ESAME CALCOLO NUMERICO PROVA DI LABORATORIO
LAUREA IN INFORMATICA
PROGETTO PER LAUREANDI FEBBRAIO-APRILE 2023

Consegna progetto: email al docente contenente una cartella zippata con tutti i files. La cartella deve essere uno "standalone", nessun file esterno deve essere chiamato e deve poter girare su un installazione di Matlab "recente" (in particolare devono essere chiamate solo functions supportate da R2022b). La cartella non deve contenere materiale inutile (ad esempio files .asv, functions non chiamate ecc). Tutte le functions devono essere corredate da help, ma non sono ammessi commenti di altro genere nei programmi che si discuteranno con il docente. Il programma consegnato deve girare in tutte le sue parti, al contrario degli esami svolti in aula, i programmi che non girano (anche parzialmente) non verranno considerati. Si può includere nella cartella una breve relazione (non richiesta).

Termine per la consegna: 22/02/2023

Discussione: lo studente deve conoscere il programma consegnato in ogni sua parte e deve essere in grado di motivare le soluzioni implementate. Questo sarà oggetto di una breve (10 minuti circa) discussione.

---

1. Introduzione al progetto

1.1. **Obiettivo.** Supponiamo noti i dati $(x_1, \dots, x_M)$ e $(y_1, \dots, y_M)$, dove assumiamo che y_i derivi da un campionamento rumoroso di una funzione ignota $f : [-1, 1] \to \mathbb{R}$, ovvero $y_i = f(x_i) + \epsilon_i$. Vogliamo stimare la soluzione del problema di ottimizzazione

$$x^* = \underset{x \in [a,b]}{\operatorname{argmin}}\, f(x). \tag{P}$$

A tal fine vogliamo costruire (a partire dai dati) un modello surrogato $\tilde{f}_\alpha$ per f (eventualmente dipendente dai parametri α) e calcolare una soluzione del problema

$$x^*_\alpha = \underset{x \in [a,b]}{\operatorname{argmin}}\, \tilde{f}_\alpha(x). \tag{$\tilde{P}_\alpha$}$$

anzichè risolvere (P). Questo problema può essere affrontato cercando gli zeri di $\frac{d}{dx}\tilde{f}_\alpha(x)$ con il metodo di Newton e controllando per ognuno di essi se vale $\frac{d}{dx}\tilde{f}_\alpha \geq 0$.

Dato che i dati sono rumorosi e i campionamenti sono assegnati (cioè non possiamo decidere dove effettuare i campionamenti) può essere conveniente utilizzare un approccio di minimi quadrati con regolarizzazione. In particolare, penalizzeremo i polinomi approssimanti con derivata terza "grande". Questo *dovrebbe* (è solo un euristica!) garantire che, se l'andamento della funzione ignota è convesso, allora anche l'approssimazione ottenuta lo è.

Prima di passare al dettaglio implementativo, descriviamo e richiamiamo tutti gli oggetti matematici e le formule che saranno utilizzate nello sviluppo del progetto. Conviene capire molto bene quanto segue prima di iniziare a programare.

1.2. **Tecniche di calcolo numerico da utilizzare.**

♦ **1** (Costruzione di un modello dei dati)**.** Il modello surrogato $\tilde{f}_\alpha$ che considereremo sarà la migliore approssimazione ai minimi quadrati di grado n, regolarizzati alla Tikonov con parametro λ ovvero($\alpha = (n, \lambda)$ con $n \in \mathbb{N}$ e $\lambda > 0$). Tale funzione è definita come

$$\tilde{f}_{n,\lambda} := \underset{q \in \mathbb{P}^n}{\operatorname{argmin}} \frac{1}{M} \sum_{i=1}^{M} |y_i - q(x_i)|^2 + \lambda \int_a^b |q'''(x)|^2 dx.$$

Si noti che questa è una generalizzazione dei classici minimi quadrati che si realizzano per $\lambda = 0$.

