

Minimi quadrati, autovalori e valori singolari

Claudio Canuto
Dipartimento di Scienze Matematiche - Politecnico di Torino

`claudio.canuto@polito.it`

Indice

- 1 Sistemi sovra-determinati e minimi quadrati
- 2 Autovalori e autovettori
- 3 La decomposizione ai valori singolari di una matrice

4. Sistemi sovra-determinati e minimi quadrati

Un esempio

Il consumo di carburante C di un'auto dipende dalla velocità V in modo bi-monotono: inizialmente decresce, poi cresce (Fig. sinistra). In prima approssimazione, possiamo ipotizzare una dipendenza quadratica, del tipo

$$C = \alpha_0 + \alpha_1 V + \alpha_2 V^2.$$

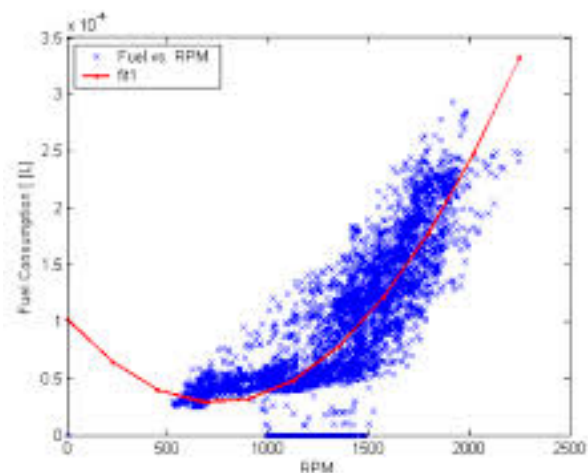
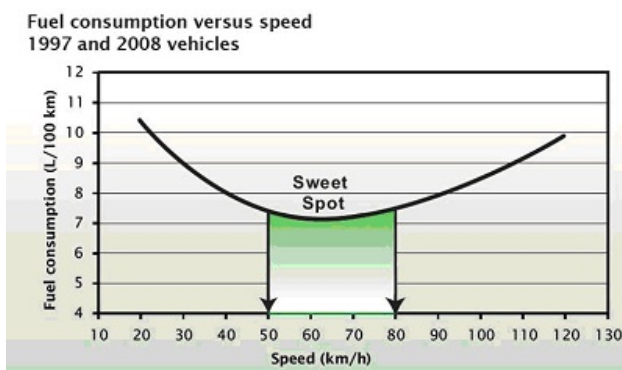


Fig. 8. Fuel Consumption vs. RPM

Una legge di tipo quadratico potrebbe descrivere anche la dipendenza di C dal numero di giri del motore (Fig. destra).

Se la legge fosse effettivamente quadratica, e se le misurazioni fossero esatte, basterebbero tre misurazioni diverse per individuare i coefficiente della parabola:

$$\begin{aligned}\alpha_0 + \alpha_1 V_1 + \alpha_2 V_1^2 &= C_1 \\ \alpha_0 + \alpha_1 V_2 + \alpha_2 V_2^2 &= C_2 \\ \alpha_0 + \alpha_1 V_3 + \alpha_2 V_3^2 &= C_3\end{aligned}$$

ossia

$$\begin{pmatrix} 1 & V_1 & V_1^2 \\ 1 & V_2 & V_2^2 \\ 1 & V_3 & V_3^2 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix}.$$

In realtà così non è, e quindi effettuiamo parecchie misurazioni cercando i coefficienti di quella parabola che meglio descrive il comportamento osservato:

$$\alpha_0 + \alpha_1 V_i + \alpha_2 V_i^2 = C_i, \quad 1 \leq i \leq m, \quad (\text{con } m \gg 3).$$

Otteniamo in questo modo un **sistema sovra-determinato**.

- Cosa intendiamo per *soluzione* di un tale sistema?
- Come la calcoliamo?

Sistemi sovra-determinati

Scriviamo (formalmente) un generico sistema sovra-determinato come

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad 1 \leq i \leq m, \quad (\text{con } m > n),$$

ossia

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

con \mathbf{A} matrice rettangolare avente m righe ed n colonne, \mathbf{x} vettore colonna di ordine n , e \mathbf{b} vettore colonna di ordine m .

- Nel seguito, **supponiamo che la matrice \mathbf{A} abbia rango massimo n** .

Ciò significa che i suoi n vettori colonna sono *linearmente indipendenti*.

Equivalentemente, dalla matrice \mathbf{A} è possibile estrarre una sottomatrice quadrata $\tilde{\mathbf{A}}$ di ordine n *non-singolare*. Dunque, esistono n equazioni del sistema linearmente indipendenti fra loro.

Per un termine noto \mathbf{b} generico, non possiamo aspettarci che esista un vettore \mathbf{x} tale che l'uguaglianza $\mathbf{Ax} = \mathbf{b}$ sia soddisfatta. Ma se ciò accade, allora il vettore $\mathbf{Ax} - \mathbf{b}$ è nullo.

Possiamo dunque cercare di *minimizzare lo scarto tra i vettori \mathbf{Ax} e \mathbf{b}* .

Per essere chiari, per ogni vettore $\mathbf{y} \in \mathbb{R}^n$ consideriamo il vettore

$$\mathbf{r}(\mathbf{y}) = \mathbf{b} - \mathbf{Ay},$$

che chiamiamo il *residuo* dell'equazione associato a \mathbf{y} .

Noi *vogliamo rendere piccolo il residuo*, e pertanto introduciamo una *norma* $\|\cdot\|$ in \mathbb{R}^m al fine di misurare la grandezza di tale vettore.

Diciamo che $\mathbf{x} \in \mathbb{R}^n$ è soluzione del sistema sovra-determinato, rispetto a tale norma, se

$$\|\mathbf{r}(\mathbf{x})\| = \min_{\mathbf{y} \in \mathbb{R}^n} \|\mathbf{r}(\mathbf{y})\|.$$

vale a dire

$$\|\mathbf{Ax} - \mathbf{b}\| = \min_{\mathbf{y} \in \mathbb{R}^n} \|\mathbf{Ay} - \mathbf{b}\|.$$

Osservazione. Se il sistema è quadrato, $m = n$, ritroviamo in tal modo la soluzione "classica" del sistema $\mathbf{Ax} = \mathbf{b}$, in quanto si ha

$$\|\mathbf{Ax} - \mathbf{b}\| = 0.$$

Il problema di minimizzare il residuo può essere risolto facilmente, come vedremo, se si sceglie la norma euclidea.

Se $\mathbf{x} \in \mathbb{R}^n$ soddisfa

$$\|\mathbf{Ax} - \mathbf{b}\|_2 = \min_{\mathbf{y} \in \mathbb{R}^n} \|\mathbf{Ay} - \mathbf{b}\|_2$$

diciamo che \mathbf{x} è soluzione del sistema sovra-determinato **nel senso dei minimi quadrati**.

In tal caso, \mathbf{x} è il vettore che minimizza lo *scarto quadratico medio*

$$\left(\frac{1}{m} \sum_{i=1}^m |(\mathbf{Ay})_i - b_i|^2 \right)^{1/2}$$

tra le componenti dei vettori \mathbf{Ay} e \mathbf{b} .

Osservazione. Se si usa la norma 1, si ha il problema di minimo

$$\sum_{i=1}^m |(\mathbf{Ax})_i - b_i| = \min_{\mathbf{y} \in \mathbb{R}^n} \sum_{i=1}^m |(\mathbf{Ay})_i - b_i|$$

mentre se si usa la norma ∞ si ha

$$\max_{1 \leq i \leq m} |(\mathbf{Ax})_i - b_i| = \min_{\mathbf{y} \in \mathbb{R}^n} \max_{1 \leq i \leq m} |(\mathbf{Ay})_i - b_i|.$$

Tali problemi possono essere riformulati come problemi di *programmazione lineare*, e in genere sono computazionalmente più onerosi rispetto al problema dei minimi quadrati.

