

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE
Corso di Laurea in Matematica

Problema generalizzato ai minimi quadrati applicato all' Information Retrieval

Tesi di Laurea in Analisi Numerica

Relatore:
Chiar.ma Prof.ssa
Valeria Simoncini

Presentata da:
Werther Solazzi

Anno Accademico 2024/2025

A chi mi vuole bene

Introduzione

In questa tesi si affronta un problema di *Information Retrieval* (IR) basato sulla rappresentazione vettoriale del contenuto testuale. L'obiettivo è individuare, a partire da una query espressa nel vocabolario dei termini, i documenti più rilevanti all'interno di due collezioni testuali differenti, descritte tramite le rispettive matrici termine-documento $\mathbf{A}$ e $\mathbf{B}$, associate a due insiemi di documenti tematicamente distinti.

Il problema viene formalizzato come un problema ai minimi quadrati generalizzato nella forma:

$$\min_X \|\mathbf{b}\mathbf{b}^T - \mathbf{A}\mathbf{X}\mathbf{B}^T\|_F$$

dove $\mathbf{b}$ rappresenta la query in forma vettoriale, mentre la matrice $\mathbf{X}$ modella il grado di pertinenza tra i documenti di $\mathbf{A}$ e quelli di $\mathbf{B}$ rispetto alla query. In altre parole, $\mathbf{X}$ ha lo scopo di evidenziare quali coppie di documenti (uno per ciascun insieme) risultino maggiormente pertinenti rispetto all'informazione espressa da $\mathbf{b}$.

La risoluzione di tale problema consente quindi di identificare quale dei due insiemi tematici riesce a descrivere meglio la query; a tal fine sono stati proposti e implementati due approcci distinti per la risoluzione del problema:

- un primo metodo, più adatto a matrici di dimensioni ridotte, che generalizza il metodo QR per la risoluzione efficiente del sistema;
- un secondo approccio, basato sull'algoritmo LSQR, pensato per matrici di grandi dimensioni e caratterizzate da sparsità.

Nella parte finale dell'elaborato si testa l'algoritmo matriciale LSQR sul dataset Reuters-21578, fonte di riferimento per lo studio di sistemi di IR, e tramite l'uso della decomposizione SVD, si traggono conclusioni sulle soluzioni trovate nelle diverse casistiche.

Indice

Introduzione	i
1 Metodo QR	1
1.1 Introduzione	1
1.2 Il metodo QR in forma standard	1
1.3 La funzione <code>minquad_house</code>	3
1.4 Il metodo QR in forma matriciale	4
1.5 La funzione <code>minquad_house_matriciale</code>	5
2 Algoritmo LSQR	9
2.1 Bidiagonalizzazione di Golub-Kahan	9
2.2 Struttura dell'algoritmo LSQR	11
2.3 La funzione <code>lsqr</code> secondo Paige e Saunders	14
3 Algoritmo LSQR matriciale	17
3.1 Prodotto di Kronecker	17
3.2 Algoritmo LSQR matriciale	19
4 Applicazione dei Minimi Quadrati all'Information Retrieval	24
4.1 Cos'è l'Information Retrieval?	24
4.2 Un esempio pratico	24
4.3 Il dataset Reuters-21578	26
4.3.1 Caso 0	29
4.3.2 Caso 1	30
4.3.3 Caso 2	32
Conclusioni	35
Bibliografia	36

Elenco delle figure

3.1	Convergenza dei due algoritmi LSQR	23
4.1	Matrice termine-documento	25
4.2	Soluzioni relative a queries diverse	26
4.3	$c = A_1$	29
4.4	$c = \mathbf{g1}(\mathbf{n+1}, :)'$	30
4.5	$c = \mathbf{g2}(\mathbf{m+1}, :)'$	31
4.6	$c = \mathbf{g3}(1, :)'$	32
4.7	$c = \mathbf{g12}(1, :)'$	33

Capitolo 1

Metodo QR

1.1 Introduzione

Il problema ai minimi quadrati è usato per approssimare la soluzione di un sistema $Ax = b$ sovradeterminato, cioè con un numero di equazioni superiore al numero di incognite. Dati $A \in \mathbb{R}^{n \times m}$, con $n \geq m$, e $b \in \mathbb{R}^n$, il problema ai minimi quadrati consiste nel determinare il vettore $x \in \mathbb{R}^m$ che minimizza il residuo $\mathbf{r} := b - Ax$, solitamente in norma euclidea, cioè:

$$\min_{x \in \mathbb{R}^m} \|\mathbf{b} - \mathbf{A}x\|_2 \quad (1.1)$$

Si presenta di seguito uno dei metodi classici per la risoluzione, chiamato metodo QR, che si basa sulla decomposizione QR della matrice dei coefficienti A . Si suppone che tale A abbia rango massimo, uguale ad m .