Ciò vuol dire che, fissata una qualsiasi base $p_0(x), p_1(x), \ldots, p_n(x)$ dello spazio $\mathbb{P}^n$ dei polinomi di grado al più n, avremo

$$\tilde{f}_{n,\lambda}(x) = \sum_{k=0}^{n} c_k^* p_k(x), \tag{1}$$

dove il vettore dei coefficienti $c^* \in \mathbb{R}^{n+1}$ è l'unico minimizzatore in $\mathbb{R}^{n+1}$ della funzione convessa

$$c \mapsto F(c) := \frac{1}{2} \left(\frac{\|y - Vc\|_2^2}{M} + \lambda c^t G_1 c \right).$$

Qui abbiamo usato la notazione V per la matrice di Vandermonde della base $p_0, p_1, \ldots, p_n$ relativa ai nodi $x_1, x_2, \ldots, x_M$, e la notazione G_1 per la matrice gramiana

$$G_1(i, j) = \int_a^b p_i'''(x) p_j'''(x) dx.$$

Derivando F possiamo dunque caratterizzare c^* come soluzione del sistema lineare

$$Gc = b, \quad G := \left(\frac{V^t V}{M} + \lambda G_1 \right), \quad b := \frac{V^t y}{M}. \tag{2}$$

Si noti che, disponendo per ogni $x \in [a, b]$ delle matrici riga di Vandermonde derivate della base $p_0, p_1, \ldots, p_n$, ossia

$$\begin{aligned} V_{0,x}(1, j) &:= p_j(x) \\ V_{1,x}(1, j) &:= p_j'(x) \\ V_{2,x}(1, j) &:= p_j''(x), \end{aligned}$$

abbiamo

$$\begin{aligned} \tilde{f}_{n,\lambda}(x) =& V_{0,x} c^* \\ \tilde{f}'_{n,\lambda}(x) =& V_{1,x} c^* \\ \tilde{f}''_{n,\lambda}(x) =& V_{2,x} c^* \end{aligned}$$

♦ **2** (Matrici di derivazione di Chebyshev)**.** Data una base $p_0, p_1, \ldots, p_n$ di $\mathbb{P}^n$, possiamo osservare che $p_j' \in \mathbb{P}^n$ per ogni $j = 0, 1, \ldots, n$. Vista la linearità dell'operazione di derivazione, esistono dei coeffocienti $D_{i,j}$ tali per cui

$$p_j'(x) = \sum_{i=0}^{n} p_i(x) D_{i,j}.$$

In altre parole, i coefficienti $c^{(1)}$ della derivata del polinomio $p(x) = \sum_{j=0}^{n} p_j(x) c_j$ sono calcolabili come

$$c_i^{(1)} = \sum_{j=0}^{n} D_{i,j} c_j, \text{ ossia } c^{(1)} = Dc,$$

dove $D = [D_{i,j}]_{i,j=0,1,\ldots,n}$.

Allo stesso modo i coefficienti $c^{(2)}$ e $c^{(3)}$ delle derivate seconda e terza del polinomio p (avente coefficienti c) saranno dati da

$$\begin{aligned} c^{(2)} &= Dc^{(1)} = DDc \\ c^{(3)} &= Dc^{(2)} = DDc^{(1)} = DDDc\,. \end{aligned}$$

Assumiamo ora di utilizzare la base di Chebyshev $p_k(x) = \cos(k \operatorname{acos}(x))$, $k = 0, 1, \ldots, n$. In questo caso abbiamo

$$p_k'(x) = \begin{cases} 0 & k = 0 \\ 2k \sum_{j=0, \text{ dispari}}^{k-1} p_j(x) & k > 0 \text{ pari} \\ 2k \left(\sum_{j=0, \text{ pari}}^{k-1} \right)' p_j(x) & k \text{ dispari} \end{cases}, \tag{3}$$

dove $\sum'$ indica una sommatoria dove il termine di ordine 0 viene dimezzato. Ad esempio per $n = 5$ si ha

$$D = \begin{pmatrix} 0 & 1 & 0 & 3 & 0 & 5 \\ 0 & 0 & 4 & 0 & 8 & 0 \\ 0 & 0 & 0 & 6 & 0 & 10 \\ 0 & 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

E' dunque evidente che se disponiamo della matrice di Vandermonde V nella base di Chebyshev per dei nodi $x_1, x_2, \ldots, x_M$, la matrice di Vandermonde delle derivate di tale base si scrive come VD con D calcolata con lo stesso valore di n usato nel calcolo di V.

