Splines

Ángeles Martínez Calomardo e Alvise Sommariva

Università degli Studi di Padova

27 novembre 2013

E' noto che nel caso dell'interpolazione polinomiale, dati $N+1$ punti $a = x_0 < \ldots < x_N = b$, e i valori $y_0, \ldots, y_N$ ivi assunti dalla funzione $y = f(x)$, esiste unico il polinomio p_N di grado N tale che

$$p_N(x_i) = f_i, \;\; i = 0, \ldots, N. \tag{1}$$

Nel caso di nodi equispaziati

$$x_k = a + k\,\frac{(b-a)}{N}, \;\; k = 0, \ldots, N; \tag{2}$$

al crescere di N, non si puó garantire che $f - p_n$ tenda a 0 puntualmente (si ricordi il fenomeno di Runge!).

Problema:
E' possibile calcolare una interpolante di tipo polinomiale s_N per cui al crescere di N si abbia $s_N \to f$ puntualmente?

Sia $[a, b] \subset \mathbb{R}$ chiuso e limitato, e sia $a = x_0 < x_1 < \ldots < x_n = b$. Una spline di grado m (o *ordine* $m + 1$) è una funzione in $C^{m-1}([a, b])$ che in ogni intervallo $[x_i, x_{i+1}]$, con $i = 0, \ldots, n-1$, è un polinomio di grado m.

Alcuni esempi:

- spline di grado 1 (lineari): una funzione in $C([a, b])$ che in ogni intervallo $[x_i, x_{i+1}]$, con $i = 0, \ldots, n-1$, è un polinomio di grado 1.
- spline di grado 3 (cubiche): una funzione in $C^2([a, b])$ che in ogni intervallo $[x_i, x_{i+1}]$, con $i = 0, \ldots, n-1$, è un polinomio di grado 3.

Splines di grado 1 e lineari a tratti

Come anticipato sono funzioni $C([a,b])$ che in ogni intervallo $[x_i, x_{i+1}]$, con $i = 0, \ldots, n-1$, è un polinomio di grado 1. Di conseguenza sono funzioni continue lineari a tratti, essendo un segmento il grafico di un polinomio di grado 1 in $[x_i, x_{i+1}]$.

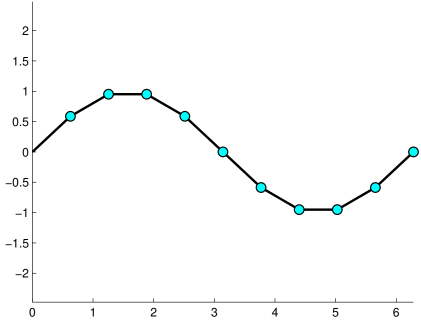

Figura: Splines grado 1 ($n = 10$ intervalli di uguale ampiezza).

Interpolazione con splines: alcuni fatti.

Dati $n+1$ punti in $[a,b]$, diciamo

$$a = x_0 < x_1 < \ldots < x_{n-1} < x_n = b,$$

supponiamo di dover interpolare le coppie (x_k, y_k) con $k = 0, \ldots, n$.

- esiste unica la spline intp. s_1 di grado 1 , cioè tale che $s_1(x_k) = y_k$ per $k = 0, \ldots, n$.
- non esiste unica la spline intp. s_3 di grado 3, cioè tale che $s_3(x_k) = y_k$ per $k = 0, \ldots, n$. Servono 2 ulteriori condizioni per determinare univocamente la spline s_3 interpolante. Alcune classiche richieste aggiuntive.
  - Spline naturale: $s_3^{(2)}(a) = s_3^{(2)}(b) = 0$.
  - Spline periodica: $s_3^{(1)}(a) = s_3^{(1)}(b)$, $s_3^{(2)}(a) = s_3^{(2)}(b)$.
  - Spline vincolata: $s_3^{(1)}(a) = y_a'$, $s_3^{(1)}(b) = y_b'$ (con y_a', y_b' assegnati).