Variando tra le norme 1, 2 e ∞ , aumenta il peso che si dà alle componenti in cui si ha lo scarto maggiore tra $(\mathbf{Ay})_i$ e b_i .

Vogliamo ora mostrare che il problema dei minimi quadrati ammette una e una sola soluzione, e caratterizzarla anche come soluzione di un sistema quadrato di n equazioni in n incognite.

Osserviamo che minimizzare la quantità $\|A\mathbf{y} - \mathbf{b}\|_2$ equivale a minimizzare la quantità

$$\Phi(\mathbf{y}) = \frac{1}{2} \|A\mathbf{y} - \mathbf{b}\|_2^2 = \frac{1}{2} (A\mathbf{y} - \mathbf{b})^T (A\mathbf{y} - \mathbf{b}).$$

e dunque \mathbf{x} soddisfa

$$\Phi(\mathbf{x}) = \min_{\mathbf{y} \in \mathbb{R}^n} \Phi(\mathbf{y}).$$

Per capire come è fatto il funzionale $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}_+$, diamo un generico incremento $\delta\mathbf{y}$ all'argomento \mathbf{y} . Con semplici passaggi (esercizio) si ottiene

$$\begin{aligned} \Phi(\mathbf{y} + \delta\mathbf{y}) &= \Phi(\mathbf{y}) + (A\mathbf{y} - \mathbf{b})^T (A\delta\mathbf{y}) + \frac{1}{2} (A\delta\mathbf{y})^T (A\delta\mathbf{y}) \\ &= \Phi(\mathbf{y}) + (A^T A\mathbf{y} - A^T \mathbf{b})^T \delta\mathbf{y} + \frac{1}{2} \delta\mathbf{y}^T (A^T A) \delta\mathbf{y}. \end{aligned}$$

Da ciò deduciamo che

$$\text{grad } \Phi(\mathbf{y}) = A^T A\mathbf{y} - A^T \mathbf{b}, \quad \text{Hess } \Phi(\mathbf{y}) = A^T A.$$

Proprietà. Sia A una matrice di ordine $m \times n$ con $m \geq n$, avente rango massimo. Allora, la matrice $A^T A$ è simmetrica e definita positiva.

Infatti, $(A^T A)^T = A^T (A^T)^T = A^T A$, dunque la matrice è simmetrica. Inoltre, per ogni $\mathbf{y} \in \mathbb{R}^n$

$$\mathbf{y}^T (A^T A) \mathbf{y} = (A\mathbf{y})^T (A\mathbf{y}) = \|A\mathbf{y}\|_2^2 \geq 0,$$

e $\|A\mathbf{y}\|_2 = 0$ implica $A\mathbf{y} = \mathbf{0}$, che a sua volta implica $\mathbf{y} = \mathbf{0}$ in quanto la matrice A ha rango massimo (i vettori colonna sono linearmente indipendenti).

In base a questi risultati, se poniamo uguale a zero il gradiente di Φ ,

$$\text{grad } \Phi(\mathbf{x}) = A^T A\mathbf{x} - A^T \mathbf{b} = \mathbf{0},$$

otteniamo l'unico punto di minimo (stretto) di Φ , in quanto dalla formula della slide precedente si ha

$$\Phi(\mathbf{x} + \delta\mathbf{x}) = \Phi(\mathbf{x}) + \frac{1}{2} \delta\mathbf{x}^T (A^T A) \delta\mathbf{x} > \Phi(\mathbf{x})$$

per ogni $\delta\mathbf{x} \neq \mathbf{0}$.

Abbiamo dunque stabilito la seguente

Proprietà. La soluzione del problema dei minimi quadrati coincide con la soluzione del sistema quadrato di ordine n

$$\mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{b}$$

detto sistema delle **equazioni normali**.

Possiamo dunque *calcolare \mathbf{x} risolvendo le equazioni normali*. Ciò ha dei *pro* e dei *contro*:

- La matrice $\mathbf{A}^T \mathbf{A}$ è facilmente calcolabile. Essendo simmetrica e definita positiva, possiamo applicare il metodo di Choleski `R = chol(A'*A)` per fattorizzarla.
- Il sistema delle equazioni normali può essere *fortemente malcondizionato*. Infatti il numero di condizionamento della matrice $\mathbf{A}^T \mathbf{A}$ è *il quadrato* del numero di condizionamento della matrice \mathbf{A} .

Per calcolare la soluzione \mathbf{x} in modo numericamente più stabile, è preferibile seguire un'altra via, ricorrendo alla *fattorizzazione QR della matrice \mathbf{A}* .

Ritorniamo sulla fattorizzazione QR di una matrice...

La fattorizzazione QR esiste anche per le matrici \mathbf{A} , non quadrate, di ordine $m \times n$. L'algoritmo che la produce è del tutto simile a quello descritto in precedenza per matrici quadrate.

- Caso $m > n$:

$$\mathbf{Q} = \begin{pmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{pmatrix} \quad \mathbf{R} = \begin{pmatrix} * & * & * \\ 0 & * & * \\ 0 & 0 & * \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

- Caso $m < n$:

$$\mathbf{Q} = \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix} \quad \mathbf{R} = \begin{pmatrix} * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \end{pmatrix}$$

Se la matrice \mathbf{A} ha rango massimo, gli elementi che stanno sulla diagonale principale di \mathbf{R} sono tutti $\neq 0$.

Mediante la fattorizzazione QR della nostra matrice A (di ordine $m \times n$ con $m > n$), possiamo dare una formulazione equivalente al problema di minimo

$$\|Ax - b\|_2 = \min_{y \in \mathbb{R}^n} \|Ay - b\|_2.$$

- Ricordando che Q è una matrice ortogonale ($Q^T Q = Q Q^T = I$), scriviamo

$$Ay - b = QRy - b = QRy - QQ^T b = Q(Ry - Q^T b) = Q(Ry - c)$$

avendo posto $c = Q^T b$.

- Ricordando che Q conserva la norma euclidea ($\|Qz\|_2 = \|z\|_2$ per ogni z), abbiamo

$$\|Ay - b\|_2 = \|Q(Ry - c)\|_2 = \|Ry - c\|_2.$$

Il problema di minimo si scrive quindi equivalentemente come

$$\|Ry - c\|_2 = \min_{y \in \mathbb{R}^n} \|Ry - c\|_2.$$

Decomponiamo la matrice rettangolare R nella forma a blocchi

$$R = \begin{pmatrix} \tilde{R} \\ O \end{pmatrix}$$

con \tilde{R} matrice quadrata di ordine n , triangolare superiore e con elementi diagonali tutti diversi da 0, e O matrice nulla di ordine $(m - n) \times n$.

Dunque

$$Ry = \begin{pmatrix} \tilde{R}y \\ 0 \end{pmatrix}.$$

Similmente, decomponiamo il vettore c nella forma a blocchi

$$c = \begin{pmatrix} \tilde{c} \\ \tilde{d} \end{pmatrix}$$

con $\tilde{c} \in \mathbb{R}^n$ e $\tilde{d} \in \mathbb{R}^{m-n}$.

Si ha quindi

$$Ry - c = \begin{pmatrix} \tilde{R}y - \tilde{c} \\ -\tilde{d} \end{pmatrix}.$$

Prendendo il quadrato della norma euclidea di tale vettore, otteniamo

$$\|Ry - c\|_2^2 = \|\tilde{R}y - \tilde{c}\|_2^2 + \|\tilde{d}\|_2^2.$$

Notiamo che il secondo addendo a secondo membro non dipende da y , mentre il primo addendo può essere reso $= 0$ scegliendo come y la soluzione x del sistema lineare quadrato

$$\tilde{R}x = \tilde{c}.$$

Abbiamo dunque ottenuto il seguente risultato.

