

1 Lezione del 07-04-25

1.1 Interpolazione di funzioni

Poniamo di avere una qualche funzione $f(x) : \mathbb{C} \rightarrow \mathbb{C}$ che non conosciamo esplicitamente ma di cui possiamo solo calcolare il valore y_0, \dots, y_k in determinati punti x_0, \dots, x_k non necessariamente equispaziati fra di loro. Il problema sarà allora trovare un'approssimazione di $f(x)$ **interpolando** i punti dati.

Un approccio valido quando si hanno pochi punti (k) è considerare un polinomio di grado k che passi per tutti questi, sperando che si raggiunga un'approssimazione sufficientemente precisa nei punti esterni ai nodi.

Un'altro approccio, quando si hanno molti punti, è trovare una funzione particolarmente semplice (solitamente una retta) che minimizzi la distanza da questi. E' questo ad esempio il caso della *regressione lineare*.

1.1.1 Interpolazione polinomiale

Vogliamo quindi trovare un polinomio $p(x)$ tale che:

$$p(x_i) = y_i, \quad \forall i \in \{0, \dots, k\}$$

Quello che facciamo è porre il polinomio come:

$$p(x) = a_0 + a_1x + \dots + a_kx^k$$

e impostare il sistema:

$$\begin{cases} a_0 + a_1x_0 + \dots + a_kx_0^k = y_0 \\ \vdots \\ a_0 + a_1x_k + \dots + a_kx_k^k = y_k \end{cases}$$

che è lineare e ha come incognita il vettore dei coefficienti del polinomio cercato.

Notiamo di aver scelto come base la base canonica dei polinomi:

$$B_{\text{canonica}} = \{1, x, x^2, \dots, x^k\}$$

Avremmo potuto scegliere qualsiasi altra base dello spazio dei polinomi $\mathbb{R}[x]$, e vedremo infatti anche questo caso.

Riscrivendo quindi il sistema di cui sopra in forma matriciale si ha:

$$\underbrace{\begin{pmatrix} 1 & x_0 & \dots & x_0^k \\ \vdots & & & \vdots \\ 1 & x_k & \dots & x_k^k \end{pmatrix}}_V \underbrace{\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{pmatrix}}_a = \underbrace{\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_k \end{pmatrix}}_y$$

dove la matrice V viene detta **matrice di Vandermonde**.

Potrebe interessarci capire quando questo sistema ammette soluzione. Ad esempio, è impossibile che esista $i \neq j$ tali che $x_i = x_j$, in quanto questo renderebbe il sistema singolare (due righe uguali $\implies \det(V) = 0$). Questo caso non è desiderabile in quanto potrebbero esserci più soluzioni (se anche $y_i = y_j$) o nessuna (se $y_i \neq y_j$).

Possiamo allora enunciare il teorema:

Teorema 1.1: Determinante della matrice di Vandermonde

Il determinante della matrice di Vandermonde è:

$$\det(V) = \prod_{0 \leq i < j \leq k} (x_j - x_i)$$

Da questo deriva che se $x_i \neq x_j \forall i \neq j$, allora $\det(V) \neq 0$, cioè V non è singolare. Per trovare $p(x)$ risolviamo quindi:

$$Va = y$$

e poniamo:

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

da cui come avevamo detto si ottiene un polinomio con $\deg(p(x)) \leq k$ (\leq perchè si potrebbe avere il coefficiente $a_k = 0$, e così via, cioè il grad è al più k).

Risolvere il sistema costa quindi, come avevamo visto, un qualcosa che è $\sim O(n^3)$, con particolare interesse al numero di condizionamento:

$$\mu(V) = |V| \cdot |V^{-1}|$$

che solitamente per la matrice di Vandermonde è particolarmente grande.

Vediamo che in verità la situazione è sotto controllo se:

$$\mu(A) \leq \frac{1}{u}$$

per u precisione macchina, in quanto l'errore dato dal cattivo condizionamento diventa comparabile con la precisione macchina, comunque piuttosto piccola. Un altro caso di interesse è quello in cui i punti x_i sono equispaziati sulla circonferenza unitaria sul piano di Argand-Gauss. Vediamo che anche qui l'errore è mantenuto sotto controllo, e questo sarà infatti il caso che considereremo per la *trasformata (discreta) veloce di Fourier*.

Un'idea potrebbe essere quindi quella di cambiare la base dei monomi in:

$$B' = \{\phi_0(x), \phi_1(x), \dots, \phi_k(x)\}$$

e porre quindi:

$$p(x) = a_0 \phi_0(x) + a_1 \phi_1(x) + \dots + a_k \phi_k(x)$$

Il sistema alla sostituzione dei nodi continuerà ad essere lineare, in quanto:

$$p(x_i) = a_0 \phi_0(x_i) + a_1 \phi_1(x_i) + \dots + a_k \phi_k(x_i), \quad \forall i \in \{0, \dots, k\}$$

Potremo quindi impostare un altro sistema lineare:

$$\begin{pmatrix} \phi_0(x_0) & \phi_1(x_0) & \dots & \phi_k(x_0) \\ \vdots & & & \vdots \\ \phi_0(x_k) & \phi_1(x_k) & \dots & \phi_k(x_k) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_k \end{pmatrix}$$