Si analizza di seguito una generalizzazione del metodo QR per risolvere il problema ai minimi quadrati matriciale, oggetto di questa tesi.

1.2 Il metodo QR in forma standard

Sia $\mathbf{A}$ come sopra, scriviamo la fattorizzazione QR di $\mathbf{A}$:

$$A = QR = [Q_1, Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1,$$

dove $Q \in \mathbb{R}^{n \times n}$ ortogonale, $Q = [Q_1, Q_2]$ con $Q_1 \in \mathbb{R}^{n \times m}$ e $Q_2 \in \mathbb{R}^{n \times n-m}$ aventi colonne ortonormali, e $R \in \mathbb{R}^{n \times m}$ tale che con $R_1 \in \mathbb{R}^{m \times m}$ rettangolare superiore. Notiamo in particolare che $\text{Range}(Q_1) = \text{Range}(A)$ e $\text{Range}(Q_2) = \text{Range}(A)^\perp$, in quanto $Q_2^T Q_1 = 0$. Si ha dunque:

$$\begin{aligned} \|b - Ax\|_2^2 &= \|b - QRx\|_2^2 = \|Q(Q^T b) - Rx\|_2^2 = \|Q^T b - Rx\|_2^2 = \\ &= \left\| Q^T b - \begin{bmatrix} R_1 x \\ 0 \end{bmatrix} \right\|_2^2 = \left\| \begin{bmatrix} Q_1^T b - R_1 x \\ Q_2^T b \end{bmatrix} \right\|_2^2 = \|Q_1^T b - R_1 x\|_2^2 + \|Q_2^T b\|_2^2. \end{aligned}$$

Per minimizzare la norma del residuo, è possibile agire solo sul primo termine della somma che appare nell'ultima espressione sopra, cioè dev'essere $Q_1^T b - R_1 x = 0$. Siccome la matrice R_1 è non singolare, il problema ai minimi quadrati (1.1) ha un'unica soluzione data da $x = R_1^{-1}(Q_1^T b)$. In tal caso si ha quindi:

$$\min_{x \in \mathbb{R}^m} \|b - Ax\|_2^2 \equiv \|Q_2^T b\|_2^2$$

dove $\|Q_2^T b\| = \|Q_2 Q_2^T b\|$ e $Q_2 Q_2^T b$ è esattamente la proiezione di b nello spazio ortogonale alle colonne di A ; da quanto appena scritto si ha che $0 = A^T r = A^T(b - Ax)$, da cui si ottiene il metodo dell'equazione normale per la risoluzione del problema ai minimi quadrati, o per lo meno si dimostra che se x risolve (1.1), allora x è una soluzione dell'equazione normale $A^T A x = A^T b$. Da quest'ultimo metodo si ricava un'importante osservazione di natura geometrica, utile per valutare la bontà dell'approssimazione numerica ottenuta: sebbene la norma del residuo classico $r = b - Ax$ non tenda in generale a zero, il residuo normale $A^T r$, ovvero la proiezione del residuo sul sottospazio generato dalle colonne di A , tende invece a zero. Questa proprietà sarà utilizzata come criterio per verificare l'accuratezza della soluzione numerica.

Osservazione 1.2.1. Il metodo QR permette di risolvere accuratamente anche quei problemi con A avente colonne quasi linearmente dipendenti, che metodi come l'equazione normale non riescono a gestire, consideriamo infatti il seguente caso. Siano:

$$A = \begin{pmatrix} 3 & 3 \\ 4 & 4 \\ 0 & \alpha \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

dove $\alpha \in \mathbb{R}$ è tale che $u \leq \alpha < 1$ e $\alpha^2 < u$, dove u indica la precisione di macchina. Mediante il metodo dell'equazione normale, calcolando i prodotti $A^T A$ e $A^T b$, si ottiene:

$$A^T A = \begin{pmatrix} 25 & 25 \\ 25 & 25 + \alpha^2 \end{pmatrix}, \quad A^T b = \begin{pmatrix} 7 \\ 7 + \alpha \end{pmatrix}$$

da cui la soluzione algebrica del sistema normale risulta essere $x = \left(\frac{7}{25} - \frac{1}{\alpha}, \frac{1}{\alpha}\right)^T$. Essendo $\alpha^2 < u$, la rappresentazione di $A^T A$ sul calcolatore è data da:

