Corso di Approssimazioni Numeriche, A.A. 2007/08

Esercitazione di laboratorio su:

Minimi quadrati e decomposizione SVD

Prof. S. De Marchi - 12 febbraio 2008

1 Equivalenza tra soluzione ai minimi quadrati ed SVD

Sia A una matrice rettangolare $m \times n$, $m \geq n$ che rappresenta la matrice di un problema di approssimazione di un insieme $X = \{(x_i, y_i),\ i = 1, ..., m\}$ con polinomi di grado $\leq n-1$, ovvero

$$\sum_{i=1}^{n} a_i x_j^{i-1} = y_j, \quad j = 1, ..., m \ . \tag{1}$$

Sappiamo che $A^T A$ è $n \times n$ simmetrica e semidefinita positiva. Usando il *metodo di Jacobi* per il calcolo di **tutti** gli autovalori di $A^T A$ possiamo determinare una matrice ortogonale U e una matrice diagonale D tale che

$$U^T (A^T A) U = D \ . \tag{2}$$

Nota: essendo $D = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, con gli autovalori in ordine decrescente, se per qualche i, λ_i è un numero negativo piccolo, lo si pone uguale a zero, poiché gli autovalori di $A^T A$ sono tutti positivi a meno di errori di arrotondamento dovuti al metodo di calcolo e alla precisone adottata.

Da (2), posto $B = AU$ $(m \times n)$, si ha che

$$B^T B = D \ . \tag{3}$$

Il che implica che le colonne di B sono ortogonali. Usando il metodo di fattorizzazione **QR** possiamo determinare una matrice ortogonale V tale che

$$V^T B = R \tag{4}$$

con R che è zero sotto la diagonale principale. Inoltre la matrice R è tale che

$$R^T R = B^T V^T V B = B^T B = D \ ;$$

che ci dice che le colonne di R sono ortogonali. Se qualche $\lambda_i = 0$ allora la corrispondente colonna di R sarà zero. Ora, poiché R è triangolare superiore ed è ortogonale, allora essa risulta essere zero anche sopra la diagonale principale. In definitiva R è diagonale ed ha la stessa forma della matrice F della decomposizione SVD di A (ricordiamo che $V^T A U = F$) cioè

$$R = \begin{pmatrix} \mu_1 & 0 & 0 & 0 & 0 \\ 0 & \mu_2 & 0 & 0 & 0 \\ \vdots & & \ddots & & \\ 0 & \cdots & & 0 & \mu_n \end{pmatrix} \tag{5}$$

Ora avremo che $R = F$ con $\mu_i = \sqrt{\lambda_i}$.

In (4), essendo $B = AU$, si ha che la decomposizione SVD richiesta è:

$$V^T A U = R \ .$$

1. **Vantaggio:** Semplicità di implementazione del metodo, una volta risolto il problema della ricerca degli autovalori di $A^T A$.

2. **Svantaggio:** Si deve fare il prodotto $A^T A$ che come noto può portare ad una perdita di informazioni ed ad un aumento del numero di condizionamento di A.

Esempio 1. `Esempio di data Fitting`. Si vuole determinare la funzione che approssima, nel senso dei minimi quadrati i punti $\{(x_i, y_i),\ 1 \le i \le m\}$, con un polinomio cubico $p_3(x) = a_1 + a_2 x + a_3 x^2 + a_4 x^3$. Questo è quello che si chiama `‘‘data fitting"`.
Per determinare i coefficienti $a_i,\ i = 1, ..., 4$ minimizzeremo l'`errore quadratico medio`

$$E(a_1, a_2, a_3, a_4) = \left(\frac{1}{m} \sum_{j=1}^{m} (y_j - p_3(x_j))^2 \right)^{\frac{1}{2}} \ .$$

Osserviamo che minimizzare E o E^2 è la stessa cosa. Ora, poichè E^2 è una funzione convessa, il minimo si troverà per $\frac{\partial E^2}{\partial a_i} = 0,\ \ i = 1, 2, 3, 4$. Ciò dà luogo ad un sistema lineare $A\,\mathbf{a} = \mathbf{y}$ con A, $m \times 4$ e i vettori $\mathbf{a}$, $\mathbf{y}$ che sono 4×1 e $m \times 1$, rispettivamente.
Vediamo il tutto in un caso concreto.

```
Si considerino i punti:

t=0:.05:1.0;
y=[.486; .866; .944;
   1.144; 1.103; 1.202;
   1.166; 1.191; 1.124;
   1.095; 1.122; 1.102;
   1.099; 1.017; 1.111;
   1.117; 1.152; 1.265;
   1.380; 1.575; 1.857];
```