Proprietà. La soluzione del problema dei minimi quadrati

$$\|Ax - b\|_2 = \min_{y \in \mathbb{R}^n} \|Ay - b\|_2.$$

coincide con la soluzione del sistema quadrato di ordine n

$$\tilde{R}x = \tilde{c}.$$

Inoltre si ha

$$\|Ax - b\|_2 = \|\tilde{d}\|_2.$$

Come risolvere il problema dei minimi quadrati

In pratica, i passi necessari per calcolare la soluzione x del problema dei minimi quadrati sono i seguenti:

- calcolare i fattori Q ed R della matrice A ;
- isolare \tilde{R} , il blocco quadrato superiore di ordine n della matrice R ;
- eseguire la moltiplicazione $c = Q^T b$;
- isolare \tilde{c} , la parte superiore di ordine n del vettore c ;
- risolvere il sistema lineare $\tilde{R}x = \tilde{c}$ mediante sostituzione all'indietro.

Comandi MATLAB

La fattorizzazione QR di una matrice rettangolare A si ottiene ancora con il comando Matlab `[Q R] = qr(A)`.

Si noti però che per ottenere \tilde{c} è sufficiente moltiplicare le prime n righe di Q^T per il vettore b . In altri termini, per risolvere il problema dei minimi quadrati, *si usano solo le prime n colonne della matrice Q (così come le prime n righe della matrice R).*

Per questo, **Matlab** ha predisposto il comando `[Q R] = qr(A,0)`, che fornisce la *forma economica* della fattorizzazione QR , ossia precisamente le prime n colonne della matrice Q e le prime n righe della matrice R .

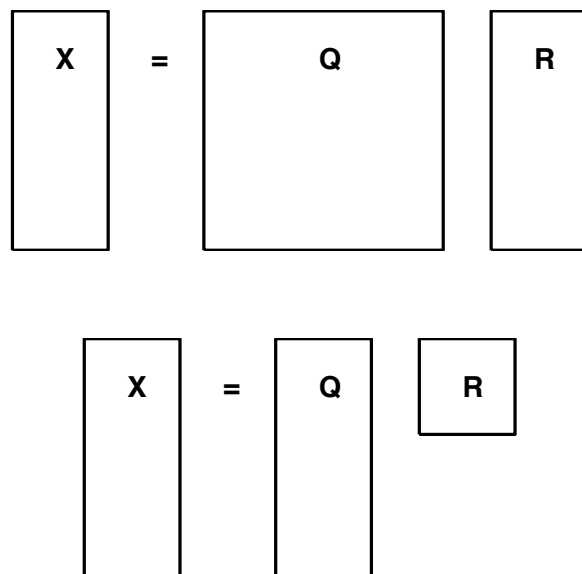


Figure: Forma piena ed economica della fattorizzazione QR di una matrice X (da C. Moler)

Osservazione. Il sistema sovra-determinato $Ax = b$, con A matrice di rango massimo, può essere direttamente risolto mediante il comando Matlab `x=A\b`, che calcola la soluzione nel senso dei minimi quadrati descritta sopra.

Il metodo dei minimi quadrati pesati

Talvolta, è opportuno attribuire dei pesi diversi alle varie equazioni che compongono il sistema sovra-determinato: alcune equazioni sono più importanti, e quindi devono essere risolte “meglio”, altre sono meno significative, e quindi possiamo accettare un errore maggiore su di esse.

Per tenere conto di ciò, si introducono m pesi $w_i > 0$, $1 \leq i \leq m$, e si minimizza lo *scarto quadratico pesato*

$$\left(\sum_{i=1}^m |(Ay)_i - b_i|^2 w_i \right)^{1/2}$$

Equivalentemente, posto $W = \text{diag}(w_1, w_2, \dots, w_m)$, si minimizza la quantità

$$\frac{1}{2} (Ay - b)^T W (Ay - b).$$

Il problema dei minimi quadrati pesati può ancora essere risolto mediante una delle tecniche illustrate precedentemente (*equazioni normali* oppure *fattorizzazione QR*): è sufficiente sostituire alla matrice A e al vettore b rispettivamente la matrice $W^{1/2}A$ e il vettore $W^{1/2}b$, dove $W^{1/2} = \text{diag}(\sqrt{w_1}, \sqrt{w_2}, \dots, \sqrt{w_m})$.

5. Calcolo di autovalori e autovettori

Un esempio

Molte situazioni di interesse portano alla necessità di calcolare autovalori di una matrice quadrata.

- Consideriamo un sistema meccanico di N masse mobili collegate tra loro da molle. Sia m_i la massa della particella che al tempo t occupa la posizione $\mathbf{x}_i = \mathbf{x}_i(t)$ nello spazio, e sia $k_{ij} \geq 0$ la costante di rigidità della molla che collega la massa m_i con la massa m_j ($k_{ij} = 0$ se le due masse non sono collegate da una molla).

Allora, l'evoluzione nel tempo delle masse è descritta dal sistema di equazioni differenziali del secondo ordine

$$m_i \mathbf{x}_i'' = \sum_{j=1}^N k_{ij} (\mathbf{x}_j - \mathbf{x}_i) \quad 1 \leq i \leq N.$$

- Indicato con \mathbf{x} il vettore di dimensione $n = 3N$ che raccoglie le coordinate di tutte le masse, tale sistema può essere scritto in forma compatta come

$$\mathbf{x}'' = \mathbf{A}\mathbf{x}.$$

- Se ora cerchiamo le *oscillazioni libere* del sistema, ossia i *moti periodici* descritti dalla legge

$$\mathbf{x}(t) = e^{i\omega t} \mathbf{w}$$

con ω reale e $\mathbf{w} \neq \mathbf{0}$, vediamo facilmente che \mathbf{w} soddisfa

$$\lambda \mathbf{w} = \mathbf{A} \mathbf{w}, \quad \text{con } \lambda = \omega^2.$$

Dunque *esistono moti periodici se e solo se la matrice \mathbf{A} ammette autovalori λ reali e non negativi.*

Per ogni tale autovalore, la *frequenza di oscillazione* del sistema è data da

$$\phi_\lambda = \frac{\sqrt{\lambda}}{2\pi}.$$

- Possiamo essere interessati a calcolare *tutte le frequenze*, oppure *solo alcune* (ad esempio, quella più bassa, o quella più vicina a una frequenza assegnata).

Il condizionamento del problema del calcolo degli autovalori

Supponiamo che \mathbf{A} sia una matrice reale quadrata di ordine n , diagonalizzabile.

Siano λ_p , $p = 1, 2, \dots, n$, i suoi autovalori e sia \mathbf{W} la matrice non-singolare le cui colonne sono i corrispondenti autovettori \mathbf{w}_p .

Proprietà. Sia $\tilde{\mathbf{A}}$ una perturbazione di \mathbf{A} , e sia $\tilde{\lambda}$ un suo autovalore. Allora

$$\min_{1 \leq p \leq n} |\tilde{\lambda} - \lambda_p| \leq \text{cond}(\mathbf{W}) \|\tilde{\mathbf{A}} - \mathbf{A}\|.$$

Dunque, il condizionamento del problema del calcolo degli autovalori dipende *non* dal numero di condizionamento della matrice \mathbf{A} bensì da quello della *matrice degli autovettori*.

Se \mathbf{A} è simmetrica, allora \mathbf{W} è ortogonale e quindi $\text{cond}_2(\mathbf{W}) = 1$. In tal caso, *il problema è sempre bencondizionato*.

Possiamo a grandi linee suddividere gli algoritmi di calcolo di autovalori (e autovettori) di una matrice quadrata in due famiglie:

- 1 Algoritmi che calcolano simultaneamente *tutti gli autovalori* della matrice (ad esempio, il metodo QR);
- 2 Algoritmi mirati che calcolano *autovalori aventi specifiche proprietà* (ad esempio, il metodo della potenza, o quello della potenza inversa con shift).

