Laboratorio di Calcolo Numerico: Algebra lineare numerica

Giacomo Elefante

Laboratorio di calcolo numerico
23/11/23

Scopo di oggi

Analizzare algoritmi per l'algebra lineare numerica:

- Sistemi sovradeterminati
- Soluzioni di minimi quadrati con le equazioni normali
- Decomposizione QR
- Soluzione di minimi quadrati con QR
- Decomposizione SVD e compressione di immagini

Nel caso di sistemi sovradeterminati, ovvero $Ax = b$ con $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ e $x \in \mathbb{R}^n$, con $m > n$. In generale non esiste una soluzione x tale che $Ax = b$, si cerca una soluzione tale che il residuo $\|Ax - b\|_2$ sia minima.
Una soluzione del problema ai minimi quadrati può essere data dalla soluzione del problema alle equazioni normali

$$A^t Ax = A^t b$$

Esercizio

Esercizio

Si crei uno script dove si cerca il polinomio di grado $n = 1$ (la retta di regressione) che meglio approssimi dei dati perturbati della retta $y = 2x + 0.2$, ottenuti aggiungendo ai dati dei valori random ottenuti con la funzione Matlab `rand`, ovvero

```
y = 2 x + 0.2 + 0.1*rand(size(x))
```

si prendano come ascisse, un vettore `x` di 20 punti equidistanti in $[0, 1]$.
Per generare la matrice di Vandermonde relativa ai nodi `x`, si usi il comando

```
V = fliplr(vander(x));
```

Tale comando genera la matrice (quadrata) di Vandermonde dove ogni elemento è $\mathtt{V}_{i,j} = \mathtt{x}_i^j$. Volendo il polinomio di grado 1 che meglio approssima i dati si definisca la matrice `A` del problema ai minimi quadrati, prendendo solo le prime due colonne della matrice `V`.
Si trovino quindi i coefficienti della retta, ovvero la soluzione (attraverso le equazioni normali) del problema ai minimi quadrati generato , e successivamente si faccia il plot sovrapponendo la retta di regressione in blu e i punti $(\mathtt{x}, \mathtt{y})$ in pallini verdi con bordo nero.

Il condizionamento della matrice A (e anche di A^tA nel caso del problema ai minimi quadrati) può portare a non trovare la soluzione corretta del poblema.

Esercizio

Si ripeta l'esercizio con `lu_solver` ma usando la matrice $\texttt{A} = \texttt{magic}(10)$ e come b il risultato di $\texttt{b} = \texttt{A} * [1:10]'$, (da cui si ha ovviamente che la soluzione esatta è $[1:10]'$)

Per migliorare si può utilizzare la decomposizione QR della matrice. Infatti data $A \in \mathbb{R}^{m\times n}$ tale che $\texttt{rank}(A) = n$, allora esistono $Q \in \mathbb{R}^{m\times n}$ (ortogonale) e $R \in \mathbb{R}^{n\times n}$ (triangolare superiore non singolare) tali che

$$A = QR.$$

Da questo si può dimostrare che il sistema alle equazioni normali può essere trasformato (attraverso la decomposizione QR di A^tA) nel sistema

$$Rx = Q^tb$$

che è risolvibile facilmente con il meg ma, sopratutto, per cui si ha che

$$\kappa(A^tA) = (\kappa(A))^2 >> \kappa(R)$$

Attenzione che utilizzando il comando per la decomposizione QR di Matlab `qr` su $A \in \mathbb{R}^{m \times n}$ viene decomposta in due matrici Q, R la prima ortogonale e la seconda triangolare superiore di dimensioni $Q \in \mathbb{R}^{m \times m}$, $R \in \mathbb{R}^{m \times n}$, che possiamo descrivere come

$$Q = [Q_1 Q_2]$$

$Q_1 \in \mathbb{R}^{m \times n}$ e $Q_2 \in \mathbb{R}^{m \times (m-n)}$ e

$$R = \begin{bmatrix} R1 \\ 0 \end{bmatrix}$$

con $R_1 \in \mathbb{R}^{n \times n}$ e $0 \in \mathbb{R}^{(m-n) \times n}$, da cui le matrici che servono per risolvere il sistema lineare sono Q_1 e R_1.
Altrimenti basta utilizzare la qr economy, con il comando `qr(A,0)`, oppure `qr(A,'econ')` che resistuisce in output direttamente la Q_1 e la R_1 richieste.