Una possibile implementazione in Matlab/Octave della *decomposizione SVD di A* è allora come segue.

```
%-------------------------------------------------
% NB: Si usa la matrice A delle equazioni normali
%-------------------------------------------------
%--- costruisco A
for i=1:length(t),
 for j=1:4,
   a(i,j)=t(i)^(j-1);
```

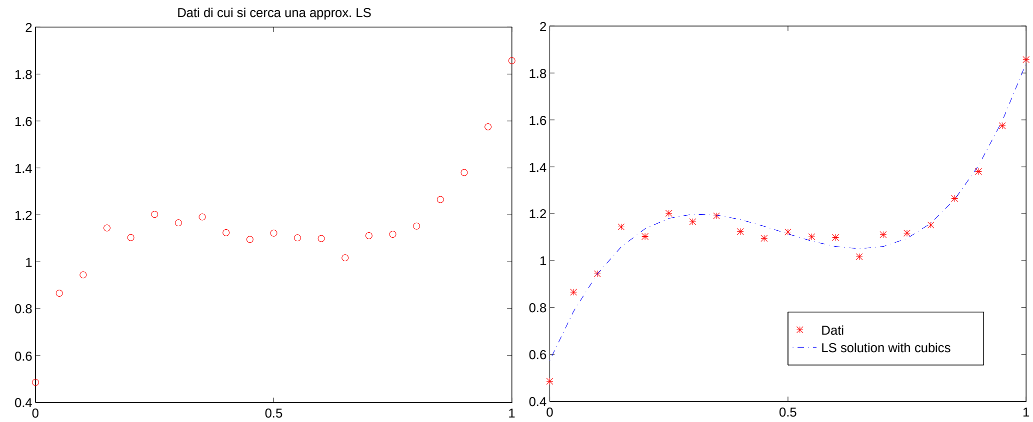

Figure 1: (Sx) dati da approssimare. (Dx) Approssimante cubica ai minimi quadrati

```
 end;
end;
a1=a'*a; [u,d]=eig(a1); b=a*u; [v,r]=qr(b);

z=inv(r'*r)*r'*(v'*y);

disp('Soluzione con la decomposizione SVD di A ');
x=u*z
```

Eseguendo il codice ecco il risulatati

```
>> Soluzione con la decomposizione SVD di A
    0.5747
    4.7259
  -11.1282
    7.6687
```

2 Esercitazione proposta

1. Ripetere l'esempio precedente usando la decomposizione `svd` della matrice A (*Sugg.* usare la chiamata `[U,S,V]=svd(A,'econ')`)

2. Il primo esercizio è un classico problema di data fitting. Si considerino i valori di tabella

x_i	1	2.5	3	5	6.5	8	9.3
y_i	4	2	3	3.5	3.9	7	5.3

   - determinare il polinomio P_m, di grado $m = 3$ approssimante le coppie di valori (x_i, y_i) nel senso dei minimi quadrati discreti.

- Si giustifichi il fatto che per $m = 6$ il polinomio è interpolante.
- Si consideri il punto $\bar{x} = 4$ e come valore corrispondente $\bar{y}$, quello dell'interpolante lineare sull'intervallo [3,5]. Si ora $|P_m(\bar{x}) - \bar{y}|$ l'errore assoluto in $\bar{x}$. Far vedere che per $m = 2$ l'errore è minimo.

3. Il secondo esercizio riguarda la `compressione di immagini`. Usando il comando Matlab/Octave, `imread`, si legga l'immagine `pout.tif` (che si trova nell'Image Processing Toolbox di Matlab), `A=imread('pout.tif')`, che è un'immagine in bianco e nero. In questo caso `A` è una matrice 291×240 di interi 8 bit (`unit8`) che ci danno le intensità di grigio.

   Per applicare la SVD ad `A`, dobbiamo dapprima convertire `A` in una matrice di numeri in doppia precisione (che sono i numeri usualmente trattati da Matlab): `A=double(A)`.

   - Usare la decomposizone SVD di A per ottenere solo alcuni dei valori singolari di `A` per ricostruire in maniera "approssimata" l'immagine `pout.tif`. Per visualizzare l'immagine originale e la sua approssimazione, usare il comando `imshow(uint8(X))` dove `X` indica l'immagine approssimata.

`Tempo assegnato:` 2 ore.

3 Approffondimenti

Questa volta non vengono forniti approffondimenti vista la natura "classica" dell'esercitazione.