$$fl(A^T A) = \begin{pmatrix} 25 & 25 \\ 25 & 25 \end{pmatrix}$$

con le due colonne uguali, cioè $fl(A^T A)$ è singolare! In questa circostanza l'equazione normale in aritmetica con precisione finita non ha soluzioni, e quindi il metodo dell'equazione normale non risolve il problema ai minimi quadrati. Consideriamo invece la fattorizzazione QR di A :

$$A = Q_1 R_1 = \begin{pmatrix} \frac{3}{5} & 0 \\ \frac{4}{5} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 5 & 5 \\ 0 & \alpha \end{pmatrix};$$

la matrice R_1 ha numero di condizionamento $k_F(R_1) = \|R_1\|_F \|R_1^{-1}\|_F \approx \mathcal{O}(\frac{1}{\alpha})$, che è significativamente sotto il valore dell'inversa della precisione di macchina, quindi la soluzione ottenuta è accurata fintanto che $k_F(R_1) \ll u^{-1}$.

1.3 La funzione minquad_house

Descriviamo ora i passi principali della procedura numerica per determinare la soluzione del problema ai minimi quadrati mediante fattorizzazione QR:

1. Applico le trasformazioni ortogonali (di Householder o Givens, dipende dalla struttura di A) ad A e b contemporaneamente:

$$[\mathbf{R}_1, \hat{\mathbf{b}}] := \mathbf{P}_m \mathbf{P}_{m-1} \cdots \mathbf{P}_1 [\mathbf{A}, \mathbf{b}]$$

2. Determino $x = (\mathbf{R}_1)^{-1}(\hat{\mathbf{b}})_{1:m}$ risolvendo il sistema triangolare superiore.

Nello specifico la funzione `minquad_house` è basata sull'applicazione delle trasformazioni ortogonali di Householder alla matrice dei coefficienti A e al vettore dei termini noti b simultaneamente. Di seguito si commentano passo dopo passo le istruzioni principali della funzione, per chiarirne il funzionamento.

```
function [x]=minquad_house(A,b)
[n,m]=size(A);

% Inizializzazione
R=A;
```

Si inizializza il procedimento definendo la matrice $\mathbf{R}$ e ponendola uguale ad A , in tal modo non si modifica direttamente la matrice dei coefficienti e alla fine del ciclo $\mathbf{R} = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$

```
for k=1:m
% creo il vettore di Householder
x=R(k:n,k);
alpha=-sign(x(1))*norm(x);
v=x-alpha*eye(n-k+1,1);
beta=2/(v'*v);
```

Si crea il k -esimo vettore di Householder $v = x - \alpha e_1$, dove $e_1 \in \mathbb{R}^{n-k+1}$

```
% applico la matrice di Householder da sinistra ad A e b
R(k:n,k:m)=R(k:n,k:m)-v*(beta*(v'*R(k:n,k:m)));
b(k:n)=b(k:n)-v*(beta*(v'*b(k:n)));
end
```

Si applica la k -esima matrice di riflessione di Householder $P_k = I_k - \beta v v^T \in \mathbb{R}^{(n-k+1) \times (n-k+1)}$

```
% risoluzione sistema triangolare superiore
x=RisolviTriangSup(R(1:m,1:m),b(1:m));
end
```

Infine si risolve il sistema triangolare superiore invertibile, avendo R_1 rango massimo.

1.4 Il metodo QR in forma matriciale

Siano $\mathbf{A} \in \mathbb{R}^{n_1 \times m_1}$ e $\mathbf{B} \in \mathbb{R}^{n_2 \times m_2}$, con $n_1 \geq m_1$ e $n_2 \geq m_2$ ¹, e siano $\mathbf{b}_1 \in \mathbb{R}^{n_1}$, $\mathbf{b}_2 \in \mathbb{R}^{n_2}$, consideriamo il seguente problema ai minimi quadrati:

$$\min_{X \in \mathbb{R}^{m_1 \times m_2}} \|\mathbf{b}_1 \mathbf{b}_2^T - \mathbf{A} \mathbf{X} \mathbf{B}^T\|_F \quad (1.2)$$

In dimensione ridotte è possibile risolvere tale problema generalizzando il metodo QR che si è precedentemente descritto; la procedura in generale è impostata nel modo seguente:

- Si risolvono separatamente i due problemi ai minimi quadrati:

$$\min_{x \in \mathbb{R}^{m_1}} \|b_1 - A x_1\|_2, \quad \min_{x \in \mathbb{R}^{m_2}} \|b_2^T - x_2^T B^T\|_2$$