Splines cubiche di tipo *not-a-knot*

La spline cubica s_3^{NAK} con vincolo not-a-knot è definita come segue:

- Interpola le coppie (x_0, y_0), ..., (x_n, y_n);
- E' un polinomio di grado 3 nell'intervallo $[x_0, x_2]$;
- E' un polinomio di grado 3 nell'intervallo $[x_{n-2}, x_n]$

Si osservi che per le spline cubiche generiche s_3 non è detto che la restrizione di s_3 a $[x_0, x_1]$ e $[x_1, x_2]$ siano lo stesso polinomio, e similmente che la restrizione a $[x_{n-2}, x_{n-1}]$ e $[x_{n-1}, x_n]$ siano lo stesso polinomio.

Vediamo il confronto dei gradi di libertá.

- **Richieste**: Abbiamo $n-2$ polinomi da determinare e quindi $4(n-2)$ condizioni.
- **Fornite**: Interpolazione $x_0, \ldots, x_n$: $n+1$ condizioni. Regolarità $x_2, \ldots, x_{n-2}$: $3(n-3)$ condizioni.

Quindi il numero di gradi di libertá richiesto e fornito è lo stesso.

Splines lineari: stima errore

Sia $s_{1,\Delta} : [a, b] \to \mathbb{R}$ una spline di grado 1, che interpola le coppie $(x_i, f(x_i))$ dove $x_0 = a < x_1 < \ldots < x_i < x_{i+1} < \ldots < x_N = b$.
$\Delta = \{x_i\}_{i=0,\ldots,N}$ indica la suddivisione $[a, b] = \cup_{i=0}^{n-1} [x_i, x_{i+1}]$.
Si può mostrare che

$$|f(x) - s_{1,\Delta}(x)| \leq h_{i,\Delta}^2 \frac{M_{i,\Delta}}{8}, \quad x \in [x_i, x_{i+1}], \tag{3}$$

con

- $h_{i,\Delta} = x_{i+1} - x_i$,
- $M_{i,\Delta} := \max_{x \in [x_i, x_{i+1}]} |f^{(2)}(x)|$.

Di conseguenza, se $f \in C^2([a, b])$, per $M = \|f^{(2)}\|_\infty = \max_i M_i$

$$\begin{aligned} \max_{x \in [a,b]} |f(x) - s_{1,\Delta}(x)| &= \max_{i=0,\ldots,N-1} \max_{x \in [x_i, x_i+1]} |f(x) - s_{1,\Delta}(x)| \\ &\leq \max_{i=0,\ldots,N-1} h_{i,\Delta}^2 \frac{M_{i,\Delta}}{8} \leq \frac{M}{8} (\max_i h_{i,\Delta})^2. \end{aligned}$$

Sia $\overline{h}_{\Delta} = \max_i h_{i,\Delta}$. Sia fissata la famiglia di suddivisioni $\{\Delta_n\}$ con la proprietà che $\overline{h}_{\Delta_n} \to 0$ (suddivisioni sempre più fini). Allora la convergenza della successioni di interpolanti s_{1,Δ_n} ad una funzione $f \in C^2([a,b])$ è uniforme. Infatti:

$$\|f - s_{1,\Delta_n}\| \leq \frac{M}{8}(\max_i h_{i,\Delta_n})^2 = \frac{M}{8}\overline{h}^2_{\Delta_n} \overset{n}{\to} 0.$$

Splines cubiche: stima errore

Consideriamo di seguito spline cubiche s_{3,Δ_n} su una suddivisione $\Delta_n = \{x_i\}$ di $[a, b]$. Poniamo $h_{i,\Delta} = x_{i+1} - x_i$, $f^{(k)}$ derivata k-sima di f.

Teorema. Supponiamo $f \in C^4([a, b])$. Posto $\overline{h}_\Delta = \max_i h_{i,\Delta}$, si consideri una sudd. $\Delta = \{x_i\}$ di $[a, b]$ con $K_\Delta = \max_i \overline{h}_\Delta / h_{i,\Delta}$.

Esiste una costante c_s indipendente da $\overline{h}_\Delta$ tale che

$$\|f^{(s)} - s_{3,\Delta}^{(s)}\|_\infty \leq c_s K_\Delta \overline{h}_\Delta^{(4-s)} \|f^{(4)}\|_\infty, \; s = 0, 1, 2, 3 \qquad (4)$$

dove $\|f\|_\infty = \max_{x \in [a,b]} |f(x)|$.