Abbiamo quindi definito la matrice di Vandermonde "generalizzata" a:

Definizione 1.1: Matrice di Vandermonde

La matrice di Vandermonde su una certa base polinomiale $\{\phi_0(x), \dots, \phi_k(x)\}$ è definita come:

$$V_{ij} = \phi_j(x_i), \quad \forall i, j \in \{0, 1, \dots, k\}$$

Finora avevamo preso il caso della base canonica, cioè $\{1, x, \dots, x^k\}$, ma nessuno ci nega di prendere altre basi. Basi che ci saranno di particolare interesse, vedremo, sono la base di **Lagrange** e la base di **Newton**.

se hai voglia esempio condizionamento (foto)

1.1.2 Base di Lagrange

Presi sempre i nostri nodi di interpolazione, decidiamo di scegliere la base guardando proprio tali nodi (anziché prenderla a priori, come avevamo fatto per Vandermonde).

Chiamiamo i polinomi di base $l_0(x), \dots, l_k(x)$, con:

$$l_i(x) = (x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_k) = \prod_{j \neq i}^n (x - x_j)$$

cioè il prodotto dei fattori che si annullano in x_j per tutti gli $j \neq i$. Spesso poi decidiamo di normalizzare i polinomi di base, indicandoli come:

$$l_i(x) = \frac{(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_k)}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_k)} = \prod_{j \neq i}^n \frac{x - x_j}{x_i - x_j}$$

Proviamo quindi ad imporre le condizione, con:

$$p(x) = a_0 l_0(x) + a_1 l_1(x) + \dots + a_k l_k(x)$$

valutando il polinomio in un punto x_i :

$$p(x_i) = y_i = a_0 l_0(x_i) + \dots + a_{i-1} l_{i-1}(x_i) + a_i l_i(x_i) + a_{i+1} l_{i+1}(x_i) + \dots + a_k l_k(x_i)$$

dove vediamo che, normalizzando, si conserva solo il termine in a_i , da cui immediatamente:

$$a_i = y_i$$

e:

$$p(x) = y_0 l_0(x) + y_1 l_1(x) + \dots + y_k l_k(x)$$

Possiamo quindi costruire il polinomio interpolante senza fare nessun conto, semplicemente prendendo la combinazione lineare a coefficienti y_i della base di lagrange l_i costruita sui nodi.

Vediamo quindi la forma della matrice di Vandermonde:

$$V_l = \begin{pmatrix} l_0(x_0) & l_1(x_0) & \dots & l_k(x_0) \\ \vdots & & & \vdots \\ l_0(x_k) & l_1(x_k) & \dots & l_k(x_k) \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

da cui abbiamo quindi che si ricava direttamente l'identità e quindi $Va = y$ diventa $Ia = y \implies a = y$.

1.1.3 Base di Newton

Vediamo quindi la base di Newton, che costruiamo come:

$$\begin{cases} n_0(x) = 1 \\ n_1(x) = x - x_0 \\ n_2(x) = (x - x_0)(x - x_1) \\ \vdots \\ n_k(x) = (x - x_0) \dots (x - x_{k-1}) \end{cases}$$

cioè ogni n_i dipende solo dai $\{x_0, \dots, x_{i-1}\}$ precedenti. Questo rende più facile l'aggiornamento se si aggiunge un nuovo nodo in x_{k+1} , per cui basterà dire:

$$n_{k+1}(x) = (x - x_0) \dots (x - x_k) = n_k(x)(x - x_k)$$

La matrice di Vandermonde sarà quindi:

$$V_n = \begin{pmatrix} n_0(x_0) & n_1(x_0) & \dots & n_k(x_0) \\ \vdots & & & \vdots \\ n_0(x_k) & n_1(x_k) & \dots & n_k(x_k) \end{pmatrix}$$

con:

$$p(x) = a_0 n_0(x) + a_1 n_1(x) + \dots + a_k n_k(x)$$

e nel nodo:

$$p(x_i) = y_i \Leftrightarrow V a = y$$

dallo svolgimento della matrice di Vandermonde si ha che la prima colonna è tutta di 1, e il resto della matrice è triangolare inferiore:

$$V_n = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & ? & \dots & 0 \\ \vdots & & & \vdots \\ 1 & ? & \dots & ? \end{pmatrix}$$

e quindi ogni a_i dipende solo dagli $\{y_0, \dots, y_i\}$. Infine, notiamo di poter scrivere ogni a_i come le **differenze divise**:

$$a_i = f[x_0, \dots, x_i]$$

indicate con le parentesi quadre, che definiamo come:

$$f[x_0, \dots, x_n, x] = \begin{cases} f(x) & n = -1 \\ \frac{f(x) - f(x_0)}{x - x_0} & n = 0 \\ \frac{f[x_0, \dots, x_{n-2}, x] - f[x_0, \dots, x_{n-1}, x]}{x - x_{n-1}}, & \text{altrimenti} \end{cases}$$