- Le due soluzioni ottenute x_1 e x_2 determinano la matrice cercata X mediante $X = x_1 x_2^T$

Osservazione 1.4.2. La soluzione del problema ai minimi quadrati trasposto, ossia:

$$\min_x \|b^T - x^T A^T\|_2$$

è equivalente alla soluzione dello stesso problema non trasposto, dal momento che:

$$\|b^T - x^T A^T\|_2 = \|(b - Ax)^T\|_2 = \|b - Ax\|_2$$

il che implica che si possono risolvere in maniera standard i due problemi ai minimi quadrati oppure, come implementato nella funzione MATLAB che si esporrà di seguito, si può risolvere il problema trasposto applicando simultaneamente ad A e b , in questo esempio, le matrici di Householder da destra, anzichè da sinistra come regolarmente opera il metodo QR.

Proposizione 1.1. Siano $A = Q_{1A} R_{1A}$ e $B = Q_{1B} R_{1B}$ le decomposizioni QR ridotte di A e B rispettivamente, allora la soluzione del problema matriciale ai minimi quadrati (1.2) è data da:

$$X = R_{1A}^{-1} (Q_{1A}^T b_1 b_2^T Q_{1B}) R_{1B}^{-T}$$

Dimostrazione. Siano A , B , b_1 e b_2 come in (1.2), scriviamo le fattorizzazioni QR di A e B :

•

$$A = Q_A R_A = [Q_{1A}, Q_{2A}] \begin{bmatrix} R_{1A} \\ 0 \end{bmatrix} = Q_{1A} R_{1A}$$

dove $Q_A \in \mathbb{R}^{n_1 \times n_1}$, tale che $Q_{1A} \in \mathbb{R}^{n_1 \times m_1}$, $Q_{2A} \in \mathbb{R}^{n_1 \times (n_1 - m_1)}$ e $R_{1A} \in \mathbb{R}^{m_1 \times m_1}$ rettangolare superiore invertibile (siccome A ha rango massimo)

¹Come in precedenza si suppone che sia A che B abbiano rango massimo, uguale rispettivamente a m_1 e m_2

•

$$B = Q_B R_B = [Q_{1B}, Q_{2B}] \begin{bmatrix} R_{1B} \\ 0 \end{bmatrix} = Q_{1B} R_{1B}$$

dove $Q_B \in \mathbb{R}^{n_2 \times n_2}$, tale che $Q_{1B} \in \mathbb{R}^{n_2 \times m_2}$, $Q_{2B} \in \mathbb{R}^{n_2 \times (n_2 - m_2)}$ e $R_{1B} \in \mathbb{R}^{m_2 \times m_2}$ rettangolare superiore invertibile (siccome B ha rango massimo)

Si ha dunque:

$$\begin{aligned} \|b_1 b_2^T - A X B^T\|_F^2 &= \|b_1 b_2^T - Q_A R_A X R_B^T Q_B^T\|_F^2 = \|Q_A (Q_A^T b_1 b_2^T Q_B - R_A X R_B^T) Q_B^T\|_F^2 = \\ &= \|Q_A^T b_1 b_2^T Q_B - R_A X R_B^T\|_F^2 = \left\| \begin{bmatrix} Q_{1A}^T b_1 b_2^T Q_{1B} & Q_{1A}^T b_1 b_2^T Q_{2B} \\ Q_{2A}^T b_1 b_2^T Q_{1B} & Q_{2A}^T b_1 b_2^T Q_{2B} \end{bmatrix} - \begin{bmatrix} R_{1A} X R_{1B}^T & 0 \\ 0 & 0 \end{bmatrix} \right\|_F^2 = \\ &= \|Q_{1A}^T b_1 b_2^T Q_{1B} - R_{1A} X R_{1B}^T\|_F^2 + \|Q_{1A}^T b_1 b_2^T Q_{2B}\|_F^2 + \|Q_{2A}^T b_1 b_2^T Q_{1B}\|_F^2 + \|Q_{2A}^T b_1 b_2^T Q_{2B}\|_F^2 \end{aligned}$$

Siccome le matrici R_{1A} e R_{1B} sono non singolari, il problema di minimo matriciale (1.2) ha un'unica soluzione data da: $X = R_{1A}^{-1} (Q_{1A}^T b_1 b_2^T Q_{1B}) R_{1B}^{-T}$.