PS. Notare le componenti dell'errore dovute alla suddivisione (cioè K_Δ e $\overline{h}_\Delta$) e alla funzione f (cioè $\|f^{(4)}\|_\infty$).

Corollario.
Supponiamo $f \in C^4([a,b])$. Si consideri la suddivisione $\Delta = \{x_i\}$ di $[a,b]$ con $\overline{h}_\Delta = h_{i,\Delta} = h_\Delta$ (cioè $\{x_i\}$ nodi equisp.). Allora esiste una costante c_0 indipendente da h tale che

$$\|f - s_{3,\Delta}\|_\infty \leq c_0\, h_\Delta^4 ||f^{(4)}||_\infty \quad (5)$$

Inoltre esistono delle costanti c_1, c_2, c_3 indipendenti da h tali che

$$\begin{aligned} \|f^{(1)} - s_{3,\Delta}^{(1)}\|_\infty &\leq c_1\, h_\Delta^3 ||f^{(4)}||_\infty, & (6)\\ \|f^{(2)} - s_{3,\Delta}^{(2)}\|_\infty &\leq c_2\, h_\Delta^2 ||f^{(4)}||_\infty, & (7)\\ \|f^{(3)} - s_{3,\Delta}^{(3)}\|_\infty &\leq c_3\, h_\Delta^1 ||f^{(4)}||_\infty, & (8) \end{aligned}$$

dove $\|f\|_\infty = \max_{x \in [a,b]} |f(x)|$.

Nel caso di spline interpolanti e vincolate si ha $c_0 = 5/384$, $c_1 = 1/24$, $c_2 = 3/8$.

Splines cubiche: convergenza uniforme

Sia $\overline{h}_{\Delta} = \max_i h_{i,\Delta}$. Sia fissata la famiglia di suddivisioni $\{\Delta_n\}$ con la proprietà che $K = max_n K_{\Delta_n} < +\infty$ e $\overline{h}_{\Delta_n} \to 0$ (suddivisioni sempre più fini). Allora la convergenza della successioni di interpolanti s_{3,Δ_n} ad una funzione $f \in C^2([a,b])$ è uniforme.
Infatti da (10):

$$\|f^{(s)} - s_{3,\Delta_n}^{(s)}\|_\infty \leq c_s K \overline{h}_{\Delta_n}^{(4-s)} ||f^{(4)}||_\infty \xrightarrow{n} 0.$$

NB: Per $s = 3$, significa che la successione di funzioni a gradino $s_{3,\Delta_n}^{(3)}$ converge a $f^{(3)}$ uniformemente.

Splines lineari: esempio di Runge.

La funzione di Runge $f(x) = 1/(1+x^2)$ appartiene a $C^{\infty}([-5,5])$ e di conseguenza, nonostante la convergenza puntuale dell'interpolazione polinomiale in nodi equispaziati non sia garantita, ciò non si può dire per l'interpolazione mediante splines lineari o cubiche (sappiamo sussistere la conv. unif.!). Vediamo una stima dell'errore più precisa. Nel caso lineare, se $\Delta = \{x_i\}$ è la sudd. equisp. di ampiezza h_Δ, se $x \in [x_i, x_{i+1}]$

$$|f(x) - s_{1,\Delta}(x)| \leq h_{i,\Delta}^2 \frac{M_{i,\Delta}}{8} = h_\Delta^2 \frac{M_{i,\Delta}}{8} \tag{9}$$

dove

$$M_{i,\Delta} = \max_{x \in (x_i, x_{i+1})} |f^{(2)}(x)|.$$

con

$$f^{(2)}(x) = (8\,x^2)/(x^2+1)^3 - 2/(x^2+1)^2.$$

Dal grafico di $|f^{(2)}|$ deduciamo che

$$M_i \leq M = \|f^{(2)}\|_\infty = |f^{(2)}(0)| = 2$$

e quindi se $x \in [x_i, x_{i+1}]$

$$|f(x) - s_{1,\Delta}(x)| \leq h_\Delta^2 \frac{M_i}{8} \leq h_\Delta^2 \frac{M}{8} \leq \frac{h_\Delta^2}{4} \Rightarrow \|f - s_{1,\Delta}\|_\infty \leq \frac{h_\Delta^2}{4}.$$

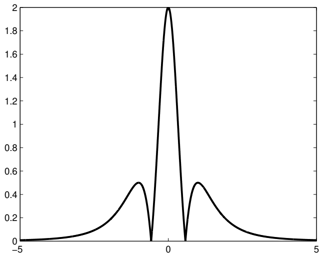

Figura: Grafico di $|f^{(2)}|$ per $f(x) = 1/(1+x^2)$, in $[-5, 5]$.