N.B. *Una funzione matlab per il calcolo della matrice D è fornita in allegato da docente.*

♦ **3** (Costruzione di formula di quadratura interpolatoria)**.** Sia V^{quad} la matrice di Vandermonde di una base $p_0, p_1, \ldots, p_r$ di $\mathbb{P}^r$ su $r + 1$ nodi distinti $x_0^{quad}, x_1^{quad}, \ldots, x_r^{quad}$ e assumiamo di saper calcolare tramite formule esatte i momenti $m = (m_0, m_1, \ldots, m_r)$ di tale base:

$$m_i = \int_a^b p_i(x) dx.$$

Possiamo calcolare dei pesi $w_0, w_1, \ldots, w_r$ di quadratura interpolatoria esatta di grado n osservando che essi risolvono il sistema dei momenti

$$(V^{quad})^t w = m. \tag{4}$$

Attenzione: in generale questa procedura non da garanzie sulla positività dei pesi. Tuttavia, se utilizziamo nodi di Chebyshev-Lobatto in $[-1,1]$ e la base di Chebyshev, i pesi calcolati sono positivi per ogni $r > 0$.

I momenti della base di Chebyshev in $[-1,1]$ si calcolano facilmente: quello di ordine 0 ($m_0 = \int_{-1}^{1} 1dx$) è ovviamente 2, gli altri sono ovviamente nulli per i gradi dispari (vista la simmetria), mentre i gradi pari si procede per sostituzione e con le formule di addizione del seno. Si ha dunque

$$m_k = \begin{cases} \frac{2}{1-k^2} & k \text{ pari} \\ 0 & k \text{ dispari} \end{cases} . \tag{5}$$

♦ **4** (Costruzione della matrice gramiana)**.** Per il calcolo della matrice G_1 possiamo procedere per quadratura, supponendo di disporre della matrice di derivazione della base $p_0, p_1, \ldots, p_n$ e di una formula di quadratura $(x_0^{quad}, x_1^{quad}, \ldots, x_{2n-6}; w_0, w_1, \ldots, w_{2n-6})$ in $[a,b]$, estatta di grado $2n-6$. Si ha infatti

$$G_1(i,j) = \int_a^b p_i'''(x) p_j'''(x) dx = \sum_{k=0}^{2n-6} p_i'''(x_k^{quad}) p_j'''(x_k^{quad}) w_k,$$

visto che $p_i'''(x)p_j'''(x) \in \mathbb{P}^{2n-6}$. Supponiamo che V^{quad} sia la matrice di Vandermonde della base $p_0, p_1, \ldots, p_n$ di $\mathbb{P}^n$ calcolata sui punti di quadratura e D la matrice di derivazione. Allora abbiamo

$$G_1 = (V^{quad}DDD)^t diag(w_0, \ldots, w_{2n-6}) V^{quad} DDD. \tag{6}$$

D'altro canto la matrice $\frac{V^tV}{M}$, dove V è la matrice di Vandermonde della base di Chebyshev nei punti di campionamento, può essere scritta come

$$G_1 = V^t diag(1/M, \ldots, 1/M) V. \tag{7}$$

Abbiamo dunque

$$G = \begin{bmatrix} V^t & (V^{quad})^t \end{bmatrix} \cdot \begin{bmatrix} \mathbb{I}/M & \mathbb{O} \\ \mathbb{O} & \lambda diag(w) \end{bmatrix} \cdot \begin{bmatrix} V \\ V^{quad} \end{bmatrix} \tag{8}$$