In genere, gli algoritmi sono di tipo *iterativo*, cioè generano una successione (di scalari, o di vettori, o di matrici) che converge verso un limite, che è la quantità cercata, o dal quale si può facilmente ottenere la quantità cercata.

Si pone quindi il problema della *convergenza* di tale successione.

Trasformazioni di similitudine

I metodi globali (che calcolano simultaneamente tutti gli autovalori) si basano su opportune *trasformazioni di similitudine*

$$A \rightarrow B = S^{-1}AS$$

(con S invertibile), le quali *lasciano invariati gli autovalori della matrice*.

Infatti, se λ è autovalore di A con autovettore w ,

$$Aw = \lambda w,$$

allora si ha

$$S^{-1}Aw = \lambda S^{-1}w,$$

che equivale a

$$S^{-1}ASS^{-1}w = \lambda S^{-1}w,$$

da cui

$$Bz = \lambda z \quad \text{con } z = S^{-1}w.$$

- Di particolare importanza sono le trasformazioni di similitudine generate da una *matrice ortogonale* Q ,

$$A \rightarrow B = Q^T A Q.$$

Riduzione preliminare a forma di Hessenberg

Combinando opportune trasformazioni di Householder, è possibile costruire una matrice ortogonale Q_H tale che $B = Q_H^T A Q_H$ sia in forma di **Hessenberg**, ossia tale che tutti i suoi elementi posti al di sotto della prima sotto-diagonale siano nulli.

$$B = \begin{pmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{pmatrix}$$

Se A è *simmetrica*, anche B risulta *simmetrica*, e dunque *tridiagonale*.

$$B = \begin{pmatrix} * & * & 0 & 0 & 0 \\ * & * & * & 0 & 0 \\ 0 & * & * & * & 0 \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{pmatrix}$$

Il **comando MATLAB** che realizza la trasformazione di Hessenberg è `B=hess(A)`.

La riduzione a forma di Hessenberg costituisce un indispensabile *preprocessing* al fine di ridurre il costo delle successive trasformazioni per il calcolo degli autovalori.

Il metodo QR (cenni)

Il **metodo QR** (e le sue molte varianti) è la procedura generalmente più efficiente e diffusa per il calcolo simultaneo di tutti gli autovalori di una matrice quadrata A .

Partendo dalla matrice $A = A_0$ (preliminarmente ridotta in forma di Hessenberg per maggiore efficienza), l'algoritmo genera una successione di matrici A_k , $k = 1, 2, \dots$, simili ad A , con la seguente legge:

$$\begin{aligned} A_k &=: Q_k R_k, \\ A_{k+1} &:= R_k Q_k. \end{aligned}$$

Proprietà:

- A_{k+1} è simile ad A_k : infatti

$$Q_k^T A_k Q_k = Q_k^T Q_k R_k Q_k = R_k Q_k = A_{k+1}.$$

- Se A è in forma di Hessenberg, tutte le A_k lo sono.
- Se A è simmetrica, tutte le A_k lo sono.

Sotto ipotesi opportune, *la successione di matrici A_k , $k = 1, 2, \dots$, converge per $k \rightarrow \infty$ verso una matrice limite A_∞* con le seguenti proprietà:

- se la matrice A è simmetrica, allora A_∞ è *diagonale*; gli elementi sulla diagonale sono gli autovalori di A ;
- se la matrice A non è simmetrica ma ha autovalori tutti reali, allora A_∞ è *triangolare superiore*; gli elementi sulla diagonale sono gli autovalori di A ;
- se la matrice A non è simmetrica e ha alcuni autovalori complessi (coniugati), allora A_∞ è *quasi triangolare superiore*, ossia triangolare superiore a blocchi; precisamente, i blocchi diagonali sono o matrici 1×1 contenenti gli autovalori reali di A , oppure matrici 2×2 , i cui autovalori sono una coppia di autovalori complessi coniugati di A .

Gli autovettori di A possono essere calcolati a partire dalle matrici di trasformazione Q_k .

Osservazioni.

- L'implementazione effettiva dell'algoritmo è ad oggi molto sofisticata. L'illustrazione fatta nelle slides precedenti vuole solo dare un'idea generale dei fondamenti del metodo.
- Ad esempio, per accelerare la convergenza, si includono delle opportune *traslazioni* (*shifts*) degli autovalori, ossia il passo k -esimo può diventare

$$\begin{aligned} A_k - s_k I &= Q_k R_k, \\ A_{k+1} &:= R_k Q_k + s_k I, \end{aligned}$$

con s_k scalari opportunamente scelti.

- Non tutti gli autovalori convergono con la stessa velocità. Se un autovalore è giunto a convergenza (rispetto a una tolleranza prefissata), è possibile “toglierlo dalla matrice”, continuando l'algoritmo con una matrice di ordine ridotto (processo di *deflazione*).
- Il metodo può non convergere. Ad esempio, cosa succede se la matrice A è ortogonale...?
- I **comandi MATLAB** sono `d=eig(A)` per ottenere un vettore d contenente gli autovalori di A , e `[V D]=eig(A)` per ottenere una matrice V le cui colonne sono gli autovettori e una matrice diagonale D contenente gli autovalori di A .

Il prototipo degli algoritmi “mirati” al calcolo di specifici autovalori di una matrice \mathbf{A} è il **metodo della potenza**.

Esso genera una successione di numeri che, sotto opportune ipotesi, converge verso *l'autovalore di modulo massimo* della matrice, e una successione di vettori che converge verso il *corrispondente autovettore*.

Supponiamo che gli autovalori di \mathbf{A} siano ordinati *in senso decrescente di modulo*, e che il primo autovalore in questo ordinamento abbia *modulo strettamente maggiore di tutti gli altri*:

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|.$$

Supponiamo inoltre che la matrice sia diagonalizzabile, e indichiamo con \mathbf{w}_p l'autovettore corrispondente all'autovalore λ_p , per $1 \leq p \leq n$. Essi dunque formano una base in \mathbb{C}^n , e quindi ogni vettore \mathbf{z} in tale spazio si rappresenta come

$$\mathbf{z} = \alpha_1 \mathbf{w}_1 + \alpha_2 \mathbf{w}_2 + \dots + \alpha_n \mathbf{w}_n$$

per opportuni coefficienti α_p .

Nota. Essendo \mathbf{A} reale per ipotesi, λ_1 è reale e \mathbf{w}_1 può essere scelto reale.

Applicando la matrice \mathbf{A} , abbiamo

$$\mathbf{A}\mathbf{z} = \alpha_1 \mathbf{A}\mathbf{w}_1 + \alpha_2 \mathbf{A}\mathbf{w}_2 + \dots + \alpha_n \mathbf{A}\mathbf{w}_n = \alpha_1 \lambda_1 \mathbf{w}_1 + \alpha_2 \lambda_2 \mathbf{w}_2 + \dots + \alpha_n \lambda_n \mathbf{w}_n.$$

Una seconda applicazione di \mathbf{A} fornisce

$$\mathbf{A}^2 \mathbf{z} = \mathbf{A}(\mathbf{A}\mathbf{z}) = \alpha_1 \lambda_1^2 \mathbf{w}_1 + \alpha_2 \lambda_2^2 \mathbf{w}_2 + \dots + \alpha_n \lambda_n^2 \mathbf{w}_n.$$

Iterando, dopo k applicazioni di \mathbf{A} otteniamo

$$\mathbf{z}^{(k)} := \mathbf{A}^k \mathbf{z} = \alpha_1 \lambda_1^k \mathbf{w}_1 + \alpha_2 \lambda_2^k \mathbf{w}_2 + \dots + \alpha_n \lambda_n^k \mathbf{w}_n.$$

Scriviamo tale vettore come

$$\mathbf{z}^{(k)} = \lambda_1^k \left(\alpha_1 \mathbf{w}_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k \mathbf{w}_2 + \dots + \alpha_n \left(\frac{\lambda_n}{\lambda_1} \right)^k \mathbf{w}_n \right) =: \lambda_1^k \mathbf{y}^{(k)}$$

Osserviamo ora che, essendo per ipotesi $|\lambda_1| > |\lambda_p|$ per ogni $p > 1$, si ha

$$\left| \alpha_p \left(\frac{\lambda_p}{\lambda_1} \right)^k \right| = |\alpha_p| \left| \frac{\lambda_p}{\lambda_1} \right|^k \rightarrow 0 \quad \text{per } k \rightarrow \infty.$$

Ciò significa che

$$\mathbf{y}^{(k)} \rightarrow \alpha_1 \mathbf{w}_1 \quad \text{per } k \rightarrow \infty,$$

cioè, supponendo $\alpha_1 \neq 0$, il vettore $\mathbf{y}^{(k)}$ tende ad allinearsi al primo autovettore \mathbf{w}_1 della matrice \mathbf{A} .