Esercizio

Esercizio

Data la funzione $f(x) = sin(2x) - x^2$, si costruisca un vettore `x` di `n` $= 100$ punti in $[-5, 5]$, dove valutare la funzione in un vettore `y`, a cui possiamo aggiungere un rumore aggiungendo $0.5 * $ `rand(size(y))`. In seguito si definisca un grado `d` $= 8$ e si vada a cercare il polinomio di grado `d` che meglio approssima ai minimi quadrati i valori `y` in `x`, ovvero che i suoi coefficienti `a` $= [a_0, \ldots, a_d]$ cerchino (il meglio possibile) la soluzione del sistema

$$Va = y$$

con V la matrice di Vandermonde, ovvero, $(V)_{i,j} = x_i^j$.
Si esegua l'esercizio sia andando a risolverlo con le equazioni normali e sia con la decomposizione QR.
Si stampino a schermo il condizionamento della matrice A^tA e della matrice R.
E infine si facciano due grafici, uno sovrapponendo la funzione f al polinomio approssimante trovato attraverso le equazioni normali e un secondo sovrapponendo la funzione f al polinomio approssimante trovato attraverso la decomposizione QR.

La Singular Value Decomposition (SVD)

Sia $A \in \mathbb{R}^{m \times n}$. Allora possiamo sempre scrivere

$$A = USV^t$$

dove $U \in \mathbb{R}^{m \times m}$ e $V \in \mathbb{R}^{n \times n}$ sono matrici ortogonali e S una matrice diagonale $m \times n$.
Questa decomposizione è detta singular value decomposition perchè i valori della diagonale di S sono detti valori singolari.

La Singular Value Decomposition (SVD)

- Il rango di A è uguale al rango di S
- I valori singolari sono tutti non negativi e si ordinano in modo decrescente, i.e., $s_1 \geq s_2 \geq \cdots \geq s_m \geq 0$
- I valori singolari sono le radici degli autovalori di AA^t (che è simmetrica, per cui può essere diagonalizzata per il teorema spettrale).
- I valori singolari possono essere interpretati come il "concentrato dell'informazione" della matrice, i primi sono i più importanti, mentre gli ultimi quelli meno importanti di cui volendo possiamo provare a fare a meno. Questa è la filosofia della Principal Component Analysis (PCA), che vediamo nell'esperimento.

La Singular Value Decomposition (SVD)

In un nuovo script, si definisca una variabile k e le si assegni il valore 15.
Successivamente, si carichi in Matlab come matrice l'immagine peppers.png che si trova su moodle con i comandi (salvandola prima nella directory)

```
I = imread('peppers.png');
I = double(I);
```

Esercizio

Si calcoli la SVD della matrice I (usare il comando help svd per info sulla struttura).
Successivamente, si plottino in scala semilogaritmica i valori singolari ottenuti (la diagonale di S).
Si ricostruisca l'immagine ricomponendo la SVD ma

- Tenendo solo i primi k valori singolari e azzerando gli altri.
- Azzerando i primi k valori singolari e tenendo gli altri.
- Azzerando gli ultimi k valori singolari e tenendo gli altri.

Per tutte e tre le diverse modifiche, una volta ricomposta la matrice (chiamiamola I_compressa) possiamo salvarla come immagine scrivendo

```
I_compressa = uint8(I_compressa);
imwrite(I_compressa,'nome_img.png');
```

Come cambia l'immagine ricostruita nei diversi modi proposti?

Esercizio (per casa)

Esercizio

Si costruisca una function `ls_matrix.m` (senza usare la function `vander`) che, dato un vettore riga di punti `x` $= (x_1, \ldots, x_n)$ ed un grado $d \leq n-1$, costruisce la matrice

$$V = \begin{pmatrix} x_1^d & x_1^{d-1} & \ldots & x_1^0 \\ x_2^d & x_2^{d-1} & \ldots & x_2^0 \\ \vdots & \vdots & & \vdots \\ x_n^d & x_n^{d-1} & \ldots & x_n^0 \end{pmatrix}$$

La function deve avere come input il vettore e il grato e come output la matrice V.