Minimi quadrati

30 giugno 2007

1 Approssimazione ai minimi quadrati

Si digiti sulla shell di Matlab/Octave

```
>> 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*.
Scriviamo sulla shell di Matlab/Octave `help polyfit`. In una recente release di Matlab appare

```
 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.
```

In altri termini `polyfit` calcola i coefficienti del polinomio p_N di grado N che *meglio* approssima (in norma 2 discreta) la funzione f avente nel vettore di nodi X i valori Y (cioè $Y(i) := f(X(i))$). Questo punto è delicato [5]. In pratica si cerca il polinomio p_N tale che è minima la quantità

$$\|f - P_N\|_{2,d} = \sqrt{\sum_i |f(x_i) - P_N(x_i)|^2}.$$

Digitiamo quindi in un file `minimiquadrati.m`

```
a=0; b=2*pi;
h=0.01;
x=a:h:b;
y=sin(2*x)+(10^(-1))*rand(size(x));
```

```
for n=2:8
    coeff=polyfit(x,y,n); % VALUTAZIONE COEFF.
                          % MIGLIORE APPROX. ("M.A.").

    z=polyval(coeff,x);   % VALORE M.A. NEI NODI "x".
    err2=norm(z-y,2);
    errinf=norm(z-y,inf);

    u=0:(h/4):2*pi;      % NODI TEST.
    v=polyval(coeff,u);  % VALORE M.A. IN NODI TEST.

    plot(x,y,'r-',u,v,'k:');
    fprintf('\n \t [GRADO]: %2.0f [ERR. 2]: %2.2e',n,err2);
    fprintf(' [ERR. INF]: %2.2e',errinf);
    pause(5);
end
```

Osserviamo che

1. `polyval` valuta nei nodi (di test) `u` il polinomio avente coefficienti `coeff` (ordinati dal grado n fino al grado 0);

2. il comando `pause` mette l'esecuzione in pausa per qualche secondo (5 nel nostro caso); la continua successivamente.

Per calcolare gli errori assoluti commessi abbiamo valutato non solo $\|f - P_N\|_{2,d}$ ma pure

$$\|f - P_N\|_{\infty,d} := \max|f(x_i) - P_N(x_i)|.$$

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.
```

Vediamo ora i risultati:

```
>> minimiquadrati

          [GRADO]:  2 [ERR. 2]: 1.63e+001 [ERR. INF]: 1.16e+000
          [GRADO]:  3 [ERR. 2]: 1.50e+001 [ERR. INF]: 1.17e+000
          [GRADO]:  4 [ERR. 2]: 1.50e+001 [ERR. INF]: 1.17e+000
          [GRADO]:  5 [ERR. 2]: 5.48e+000 [ERR. INF]: 6.48e-001
          [GRADO]:  6 [ERR. 2]: 5.48e+000 [ERR. INF]: 6.45e-001
          [GRADO]:  7 [ERR. 2]: 1.23e+000 [ERR. INF]: 1.56e-001
          [GRADO]:  8 [ERR. 2]: 1.23e+000 [ERR. INF]: 1.51e-001
```

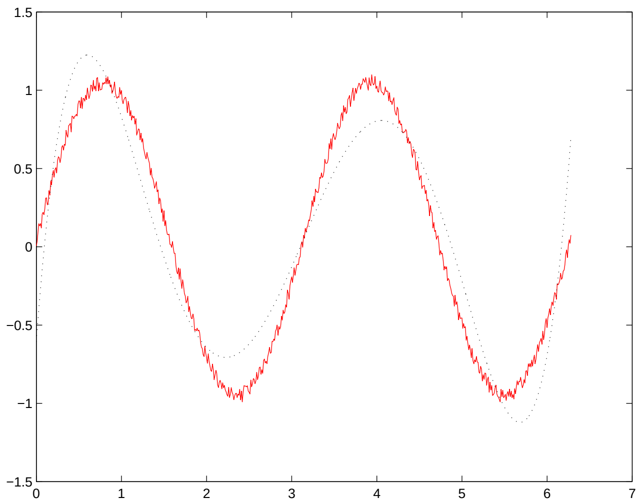

Figure 1: Grafico che illustra l'approssimazione ai minimi quadrati di grado 5 su una *perturbazione* della funzione $\sin(2x)$ (campionamento in nodi equispaziati)).

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.

2 Online

Per ulteriori delucidazioni si considerino le seguenti pagine web

http://it.wikipedia.org/wiki/Minimi_Quadrati
http://www.dsm.units.it/%7Ebellen/calcolo_numerico/CAPIT-4.PDF
http://www.mathworks.com/moler/leastsquares.pdf
http://en.wikipedia.org/wiki/Least_squares

References

[1] K. Atkinson, *Introduction to Numerical Analysis*, Wiley, 1989.

[2] V. Comincioli, *Analisi Numerica, metodi modelli applicazioni*, Mc Graw-Hill, 1990.

[3] S.D. Conte e C. de Boor, *Elementary Numerical Analysis, 3rd Edition*, Mc Graw-Hill, 1980.

[4] The MathWorks Inc., *Numerical Computing with Matlab*, `http://www.mathworks.com/moler`.

[5] A. Quarteroni e F. Saleri, *Introduzione al calcolo scientifico*, Springer Verlag, 2006.

[6] A. Suli e D. Mayers, *An Introduction to Numerical Analysis*, Cambridge University Press, 2003.