Corrispondentemente, anche il vettore calcolato $\mathbf{z}^{(k)} = \lambda_1^k \mathbf{y}^{(k)}$ tende ad allinearsi a \mathbf{w}_1 , ma la sua lunghezza tende a 0 oppure a ∞ nel caso in cui $|\lambda_1|$ sia rispettivamente minore o maggiore di 1.

È quindi opportuno *normalizzare* tale vettore, riportandolo ad ogni iterazione ad avere lunghezza unitaria.

Partendo da un vettore iniziale $\mathbf{z}^{(0)}$, costruiamo per $k = 1, 2, \dots$ iterativamente la successione di vettori

$$\begin{aligned} \mathbf{x}^{(k)} &= \frac{\mathbf{z}^{(k)}}{\|\mathbf{z}^{(k)}\|_2} \\ \mathbf{z}^{(k+1)} &= \mathbf{A} \mathbf{x}^{(k)} \end{aligned}$$

Per quanto riguarda l'approssimazione dell'autovalore, osserviamo che se $\mathbf{x}^{(k)}$ approssima l'autovettore \mathbf{w}_1 relativo all'autovalore λ_1 , e dunque

$$\mathbf{A} \mathbf{x}^{(k)} \sim \lambda_1 \mathbf{x}^{(k)}$$

allora

$$(\mathbf{x}^{(k)})^T \mathbf{A} \mathbf{x}^{(k)} \sim \lambda_1 (\mathbf{x}^{(k)})^T \mathbf{x}^{(k)} = \lambda_1.$$

Appare quindi naturale calcolare il *quoziente di Rayleigh*

$$\lambda_1^{(k)} = \frac{(\mathbf{x}^{(k)})^T \mathbf{A} \mathbf{x}^{(k)}}{(\mathbf{x}^{(k)})^T \mathbf{x}^{(k)}} = (\mathbf{x}^{(k)})^T \mathbf{A} \mathbf{x}^{(k)} = (\mathbf{x}^{(k)})^T \mathbf{z}^{(k+1)}$$

e prenderlo come approssimazione dell'autovalore λ_1 al passo k dell'iterazione.

Proprietà. La velocità con cui $\lambda_1^{(k)}$ converge a λ_1 è espressa da queste stime dell'errore:

$$|\lambda_1^{(k)} - \lambda_1| \leq C \left| \frac{\lambda_2}{\lambda_1} \right|^k \quad \text{se } \mathbf{A} \text{ non è simmetrica,}$$

oppure

$$|\lambda_1^{(k)} - \lambda_1| \leq C \left| \frac{\lambda_2}{\lambda_1} \right|^{2k} \quad \text{se } \mathbf{A} \text{ è simmetrica.}$$

Osservazioni.

- L'ipotesi che il vettore iniziale $\mathbf{x}^{(0)}$ abbia componente non nulla rispetto al primo autovettore ($\alpha_1 \neq 0$) non è restrittiva nella pratica, perchè gli errori di arrotondamento introducono dopo poche iterazioni una (piccola) componente, che poi viene amplificata.
- Se λ_1 ha molteplicità geometrica $m > 1$, il metodo genera ancora una approssimazione di tale autovalore, e genera una approssimazione di uno degli autovettori relativo a λ_1 , dipendente dal vettore iniziale scelto.
- Se invece $|\lambda_1| = |\lambda_2|$ ma $\lambda_1 \neq \lambda_2$, allora in generale il metodo non converge. Un esempio è dato dalla matrice

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

che ha autovalori $\lambda_1 = 1$ e $\lambda_2 = -1$, e che semplicemente scambia tra loro le due componenti del vettore a cui è applicata.

- Una volta calcolato l'autovalore di modulo più grande di \mathbf{A} , si possono adottare tecniche di *deflazione* per ridurre l'ordine della matrice e calcolare il secondo autovalore di modulo più grande, e così via.

Il metodo della potenza inversa

Ricordando che

$$\mathbf{A}\mathbf{w} = \lambda\mathbf{w} \quad \text{equivale a} \quad \mathbf{A}^{-1}\mathbf{w} = \frac{1}{\lambda}\mathbf{w},$$

possiamo applicare il metodo della potenza alla matrice inversa \mathbf{A}^{-1} per calcolare l'autovalore di modulo minimo di \mathbf{A} , nell'ipotesi che

$$|\lambda_1| < |\lambda_2| \leq |\lambda_3| \leq \dots \leq |\lambda_n|.$$

Il passo $\mathbf{z}^{(k+1)} = \mathbf{A}^{-1}\mathbf{x}^{(k)}$ equivale a $\mathbf{A}\mathbf{z}^{(k+1)} = \mathbf{x}^{(k)}$, e dunque non si inverte la matrice \mathbf{A} , bensì la si fattorizza una volta per tutte.

Il **metodo della potenza inversa** si esplicita in questo modo: **partendo da un vettore iniziale $\mathbf{z}^{(0)}$, costruiamo per $k = 1, 2, \dots$ iterativamente la successione di vettori**

$$\begin{aligned} \mathbf{x}^{(k)} &= \frac{\mathbf{z}^{(k)}}{\|\mathbf{z}^{(k)}\|_2} \\ \mathbf{z}^{(k+1)} &\text{ soluzione di } \mathbf{A}\mathbf{z}^{(k+1)} = \mathbf{x}^{(k)} \\ \lambda_1^{(k)} &= \frac{1}{(\mathbf{x}^{(k)})^T \mathbf{z}^{(k+1)}} \end{aligned}$$

Per $k \rightarrow \infty$ abbiamo $\lambda_1^{(k)} \rightarrow \lambda_1$ e $\mathbf{x}^{(k)} \rightarrow \mathbf{w}_1$, con $\mathbf{A}\mathbf{w}_1 = \lambda_1\mathbf{w}_1$.

Una variante del metodo della potenza inversa può essere utilizzato *per calcolare l'autovalore di A più vicino a un valore assegnato σ* .

Infatti un tale autovalore, sia esso λ_p , soddisfa per definizione

$$|\lambda_p - \sigma| \leq |\lambda_q - \sigma| \quad \text{per ogni } q \neq p.$$

Se vale la disuguaglianza stretta, allora siamo nelle ipotesi di applicazione del metodo della potenza inversa alla matrice

$$A - \sigma I,$$

i cui autovalori sono dati precisamente da $\lambda_q - \sigma$ al variare di $q = 1, 2, \dots, n$.

Il **comando MATLAB** `[V D]=eigs(A,K,SIGMA)` fornisce k autovalori e autovettori di A individuati dal parametro SIGMA. Se questo è un numero reale o complesso σ , otteniamo i k autovalori più vicini a σ .

