT(X,Y,N) finds the coefficients of a polynomial P(X)  
of degree N that fits the data, P(X(I))~=Y(I), in a least  
-squares sense.
```

L'help dice che `polyfit` calcola i coefficienti del polinomio di grado N che **meglio** approssima i dati $(X(k), Y(k))$, $k = 1, \dots, M$ nel senso dei **minimi quadrati**, ovvero determina l'unico polinomio $p_N \in \mathbb{P}_N$ tale che sia minima la quantità

$$\|f - P_N\|_2 = \sqrt{\sum_{i=1}^M |f(x_i) - P_N(x_i)|^2}.$$

Approssimazione ai minimi quadrati (discreti)

Nota. (Norme e norm)

Ricordiamo che se $u \in \mathbb{R}^M$ allora

$$\|u\|_2 := \sqrt{\sum_{i=1}^M u^2}$$

In Matlab, per calcolare tale quantità si usa il comando

`norm(u,2)`.

```
>> u=[1 2 3]
u =
     1     2     3
>> norm(u,2)
ans =
     3.7417
>> sqrt(1^2+2^2+3^2)
ans =
     3.7417
>>
```

Regressione lineare

Il problema di determinare il polinomio di grado 1 che **meglio** approssima i dati $(X(k), Y(k))$, $k = 1, \dots, M$ nel senso dei **minimi quadrati**, viene usualmente detto **regressione lineare**.

Vediamone un esempio in Matlab.

```
function demo_regressione_lineare

% Esempio regressione lineare.
a=0; b=1;
h=0.05; x=a:h:b; % ascisse equispaziate
y=0.1+x+(10^(-1))*rand(size(x)); % ordinate

% valutazione coeff. dell'approssimante ai minimi quadrati
% "p_n" di "f" di grado "1"
coeff=polyfit(x,y,1);

% valore "p_1" nelle ascisse "x"
z=polyval(coeff,x);

% errore ||f-p_1||_2
err2=norm(z-y,2);
fprintf('\n \t Errore regressione norma2: %1.2e',err2);

% grafico del polinomio ai minimi quadrati di grado "1"
ht=1/10000; u=a:ht:b;
v=polyval(coeff,u);
```

Regressione lineare

```
clf;
% plot punti
plot(x,y,'go','LineWidth',1,...
      'MarkerEdgeColor','k',...
      'MarkerFaceColor','g',...
      'MarkerSize',10);
hold on;
% plot retta regressione
plot(u,v,'k-','LineWidth',2);
% titoli e legenda
title('Regressione lineare');
legend('Dati','Retta di regressione');
hold off;

fprintf('\n \n');
```

Regressione lineare

Tale routine,

- definisce una funzione che corrisponde a una **perturbazione** della retta $r(x) = 0.1 + x$;

- determina mediante il comando

```
coeff=polyfit(x,y,1);
```

i coefficienti del polinomio p_1 di grado 1 che **meglio** approssima i dati $(x(k), y(k))$, $k = 1, \dots, 21 = (1/h) + 1$ nel senso dei **minimi quadrati**;

- valuta $\|f - p_1\|_2$;
- disegna in una stessa figura il grafico dei dati $(x(k), y(k))$, $k = 1, \dots, 21$ e del polinomio p_n , valutato mediante il comando

```
v=polyval(coeff,u);
```

nelle ascisse di test u ; si osservi che i punti vengono rappresentati mediante cerchietti, utilizzando varie preferenze di colore e grandezza;

- inserito il titolo **regressione lineare**, e una legenda.

Lanciato da command-window abbiamo