Splines cubiche: esempio di Runge.

Usiamo le notazioni precedenti. Sappiamo che $\|f - s_{3,\Delta}\|_\infty \leq c_0\, h_\Delta^4 ||f^{(4)}||_\infty$ e si può vedere che

$$f^{(4)}(x) = 24/(x^2+1)^3 - 288 \cdot x^2/(x^2+1)^4 + 384 \cdot x^4/(x^2+1)^5$$

il cui massimo modulo in $[-5, 5]$ è $|f^{(4)}(0)| = 24$. Quindi per splines cubiche intp. vincolate, essendo $c_0 = 5/384$

$$\|f - s_{3,\Delta}\|_\infty \leq c_0\, h_\Delta^4 ||f^{(4)}||_\infty \leq (120/384) h_\Delta^4.$$

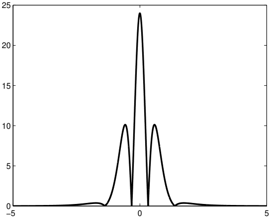

Figura: Grafico di $|f^{(4)}|$ per $f(x) = 1/(1+x^2)$, in $[-5, 5]$.

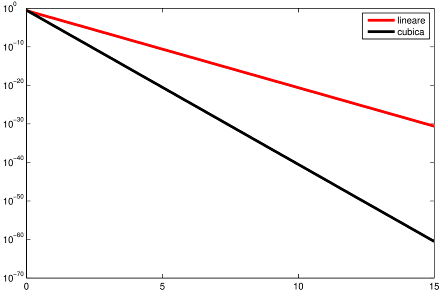

Figura: Grafico stime errori intp. splines lineari e cubiche vincolate (ascissa: $x = -log_{10}(h)$).

```
>> n=0:0.2:15; h=10.^(-n);
>> e1=(h.^2)/4; e3=(120/384)*(h.^4);
>> semilogy(n,e1,'r-',n,e3,'k-');
>> legend('lineare','cubica');
```

Alcuni comandi per calcolare interpolazione spline in Matlab/Octave

```
SPLINE Cubic spline data interpolation.
YY = SPLINE(X,Y,XX) uses cubic spline interpolation to find
    YY, the values of the underlying function Y at the
    points in the vector XX.
The vector X specifies the points at which the data Y is
    given. ...
Ordinarily, the not-a-knot end conditions are used.However,
    if Y contains two more values than X has entries, then
    the first and last value in Y are used as the endslopes
    for the cubic spline.
...
```

Splines interpolanti in Matlab/Octave: spline

```
Example: This generates a sine curve, then samples the
    spline over a finer mesh:

x = 0:10;  y = sin(x);
xx = 0:.25:10; yy = spline(x,y,xx);
plot(x,y,'o',xx,yy)

Example: This illustrates the use of clamped or complete
    spline interpolation where end slopes are prescribed.
    Zero slopes at the ends of an interpolant to the values
    of a certain distribution are enforced:

x = -4:4; y = [0 .15 1.12 2.36 2.36 1.46 .49 .06 0];
cs = spline(x,[0 y 0]);
xx = linspace(-4,4,101);
plot(x,y,'o',xx,ppval(cs,xx),'-');

See also INTERP1, PPVAL, SPLINES (The Spline Toolbox).
```

```
INTERP1 1-D interpolation (table lookup).
YI = INTERP1(X,Y,XI) interpolates to find YI, the values of
    the underlying function Y at the points in the vector XI
    .The vector X specifies the points at which the data Y
    is given. ...
YI = INTERP1(X,Y,XI,'method') specifies alternate methods.
The default is linear interpolation.  Available methods are:
'nearest'  - nearest neighbor interpolation
'linear'   - linear interpolation
'spline'   - piecewise cubic spline interpolation (SPLINE)
'pchip'    - piecewise cubic Hermite interpolation (PCHIP)
'cubic'    - same as 'pchip'
'v5cubic'  - the cubic interpolation from MATLAB 5, which
    does not extrapolate and uses 'spline' if Xmis not
    equally spaced. ...
```