Osservazione. L'idea di applicare traslazioni successive alla matrice A può essere usata nel metodo della potenza inversa per *accelerare la convergenza* verso un autovalore. Nelle iterazioni, si usano matrici di tipo

$$A - \sigma^{(k)} I,$$

dove $\sigma^{(k)}$ sono successive approssimazioni dell'autovalore cercato.

5. La decomposizione ai valori singolari di una matrice

La *decomposizione ai valori singolari* (**SVD - Singular Value Decomposition**) di una generica matrice reale A di ordine $m \times n$ ha la seguente forma

$$A = U\Sigma V^T$$

dove

- U è una matrice quadrata di ordine m , *ortogonale*;
- Σ è una matrice di ordine $m \times n$, *diagonale*;
- V è una matrice quadrata di ordine n , *ortogonale*.

Mediante la SVD, possiamo effettuare svariate operazioni di Algebra Lineare Numerica *nel modo numericamente più stabile*, quali ad esempio:

- calcolare il rango di una matrice
- decidere se una famiglia di vettori sono linearmente indipendenti
- risolvere un sistema lineare sovra- o sotto-determinato
- calcolare una “inversa” di una matrice rettangolare.

Premessa. Un risultato fondamentale

Sia B una matrice reale quadrata di ordine n , avente rango $r \leq n$, *simmetrica e semi-definita positiva*, cioè tale che

$$x^T B x \geq 0 \quad \text{per ogni } x \in \mathbb{R}^n.$$

Allora

- B ha r autovalori reali strettamente positivi e $n - r$ autovalori nulli (contati con la loro molteplicità), che possiamo ordinare in modo decrescente:

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > \lambda_{r+1} = \dots = \lambda_n = 0.$$

- B è diagonalizzabile nella forma $B = W\Lambda W^T$, con Λ matrice diagonale degli autovalori e W matrice ortogonale i cui vettori colonna sono gli autovettori di B , soddisfacenti

$$Bw_i = \lambda_i w_i, \quad 1 \leq i \leq n.$$

- L'insieme degli autovettori w_i di B forma dunque una *base ortonormale* di \mathbb{R}^n .

Se \mathbf{x} è un generico vettore colonna di \mathbb{R}^n , avremo dunque

$$\mathbf{x} = \alpha_1 \mathbf{w}_1 + \alpha_2 \mathbf{w}_2 + \cdots + \alpha_n \mathbf{w}_n = \sum_{i=1}^n \alpha_i \mathbf{w}_i$$

e poichè la base è ortonormale ($\mathbf{w}_j^T \mathbf{w}_i = \delta_{ij}$) si ha

$$\alpha_i = \mathbf{w}_i^T \mathbf{x}, \quad 1 \leq i \leq n,$$

da cui otteniamo la “classica” rappresentazione di un vettore rispetto a una base ortonormale

$$\mathbf{x} = \sum_{i=1}^n (\mathbf{w}_i^T \mathbf{x}) \mathbf{w}_i.$$

Si noti che possiamo scrivere

$$\mathbf{x} = \mathbf{W} \boldsymbol{\alpha} \quad \text{con} \quad \boldsymbol{\alpha} = \mathbf{W}^T \mathbf{x}$$

e dunque la rappresentazione precedente non è altro che l'identità

$$\mathbf{x} = \mathbf{W} \mathbf{W}^T \mathbf{x},$$

conseguenza della proprietà di ortogonalità $\mathbf{W} \mathbf{W}^T = \mathbf{I}$ della matrice \mathbf{W} .

Il rango di una matrice

Ricordiamo che per un generica matrice \mathbf{A} di ordine $m \times n$, definiamo il rango $r = \text{rank}(\mathbf{A})$ come la *dimensione della sua immagine*

$$\text{rank}(\mathbf{A}) = \dim \text{Im} \mathbf{A} = \dim \{ \mathbf{A} \mathbf{x} : \mathbf{x} \in \mathbb{R}^n \},$$

cioè il *massimo numero di colonne linearmente indipendenti di \mathbf{A}* .

Si dimostra che

$$\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}^T),$$

dunque il rango è anche il *massimo numero di righe linearmente indipendenti di \mathbf{A}* .

Inoltre, vale la relazione fondamentale tra le dimensioni del nucleo e dell'immagine di \mathbf{A}

$$\text{rank}(\mathbf{A}) = n - \dim \text{Ker} \mathbf{A},$$

dove $\text{Ker} \mathbf{A} = \{ \mathbf{x} \in \mathbb{R}^n : \mathbf{A} \mathbf{x} = \mathbf{0} \}$.

Abbiamo allora il seguente semplice ma importante risultato.

Proprietà. Per ogni matrice A , si ha

$$\text{rank}(A^T A) = \text{rank}(A A^T) = \text{rank}(A).$$

Infatti, scriviamo la relazione fondamentale per A e $A^T A$:

$$\begin{aligned}\text{rank}(A) &= n - \dim \text{Ker} A, \\ \text{rank}(A^T A) &= n - \dim \text{Ker}(A^T A).\end{aligned}$$

Ma si ha

$$\text{Ker} A = \text{Ker}(A^T A),$$

in quanto se $Ax = 0$, anche $A^T Ax = 0$, e viceversa se $A^T Ax = 0$ allora $0 = x^T A^T Ax = (Ax)^T Ax = \|Ax\|_2^2$, da cui $Ax = 0$.

Concludiamo che

$$\text{rank}(A) = \text{rank}(A^T A).$$

L'altra relazione si ottiene in modo analogo.

Autovalori e autovettori delle matrici AA^T e $A^T A$

- Consideriamo la matrice quadrata AA^T , di ordine m e rango r .

- I suoi autovalori λ_i soddisfano

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > 0, \quad \lambda_{r+1} = \lambda_{r+2} = \dots = \lambda_m = 0.$$

- I corrispondenti autovettori $u_i \in \mathbb{R}^m$ soddisfano

$$AA^T u_i = \lambda_i u_i, \quad 1 \leq i \leq m,$$

e formano la matrice ortogonale U .

- Similmente, consideriamo la matrice quadrata $A^T A$, di ordine n e rango r .

- I suoi autovalori μ_j soddisfano

$$\mu_1 \geq \mu_2 \geq \dots \geq \mu_r > 0, \quad \mu_{r+1} = \mu_{r+2} = \dots = \mu_n = 0.$$

- I corrispondenti autovettori $v_j \in \mathbb{R}^n$ soddisfano

$$A^T A v_j = \mu_j v_j, \quad 1 \leq j \leq n,$$

e formano la matrice ortogonale V .

Proprietà. Valgono i seguenti risultati:

- Gli autovalori > 0 di $\mathbf{A}\mathbf{A}^T$ e $\mathbf{A}^T\mathbf{A}$ coincidono:

$$\lambda_i = \mu_i, \quad 1 \leq i \leq r.$$

- I corrispondenti autovettori \mathbf{u}_i e \mathbf{v}_i possono essere scelti in modo da soddisfare le relazioni

$$\mathbf{A}\mathbf{v}_i = \sigma_i \mathbf{u}_i,$$

$$\mathbf{A}^T \mathbf{u}_i = \sigma_i \mathbf{v}_i,$$

avendo posto

$$\sigma_i = \sqrt{\lambda_i} = \sqrt{\mu_i}.$$

Infatti, se ad esempio fissiamo gli autovettori \mathbf{u}_i e poniamo $\sigma_i = \sqrt{\lambda_i}$, possiamo definire

$$\mathbf{v}_i := \frac{1}{\sigma_i} \mathbf{A}^T \mathbf{u}_i, \quad 1 \leq i \leq r.$$

Allora si ha

$$\mathbf{A}\mathbf{v}_i = \frac{1}{\sigma_i} \mathbf{A}\mathbf{A}^T \mathbf{u}_i = \frac{\lambda_i}{\sigma_i} \mathbf{u}_i = \sigma_i \mathbf{u}_i$$

e quindi

$$\mathbf{A}^T \mathbf{A}\mathbf{v}_i = \sigma_i \mathbf{A}^T \mathbf{u}_i = \sigma_i^2 \mathbf{v}_i = \lambda_i \mathbf{v}_i.$$

Dunque \mathbf{v}_i è autovettore di $\mathbf{A}^T\mathbf{A}$, con autovalore $\mu_i = \lambda_i$.