```
>> demo_regressione_lineare
    Errore regressione norma2: 1.29e-01
>>
```

e la figura che segue.

Regressione lineare

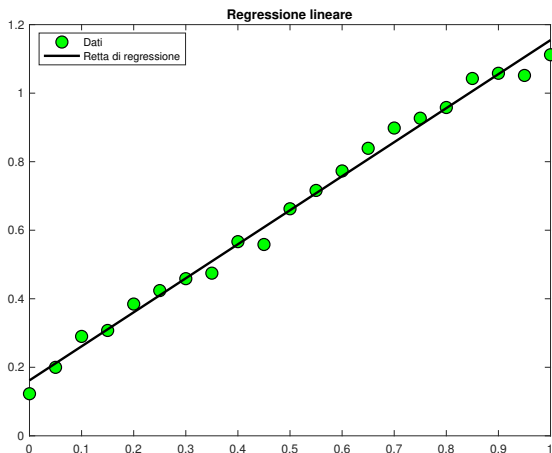


Figura: Grafico che illustra l'approssimazione ai minimi quadrati di grado 1 su una perturbazione della retta $0.1 + x$ (campionamento in nodi equispaziati).

La figura mostra che la retta soluzione del problema non interpola i dati, ma in generale li **approssima** tutti relativamente bene.

Nota.

Vista la presenza di numeri casuali, ogni esempio che viene compiuto mediante tale routine è potenzialmente diverso.

Un altro esempio

Digitiamo quindi in un file `demo_minimiquadrati.m`

```
function demo_minimiquadrati
a=0; b=2*pi;
h=0.01; x=a:h:b; % ascisse equispaziate
y=sin(2*x)+(10^(-1))*rand(size(x)); % ordinate
for n=0:8
    % valutazione coeff. dell'approssimante ai minimi quadrati
    % "p_n" di "f" di grado "n"
    coeff=polyfit(x,y,n);
    % valore "p_n" nelle ascisse "x"
    z=polyval(coeff,x);
    % errore ||f-p_n||_2
    err2=norm(z-y,2);
    fprintf('\n \t n: %2.0f norma2: %1.2e',n,err2);
    % grafico del polinomio ai minimi quadrati di grado "n"
    ht=1/10000; u=a:ht:b;
    v=polyval(coeff,u);
    clf;
    plot(x,y,'r. '); hold on;
    plot(u,v,'k-', 'LineWidth',2);
    titlestr=strcat('Minimi quadrati di grado:',num2str(n));
    title(titlestr);
    legend('Dati','Approssimante Minimi quadrati');
    hold off;
    % pausa di 3 secondi tra un grafico e il successivo
    pause(3);
end
fprintf('\n \n');
```

Un altro esempio

Tale routine, dapprima definisce una funzione che corrisponde a una **perturbazione** di $\sin(x)$ in $[0, 2\pi]$.

Di seguito per n che varia da 1 a 8,

- determina i coefficienti del polinomio p_n di grado n che **meglio** approssima i dati $(x(k), y(k))$, $k = 1, \dots, 101 = (1/h) + 1$ nel senso dei **minimi quadrati**;
- valuta $\|f - p_n\|_2$;
- disegna in una stessa figura il grafico dei dati $(x(k), y(k))$, $k = 1, \dots, 101$ e del polinomio p_n , cambiando il titolo al variare del grado, e inserendo una legenda;
- mette in pausa per 5 secondi il processo.

Un altro esempio

Dal punto di vista numerico se il grado è basso, in virtù delle concavità e delle convessità di $\sin(2x)$, il polinomio p_N fornisce un'approssimazione scadente, mentre migliora per $N \geq 5$.

Osserviamo che per $N > 10$ tale routine incorre in problemi di condizionamento e stampa un warning del tipo