Splines interpolanti in Matlab/Octave: `interp1`

```
For example, generate a coarse sine curve and interpolate
    over a finer abscissa:
x = 0:10; y = sin(x); xi = 0:.25:10;
yi = interp1(x,y,xi); plot(x,y,'o',xi,yi)

See also INTERP1Q, INTERPFT, SPLINE, INTERP2, INTERP3,
    INTERPN.
```

Quindi dall'help di `spline` e `interp1` deduciamo che:

- spline: calcola spline **cubica** intp. con vincolo *not-a-knot* e con qualche assegnazione anche *vincolate*;
- interp1: calcola spline **lineare** o spline **cubica** intp. con vincolo *not-a-knot* e con qualche assegnazione anche di altro tipo.

Splines interpolanti in Matlab/Octave: esempio di Runge

Vediamo il comportamento della spl. lineare intp. la funzione di Runge in 13 nodi equisp., ricordando che l'interpolante polinomiale p_{12} su 13 nodi, aveva $\|f - p_{12}\| \approx 3.66 + 00$. Il file è `runge_lin.m`.

```
n=13;
f=inline('1./(1+x.^2)'); % RUNGE.
t=linspace(-5,5,1000); % NODI TEST.
x=linspace(-5,5,n); % NODI SUDDIV.
y=feval(f,x); % (x,y) COPPIE INTP.
st=interp1(x,y,t,'linear');
ft=feval(f,t);
err=norm(ft-st,inf); % ERRORE |f-s3|
fprintf('\n \t err: %2.2e \n',err);
```

Otteniamo, come supposto, un migliore risultato rispetto a p_{12} (stime: $\leq ((10/12)^2)/4 \approx 0.17$.)

```
>> runge_lin
   err: 6.50e-02
>>
```

Splines interpolanti in Matlab/Octave: esempio di Runge

Vediamo il comportamento delle spline cubiche not-a-knot interpolanti la funzione di Runge in 13 nodi equispaziati (avevamo $\|f - p_{12}\| \approx 3.66 + 00$). Il file è runge_cub.m.

```
n=13;
f=inline('1./(1+x.^2)'); % RUNGE.
t=linspace(-5,5,1000); % NODI TEST.
x=linspace(-5,5,n); % NODI SUDDIV.
y=feval(f,x); % (x,y) COPPIE INTP.
st=interp1(x,y,t,'spline');
ft=feval(f,t);
err=norm(ft-st,inf); % ERRORE |f-s3|
fprintf('\n \t err: %2.2e \n',err);
```

Otteniamo, come supposto, un migliore risultato rispetto a p_{12} e alle splines lineari (errore come h^4 e non h^2)

```
>> runge_lin
   err: 6.91e-03
>>
```

Esercizio

Un esempio interessante è l'approssimazione del cerchio con splines. Essendo l'equazione del cerchio

$$f(t) = \begin{pmatrix} \cos(t) \\ \sin(t) \end{pmatrix} \tag{10}$$

per $t = [0, 2\pi]$. Se

$$t_k = \frac{2k\pi}{N},\ k = 0, \ldots, N,$$

l'interpolante spline puó essere facilmente calcolata via il comando Matlab `interp1` applicato alla coppia di vettori `t`, `u` e `t`, `v` dove al solito $t = (t_k)$, $u = (u(t_k))$, $v = (v(t_k))$.

Esercizio

Per testare il comportamento di tale ricostruzione,

- fissati dei nodi test

$$t_k^* = \frac{2k\pi}{100}, \; k = 0, \ldots, 100,$$

  si calcolino per $M = 10$ le splines s_x, s_y lineari (o cubiche not-a-knot) interpolanti le coppie

$$(t_k, \cos(t_k)), \; t_k = \frac{2k\pi}{M}, \; k = 0, \ldots, M,$$

$$(t_k, \sin(t_k)), \; t_k = \frac{2k\pi}{M}, \; k = 0, \ldots, M,$$

- si plottino le coppie $(s_x(t_k^*), s_y(t_k^*))$.

Ripetere l'esercizio per $M = 100$.