Verso la decomposizione ai valori singolari di \mathbf{A}

Usando la proprietà che \mathbf{U} e \mathbf{V} sono matrici ortogonali, e dunque

$$\mathbf{U}\mathbf{U}^T = \mathbf{I}_m, \quad \mathbf{V}\mathbf{V}^T = \mathbf{I}_n$$

(dove \mathbf{I}_k indica la matrice identità di ordine k), possiamo scrivere

$$\mathbf{A} = \mathbf{I}_m \mathbf{A} \mathbf{I}_n = \mathbf{U}\mathbf{U}^T \mathbf{A} \mathbf{V}\mathbf{V}^T = \mathbf{U}(\mathbf{U}^T \mathbf{A} \mathbf{V})\mathbf{V}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,$$

avendo introdotto la matrice di ordine $m \times n$

$$\mathbf{\Sigma} = \mathbf{U}^T \mathbf{A} \mathbf{V}.$$

Dobbiamo ora capire come è fatta la matrice $\mathbf{\Sigma}$...

Definizione. Chiamiamo **valori singolari** della matrice \mathbf{A} i numeri

$$\sigma_i = \sqrt{\lambda_i}, \quad 1 \leq i \leq \min(m, n).$$

Se r è il rango della matrice, essi soddisfano

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0, \quad \sigma_{r+1} = \sigma_{r+2} = \cdots = \sigma_{\min(m, n)} = 0.$$

Vale allora il seguente risultato.

Proprietà. La matrice Σ è diagonale, e contiene i valori singolari sulla diagonale principale:

$$(\Sigma)_{ij} = \begin{cases} \sigma_i & \text{per } 1 \leq i = j \leq \min(m, n), \\ 0 & \text{altrimenti.} \end{cases}$$

Ad esempio

$$\Sigma = \begin{pmatrix} * & 0 & 0 \\ 0 & * & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{oppure} \quad \Sigma = \begin{pmatrix} * & 0 & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Dimostrazione della Proprietà. Per definizione di Σ , abbiamo

$$(\Sigma)_{ij} = \mathbf{u}_i^T \mathbf{A} \mathbf{v}_j.$$

Distinguiamo due casi, a seconda che $\sigma_j > 0$ oppure $\sigma_j = 0$.

- Se $\sigma_j > 0$, allora $\mathbf{A} \mathbf{v}_j = \sigma_j \mathbf{u}_j$, dunque

$$\mathbf{u}_i^T \mathbf{A} \mathbf{v}_j = \sigma_j \mathbf{u}_i^T \mathbf{u}_j = \sigma_j \delta_{ij}.$$

- Se $\sigma_j = 0$, dobbiamo ricordare che possiamo decomporre \mathbb{R}^m come

$$\mathbb{R}^m = \text{Im } \mathbf{A} +_{\perp} (\text{Im } \mathbf{A})^{\perp} = \text{Im } \mathbf{A} +_{\perp} \text{Ker } \mathbf{A}^T.$$

Dunque scriviamo

$$\mathbf{u}_i = \mathbf{y}_i + \mathbf{z}_i \quad \text{con } \mathbf{y}_i = \mathbf{A} \mathbf{w}_i \in \text{Im } \mathbf{A} \quad \text{e} \quad \mathbf{z}_i \in \text{Ker } \mathbf{A}^T,$$

da cui

$$\mathbf{u}_i^T \mathbf{A} \mathbf{v}_j = (\mathbf{A} \mathbf{w}_i + \mathbf{z}_i)^T \mathbf{A} \mathbf{v}_j = \mathbf{w}_i^T \mathbf{A}^T \mathbf{A} \mathbf{v}_j + (\mathbf{A}^T \mathbf{z}_i)^T \mathbf{v}_j = \mathbf{w}_i^T \mathbf{0} + \mathbf{0}^T \mathbf{v}_j = 0.$$

Abbiamo dunque stabilito il seguente risultato.

Proprietà. Ogni matrice reale A , di ordine $m \times n$ e di rango r , può essere fattorizzata nella forma

$$A = U \Sigma V^T$$

dove

- U è una matrice quadrata di ordine m , *ortogonale*; precisamente, U è formata dagli autovettori della matrice AA^T .
- Σ è una matrice di ordine $m \times n$, *diagonale*; essa contiene i *valori singolari* di A , ordinati in senso decrescente e tali che r di essi sono strettamente positivi, mentre i rimanenti sono nulli.
- V è una matrice quadrata di ordine n , *ortogonale*; precisamente, V è formata dagli autovettori della matrice $A^T A$.

Più precisamente, possiamo dire che:

- Le prime r colonne di U , ossia u_1, \dots, u_r , formano una base ortonormale dello spazio vettoriale

$$\text{Im } A = \{Ax : x \in \mathbb{R}^n\}.$$

- Le rimanenti $m - r$ colonne di U , ossia u_{r+1}, \dots, u_m , formano una base ortonormale dello spazio vettoriale

$$\text{Ker } A^T = \{y \in \mathbb{R}^m : A^T y = 0\}.$$

- Le prime r colonne di V , ossia v_1, \dots, v_r , formano una base ortonormale dello spazio vettoriale

$$\text{Im } A^T = \{A^T y : y \in \mathbb{R}^m\}.$$

- Le rimanenti $n - r$ colonne di V , ossia v_{r+1}, \dots, v_n , formano una base ortonormale dello spazio vettoriale

$$\text{Ker } A = \{x \in \mathbb{R}^n : Ax = 0\}.$$

Comandi MATLAB

I principali comandi relativi alla decomposizione ai valori singolari sono:

- `s=svd(A)` per ottenere un vettore contenente i valori singolari di A ;
- `[U S V]=svd(A)` per ottenere le tre matrici che compongono la fattorizzazione.

Dal punto di vista numerico, la SVD può essere calcolata riducendo dapprima la matrice a *forma bidiagonale*, e poi applicando un *metodo iterativo basato su trasformazioni QR*.

Il numero di operazioni richieste è $O(mn^2)$.

Interpretazione della decomposizione ai valori singolari

L'applicazione della matrice A a un vettore $x \in \mathbb{R}^n$ per ottenere un vettore $y \in \mathbb{R}^m$,

$$y = Ax,$$

può essere “letta” nel seguente modo:

- esprimiamo il vettore x nella base ortonormale v_1, \dots, v_n di \mathbb{R}^n , come

$$x = V\alpha, \quad \text{mediante i coefficienti } \alpha = V^T x;$$

- esprimiamo il vettore y nella base ortonormale u_1, \dots, u_m di \mathbb{R}^m , come

$$y = U\beta, \quad \text{mediante opportuni coefficienti } \beta;$$

- rispetto alle basi ortonormali, l'applicazione della matrice A diventa una semplice trasformazione diagonale dei coefficienti (*scaling*):

$$\beta = \Sigma\alpha.$$

Abbiamo visto che *il numero di valori singolari di \mathbf{A} è pari al rango di \mathbf{A}* . Dunque, una volta effettuata la decomposizione ai valori singolari di \mathbf{A} , potremmo porre

$$\text{rank}(\mathbf{A}) = \#\{ \sigma_i > 0, 1 \leq i \leq \min(m, n) \}.$$

Tuttavia, lavorando in aritmetica con precisione finita, si preferisce selezionare i valori singolari sulla base di una *soglia*, diciamo $\tau > 0$, che viene scelta in base alla precisione di macchina e anche alla “accuratezza” con cui è stata calcolata la matrice \mathbf{A} .