```
Warning: Polynomial is badly conditioned. Remove repeated data  
points or try centering and scaling as described in  
HELP POLYFIT.
```

Un altro esempio

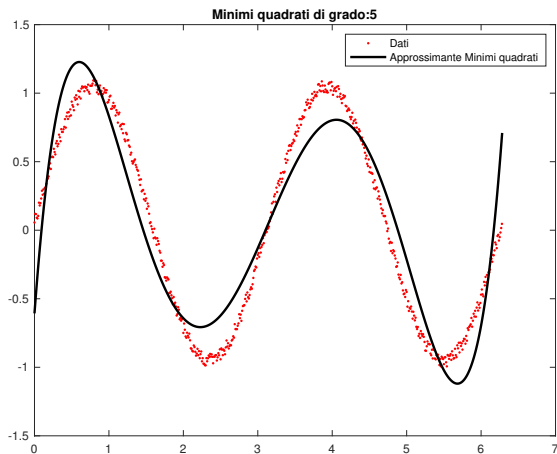


Figura: Grafico che illustra l'approssimazione ai minimi quadrati di grado 5 su una perturbazione della funzione $\sin(2x)$ (campionamento in nodi equispaziati)).

Un altro esempio

Vediamo ora i risultati:

```
>> demo_minimiquadrati
n: 1 norma2: 1.63e+01
n: 2 norma2: 1.63e+01
n: 3 norma2: 1.50e+01
n: 4 norma2: 1.50e+01
n: 5 norma2: 5.50e+00
n: 6 norma2: 5.50e+00
n: 7 norma2: 1.22e+00
n: 8 norma2: 1.22e+00
>>
```

Nota.

I risultati variano da esperimento a esperimento, visto la presenza di numeri casuali.

Nota.

*Ricordiamo che in generale, non ha senso cercare un grado troppo alto del polinomio **di miglior approssimazione** p_N in quanto si otterrebbe a partire da un certo valore il polinomio interpolante, mentre un grado troppo basso, come già detto, non ricostruirebbe adeguatamente l'andamento della funzione f .*

Esercizio

Si supponga che le coppie, (x_k, y_k) , $k = 1, \dots, 21$, corrispondano alla k -sima riga della matrice

```
dati=[ 1.0000e+00  -1.9450e+00  
       1.2000e+00  -1.2530e+00  
       1.4000e+00  -1.1400e+00  
       1.6000e+00  -1.0870e+00  
       1.8000e+00  -7.6000e-01  
       2.0000e+00  -6.8200e-01  
       2.2000e+00  -4.2400e-01  
       2.4000e+00  -1.2000e-02  
       2.6000e+00  -1.9000e-01  
       2.8000e+00   4.5200e-01  
       3.0000e+00   3.3700e-01  
       3.2000e+00   7.6400e-01  
       3.4000e+00   5.3200e-01  
       3.6000e+00   1.0730e+00  
       3.8000e+00   1.2860e+00  
       4.0000e+00   1.5020e+00  
       4.2000e+00   1.5820e+00  
       4.4000e+00   1.9930e+00  
       4.6000e+00   2.4730e+00  
       4.8000e+00   2.5030e+00  
       5.0000e+00   2.3220e+00];
```




Si definisca un file `esercizio_regressione_lineare` come segue.

- Mediante il comando `dati(:,1)`, `dati(:,2)`, si selezionino le ascisse x e le ordinate y .
- Si calcolino i coefficienti P della retta di regressione.
- Stampare il polinomio di regressione p_1^* utilizzando `fprintf`. Il polinomio è $P(1) \cdot x + P(2)$ oppure $P(2) \cdot x + P(1)$?
- Si valuti il polinomio di regressione p_1^* nei punti equispaziati x_j e sia $z_k = p_1^*(x_k)$.
- Si calcoli mediante tali valutazioni, l'errore di regressione

$$\sqrt{\sum_{k=1}^{21} (y_k - z_k)^2}$$

e lo si stampi a video.

- In una figura si determino le coppie estratte dal file di dati e si disegni la retta di regressione.

-  S.D. Conte e C. de Boor, **Elementary Numerical Analysis, 3rd Edition**, Mc Graw-Hill, 1980.
-  Wikipedia, Least Squares,
http://en.wikipedia.org/wiki/Least_squares.
-  Wikipedia, Minimi quadrati,
http://it.wikipedia.org/wiki/Minimi_quadrati.