$$= \begin{bmatrix} V^t/\sqrt{M} & (V^{quad})^t diag(\sqrt{\lambda w}) \end{bmatrix} \cdot \begin{bmatrix} V/\sqrt{M} \\ diag(\sqrt{\lambda w}) V^{quad} \end{bmatrix}$$

$$=: L^t L \tag{9}$$

♦ **5** (Inversione della matrice gramiana)**.** Come visto a lezione per i minimi quadrati standard, se è nota la fattorizzazione (9), conviene lavorare sulla matrice L.

Infatti, una volta calcolata fattorizzazione qr di L, $L = QR$, si ha $G = R_0^t R_0$ (fattorizzazione di Choleski), dove R_0 è la parte quadrata superiore di R, e il sistema $Gc = b$, equivalente a $R_0^t R_0 c = b$, può essere risolto usando gli algoritmi di sostituzione avanti e indietro.

2. Specifiche dell'implementazione

Passo 1. Si scriva l'help per la function `cheb_diff.m` fornita in allegato dal docente. Essa calcola la matrice di derivazione D della base di Chebishev di grado `n` descritta nella formula (3).

Si crei una function avente la chiamata

`V=cheb_vand(n,x)`

che, preso in ingresso il grado `n` da considerarsi e il vettore colonna `x` di punti in $[-1,1]$ calcoli la matrice di Vandermonde `V` di tale base sui punti `x`.

Passo 2. Si crei una function avente la chiamata

`[xquad,w]=cheb_quad(n)`

che preso in ingresso il grado di esattezza polinomiale `n` da considerare costruisca il vettore colonna `xquad` di nodi di quadratura e il vettore riga dei relativi pesi calcolati risolvendo il problema dei momenti (4) con termine noto definito in (5). Per la soluzione del sistema lineare si utilizzi il metodo LU con pivoting parziale e le function di sostituzione avanti ed indietro fornite dal docente.

Passo 3. Si crei una function avente la chiamata

`[cstar,Rzero,b]=cheb_tikonov(n,lambda,xsample,ysample)`

che presi in ingresso il grado polinomiale `n` da considerare, il parametro di regolarizzazione `lambda`, i valori di `xsample,ysample` del data-set, costruisca la matrice `Rzero` (fattorizzazione di Choleski di G) e il termine noto `b` del sistema lineare (2) (si veda anche (9) e righe a seguire) e calcoli la sua soluzione `cstar` tramite algoritmi di sostituzione avanti ed indietro (implementazioni fornite dal docente).

Passo 4. Si crei uno script `main.m` che

(1) carichi il dataset fornito dal docente tramite il comando `load dataset_chebtikonov`;
(2) definisca i parametri `n=12 lambda=1/M`, dove `M` è il numero di punti nel dataset, e calcoli utilizzando le precedenti functions `cstar` vettore dei coefficienti di $\tilde{f}_{\lambda,n}$;
(3) definisca, a partire da `cstar`, le tre anonymous functions `f,df,d2f` che preso in ingresso `x` calcolino rispettivamente $\tilde{f}_{\lambda,n}$, $\tilde{f}'_{\lambda,n}$, $\tilde{f}''_{\lambda,n}$ in `x`;
(4) chiami il metodo di Newton (usando la function vista a lezione e fornita dal docente) per approssimare un minimo di $\tilde{f}_{\lambda,n}$ approssimando uno zero di $\tilde{f}'_{\lambda,n}$ partendo da un valore iniziale ottenuto da una stima grafica (si settino gli altri parametri del metodo in modo sensato);
(5) stampi a video l'approssimazione del minimo di $\tilde{f}_{\lambda,n}$ così ottenuta, il valore di $\tilde{f}''_{\lambda,n}$ calcolata nel candidato punto di minimo (risulta davvero un minimo?), ed eventualmente delle figure utili ad illustrare i risultati.