Si definisce quindi il *rango numerico*, (o *rango effettivo*) di \mathbf{A} la quantità

$$\text{rank}_\tau(\mathbf{A}) = \#\{ \sigma_i > \tau, 1 \leq i \leq \min(m, n) \}.$$

Osservazione. Il rango della matrice potrebbe anche essere calcolato mediante un metodo di fattorizzazione di tipo $\mathbf{PA} = \mathbf{LU}$, contando dopo quanti passi si trovano tutti *elementi pivot* uguali a 0 (oppure al di sotto di una certa soglia τ). Però, l'uso della SVD, per quanto più costoso, fornisce in genere risultati numericamente più affidabili.

Applicazioni. II - Selezione di un insieme di vettori indipendenti

Supponiamo di avere n vettori $\mathbf{a}_1, \dots, \mathbf{a}_n$ di ordine m .

Ci poniamo le seguenti domande:

- ① Qual è il numero massimo di vettori linearmente indipendenti tra loro?
- ② Come selezioniamo un sottoinsieme di vettori linearmente indipendenti?
- ③ Come esprimiamo i rimanenti vettori in termini di questi?

Vediamo le risposte.

1. Numero massimo di vettori linearmente indipendenti

Scrivendo i vettori come vettori colonna, è sufficiente formare la matrice di ordine $m \times n$

$$\mathbf{A} = (\mathbf{a}_1 \quad \mathbf{a}_2 \quad \dots \quad \mathbf{a}_n)$$

e calcolarne il rango

$$r = \text{rank}(\mathbf{A}).$$

Questo è il numero cercato.

2. Selezione di un sottoinsieme di r vettori linearmente indipendenti

Ovviamente, il problema è non banale solo se $r < n$. Eseguiamo la fattorizzazione SVD di A

$$A = U\Sigma V^T$$

e ricordiamo che le ultime $n - r$ colonne di V formano una base ortonormale di $\text{Ker}A$. Ciò implica che

$$AV_{n-r} = O$$

dove V_{n-r} è la sottomatrice di V di ordine $n \times (n - r)$ che ne raccoglie le ultime $n - r$ colonne.

Poichè le colonne di V_{n-r} sono ortogonali, tale matrice ha rango massimo $n - r$. Dunque possiamo trovare $n - r$ righe di V_{n-r} linearmente indipendenti.¹

Siano i_1, \dots, i_r gli indici delle rimanenti r righe.

Proprietà. Gli r vettori

$$a_{i_1}, \dots, a_{i_r}$$

formano un insieme massimale di vettori linearmente indipendenti.

¹Ad esempio, mediante una fattorizzazione di tipo LU della matrice rettangolare V_{n-r} .

3. Rappresentazione dei rimanenti $n - r$ vettori

Applicando una *permutazione di righe*, non è restrittivo supporre che le $n - r$ righe linearmente indipendenti di V_{n-r} siano le prime:

$$PV_{n-r} = \begin{pmatrix} W_{n-r} \\ W_r \end{pmatrix}$$

con P matrice $n \times n$ di permutazione, W_{n-r} matrice quadrata di ordine $n - r$ *non-singolare*, e W_r matrice di ordine $r \times (n - r)$.

La matrice

$$\tilde{A} = AP = \begin{pmatrix} \tilde{A}_{n-r} & \tilde{A}_r \end{pmatrix}$$

ottenuta da A mediante una *permutazione di colonne*, contiene gli r vettori indipendenti selezionati nelle ultime r colonne, e i rimanenti vettori nelle prime $n - r$ colonne.

Allora

$$O = AV_{n-r} = AP PV_{n-r} = \begin{pmatrix} \tilde{A}_{n-r} & \tilde{A}_r \end{pmatrix} \begin{pmatrix} W_{n-r} \\ W_r \end{pmatrix}$$

da cui

$$\tilde{A}_{n-r} W_{n-r} + \tilde{A}_r W_r = O.$$

Di qui otteniamo la rappresentazione cercata

$$\tilde{A}_{n-r} = -\tilde{A}_r (W_r W_{n-r}^{-1}).$$

Se A è una matrice quadrata, che si fattorizza come

$$A = U\Sigma V^T$$

con U , V ortogonali e Σ quadrata invertibile, allora A è invertibile e si ha

$$A^{-1} = (V^T)^{-1}\Sigma^{-1}U^{-1} = V\Sigma^{-1}U^T.$$

Questa osservazione motiva la seguente definizione di *pseudo-inversa* di una qualunque matrice A di ordine $m \times n$, a partire dalla sua decomposizione ai valori singolari $A = U\Sigma V^T$.

Definizione. La **pseudo-inversa** di A (detta anche **inversa di Moore-Penrose**) è la matrice di ordine $n \times m$

$$A^\dagger = V\Sigma^\dagger U^T$$

dove Σ^\dagger è la matrice diagonale di ordine $n \times m$ i cui elementi diagonali sono

$$(\Sigma^\dagger)_{ii} = \begin{cases} \frac{1}{\sigma_i} & \text{se } \sigma_i > 0, \\ 0 & \text{se } \sigma_i = 0. \end{cases}$$

Proprietà

- La pseudo-inversa A^\dagger è caratterizzata dalla seguenti proprietà:

$$AA^\dagger \text{ è simmetrica, } \quad \text{e soddisfa } (AA^\dagger)A = A,$$

$$A^\dagger A \text{ è simmetrica, } \quad \text{e soddisfa } (A^\dagger A)A^\dagger = A^\dagger.$$

- Se A è quadrata non-singolare, allora

$$A^\dagger = A^{-1}.$$

- Se A è di ordine $m \times n$ con $n < m$, e ha rango massimo $r = n$, allora

$$A^\dagger = (A^T A)^{-1} A^T.$$

Comandi MATLAB

- $X = \text{pinv}(A)$ fornisce la pseudo-inversa di A , a partire dalla sua SVD.
I valori singolari al di sotto di una tolleranza di default sono trattati come zeri.
- $X = \text{pinv}(A, tol)$ permette di specificare la tolleranza tol .

Finora, abbiamo visto come risolvere il sistema lineare

$$Ax = b$$

- quando A è una matrice quadrata invertibile: la soluzione si esprime (formalmente) come

$$x = A^{-1}b;$$

- quando A è una matrice di ordine $m \times n$ con $n < m$, e ha rango massimo $r = n$: la soluzione nel senso dei minimi quadrati si esprime (formalmente) come

$$x = (A^T A)^{-1} A^T b;$$

Possiamo parlare di *soluzione nel senso dei minimi quadrati* del sistema lineare anche nei rimanenti casi, ossia quando A è una matrice di ordine $m \times n$

- con $n \leq m$, ma avente *rango non massimo* $r < n$; oppure
- con $m < n$ (*sistema lineare sotto-determinato*).

Notiamo che in entrambi i casi, si ha $r < n$.

Definizione. Chiamiamo *soluzione nel senso dei minimi quadrati* del sistema lineare $Ax = b$, con A matrice di ordine $m \times n$ arbitraria e b vettore di ordine m , ogni vettore x di ordine n che soddisfa

$$\|Ax - b\|_2 = \min_{y \in \mathbb{R}^n} \|Ay - b\|_2.$$

Nota. Se la matrice ha rango $r < n$, allora il problema ammette *infinite soluzioni*.

Infatti, il nucleo $\text{Ker } A$ contiene infiniti vettori non nulli z , tali che $Az = 0$; dunque se x è soluzione del senso dei minimi quadrati del sistema, anche $x + z$ lo è, in quanto $A(x + z) - b = Ax - b$.

Proprietà. Il vettore

$$x = A^\dagger b$$

è soluzione del sistema lineare $Ax = b$ nel senso dei minimi quadrati; precisamente, è la soluzione avente la minima norma euclidea $\|x\|_2$.