Laboratorio di Calcolo Numerico: Interpolazione e approssimazione di funzioni

Giacomo Elefante

Laboratorio di calcolo numerico
14/12/23

Scopo di oggi

Analizzare algoritmi per l'interpolazione e approssimazione di dati e funzioni:

- Problema di interpolazione e matrice di Vandermonde
- Interpolazione su nodi equispaziati
- Interpolazione su nodi di Chebyshev

Problema di interpolazione

Dato $\mathbb{P}_n$ lo spazio dei polinomi di gradi al più n, ovvero i polinomi del tipo

$$a_n x^n + a_{n-1} x^{n-1} + \dots a_1 x + a_0$$

con $a_i \in \mathbb{R}, i = 0, \dots n$ allora il problema di interpolazione, ovvero trovare una funzione $p(x)$ che dati $n+1$ coppie $(x_0, y_0), \dots (x_n, y_n)$ soddisfa alla relazione

$$p(x_i) = y_i, \qquad i = 0, \dots, n,$$

ha una soluzione (se gli x_i sono distinti) con $p(x) \in \mathbb{P}_n$ ed è unica.

Problema di interpolazione

In particolare, il polinomio si può costruire utilizzando le basi monomiali $\{1, x, x^2, ...\}$ e andando a risolvere il problema

$$Va = y, \tag{1}$$

dove V è la matrice di Vandermonde $V_{ij} = x_i^j$, $a = [a_0, \ldots, a_n]'$ e $y = [y_0, \ldots, y_n]'$.
Scegliendo una base differente di $\mathbb{P}_n$, si può cercare il polinomio interpolante costruito su quella base, ovvero data $\{b_i\}_{i=0}^n$ un'altra base di $\mathbb{P}_n$, si può cercare il polinomio

$$p(x) = \sum_{i=0}^{n} a_i b_i(x)$$

che soddisfi il problema di interpolazione.

Problema di interpolazione

I coefficienti del polinomio interpolante saranno la soluzione del problema

$$Va = y, \tag{2}$$

dove V è la matrice di Vandermonde $V_{ij} = b_j(x_i)$, $a = [a_0, \dots, a_n]'$ e $y = [y_0, \dots, y_n]'$.
Un'ottima base è quella data dai polinomi di Chebyshev

$$T_n(x) = \cos(n \arccos x)$$

infatti il condizionamento della matrice di Vandermonde in quella base è assai migliore rispetto a quello con la base monomiale.

Esercizio

Usando il comando `gallery('chebvand',x)`, si costruisca la matrice di Vandermonde con la base di Chebyshev su $n+1$ punti equispaziati e si calcoli il condizionamento di tale matrice. Ugualmente si faccia con la base monomiale e si faccia un grafico in scala semilogaritmica dei condizionamenti per le matrici ottenute da 1 a $M = 30$ punti.

Interpolazione su Matlab

Su Matlab, gli $n + 1$ coefficienti a_i, possono essere trovati attraverso facilmente con il comando `polyfit`, infatti date delle coppie di dati $(x_0, y_0), \ldots, (x_n, y_n)$, `polyfit` può generare i coefficienti del polinomio interpolatorio tra questi punti.
In seguito per valutare un polinomio su dei punti, dati i coefficienti, è utile il comando `polyval`.

```
function t=interpol(x,y,s)
% Input:
%         x,y  vettori di ascisse cui si associa il polinomio interpolatore "p_n"
%         s    vettore di ascisse in cui valutare il polinomio interpolatore
% Output:
%         t    vettore di valutazioni "p_n(s)"

m=length(x)-1;
coeff=polyfit(x,y,m);
t=polyval(coeff,s);
```

Figure: Codice per generare le valutazioni del polinomio interpolante.

Nodi equispaziati

Consideriamo ora come nodi $n+1$ punti equispaziati in un intervallo $[a, b]$, generabili attraverso i comandi `linspace` oppure `a:h:b` definendo però il passo `h` come

$$h = \frac{b-a}{n}.$$

Essi potranno essere usati per interpolare una funzione f.

Esercizio

Esercizio

Data la funzione $f(x) = \frac{e^{-2x}}{3} + x^2 + 7x^3$, si crei uno script dove vengono definiti

- Una variabile `n` indicante il grado massimo del polinomio interpolante
- Un vettore `x` contenente gli $n + 1$ nodi equispaziati in $[-1, 1]$
- Un vettore `y` i cui elementi sono $f(x_i)$
- Un vettore `s` con 500 punti di valutazione in $[-1, 1]$.

Utilizzare quindi la routine `interpol` per valutare il polinomio interpolante in questi punti e creare un grafico dove vengono sovrascritti la funzione f in blu e il polinomio interpolante in rosso tratteggiato.
Calcolare quindi l'errore assoluto massimo tra il polinomio e la funzione e stamparne il valore a schermo. Come si comporta al variare di n?

In questo caso la convergenza dovrebbe essere assicurata per il tipo di funzione scelta ma al crescere di n si vede l'interpolante costruita sui nodi equispaziati perdere di efficacia e crescere il suo errore. Infatti, tale interpolante non è molto stabile poiché la sua costante di Lebesgue cresce esponenzialmente con n.

Per l'interpolazione infatti, si preferisce utilizzare delle distribuzioni di punti migliori come i punti di Chebyshev di prima specie

$$x_i = -\cos\left(\frac{(2i+1)}{2(n+1)}\pi\right), \qquad i = 0, \dots, n$$

o quelli di Chebyshev-Lobatto

$$x_i = -\cos\left(\frac{i\pi}{n}\right), \quad i = 0, \dots, n,$$

la cui interpolante costruita su questi punti ha una costante di Lebesgue che cresce come $\log(n)$, quindi più lentamente.

Esercizio

Ripetere l'esercizio precedente ma utilizzando i due insiemi di punti descritti sopra al posto di quelli equispaziati.

Esercizio

In alcuni casi invece la convergenza con i punti equispaziati può non esserci proprio, come può essere il caso della funzione di Runge in $I = [-1, 1]$, ovvero

$$f(x) = \frac{1}{1 + 25x^2}$$

Si costruisca, quindi, uno script dove al variare del grado del polinomio n da 1 a 25, si calcola il polinomio interpolante sia su nodi equispaziati sia su nodi di Chebyshev-Lobatto, valutandone poi il valore su 1000 punti equispaziati s in I.

Trovare quindi l'errore assoluto massimo dei polinomi con la funzione in s e fare in seguito la sovrapposizione dei grafici in scala semilogaritmica degli errori in funzione del grado del polinomio.

Esercizio (per casa)

Esercizio

Creare una funzione `leja` che genera una approssimazione dei punti di leja dato un grande numero di punti nell'intervallo $[-1, 1]$. Essa prenderà in input un valore `n` corrispondente a quanti punti si vogliono generare. Internamente si andrà a creare una griglia di 10000 punti in $[-1, 1]$, denominata `T`, e partendo dall'elemento $x_1 = 1$, andrà a calcolare l'elemento x_i a partire dai $i - 1$ punti estratti precedentemente. Ovvero, l'elemento i sarà il punto $\bar{x} \in \mathtt{T}$ tale che

$$\bar{x} = \arg\max_{x \in \mathtt{T}} \prod_{j=1}^{i-1} |x - x_j|$$

Una volta estratti gli `n` punti, devono essere ordinati in maniera crescente prima di essere restituiti come output.