Per tale X vale:

$$\min_{X \in \mathbb{R}^{m_1 \times m_2}} \|b_1 b_2^T - A X B^T\|_F^2 \equiv \|Q_{1A}^T b_1 b_2^T Q_{2B}\|_F^2 + \|Q_{2A}^T b_1 b_2^T Q_{1B}\|_F^2 + \|Q_{2A}^T b_1 b_2^T Q_{2B}\|_F^2 \quad (1.3)$$

□

Osservazione 1.4.1. Se si definiscono $x_1 = R_{1A}^{-1} (Q_{1A}^T b_1)$ e $x_2 = R_{1B}^{-1} (Q_{1B}^T b_2)$, si nota che $X = x_1 x_2^T$, il che è coerente con la struttura del problema, infatti si sta cercando di approssimare, tramite trasformazioni lineari sui sottospazi generati da A e B , una matrice di rango 1, ciò implica che la soluzione migliore nel senso di Frobenius, tra tutte le matrici ottenibili come $A X B^T$, è quella più vicina a $b_1 b_2^T$, e che quindi l'approssimazione migliore preserva il rango 1 del termine noto.

1.5 La funzione minquad_house_matriciale

In questa sezione si descrive la struttura e il funzionamento della funzione `minquad_house_matriciale` per la risoluzione del problema matriciale ai minimi quadrati (1.2).

```
function [X]=minquad_house_matriciale(A,B,b1,b2)
% risolve in dimensioni piccole min_X ||b1b2^T - AXB^T||,
% applico le trasformazioni di householder sulle colonne di A
% e sulle righe di B^T in modo da ridurre il problema
% iniziale a R1XR2^T dove le due matrici rettangolari mi
% provengono dalla ortogonalizzazione

[n1,m1]=size(A); [n2,m2]=size(B);

R1=A; R2=B';

b2=b2';
```

Come visto in precedenza definiamo le matrici R_A, R_B^T ponendole rispettivamente uguali ad A e B^T , in modo tale che andando a modificarle tramite trasformazioni ortogonali a fine di entrambi i cicli si ottiene $R_A = \begin{bmatrix} R_{1A} \\ 0 \end{bmatrix}$ e $R_B^T = \begin{bmatrix} R_{1B}^T & 0 \end{bmatrix}$ dove R_{1B}^T triangolare inferiore

```

for k=1:m1
% lavoro sulle colonne di A
    x=R1(k:n1,k);

% creo il vettore di householder
    alpha=-sign(x(1))*norm(x);
    v=x-alpha*eye(n1-k+1,1);
    beta=2/(v'*v);

% applico la matrice householder da sinistra ad A e b1
    R1(k:n1,k:m1)=R1(k:n1,k:m1)-v*(beta*(v'*R1(k:n1,k:m1)));
    b1(k:n1)=b1(k:n1)-v*(beta*(v'*b1(k:n1)));
end

```

Il ciclo sopra descritto è la procedura di Householder già discussa nella sezione 1.3.

```

for k=1:m2
% lavoro sulle righe di B'
    x=R2(k,k:n2);

% creo il vettore di householder
    alpha=-sign(x(1))*norm(x);
    v=x'-alpha*eye(n2-k+1,1);
    beta=2/(v'*v);

% applico la matrice householder da destra a B^T e b2^T
    R2(k:m2,k:n2)=R2(k:m2,k:n2)-beta*(R2(k:m2,k:n2)*v)*v';
    b2(k:n2)=b2(k:n2)-beta*(b2(k:n2)*v)*v';
end

```

Applicare la k -esima matrice di Householder da destra annulla tutti gli $(n - k + 1)$ coefficienti della riga k -esima, infatti nella procedura di ortogonalizzazione standard α e v sono scelti in modo tale che $Px = [\alpha, 0, \dots, 0]^T$, allora con gli stessi α e v , applicando da destra si ottiene $(Px)^T = x^T P^T = x^T P = [\alpha, 0, \dots, 0]$, da cui: $(Q_B^T B)^T = R_B^T \iff B^T Q_B = R_B^T \iff B^T = R_B^T Q_B^T$, che è il risultato analogo a quello che avremo ottenuto operando in maniera standard su B e poi trasponendo.

```

% risoluzione sistemi triangolari
    x1=RisolviTriangSup(R1(1:m1,1:m1),b1(1:m1));
    x2=RisolviTriangSup(R2(1:m2,1:m2)',b2(1:m2)');

% costruzione della soluzione
    X=x1*x2';
end

```

Infine si risolvono i due sistemi triangolari superiori invertibili e si ottiene la soluzione così come descritto in 1.4.1.

Esempio 1.2. Consideriamo il seguente script MATLAB:

```
% prova di minquad_house_matriciale

A=[2 -1 3; 0 4 1; 5 2 -2; 1 0 7; -3 5 6]; % matrice 5x3
B=[1 0; 1 1; 1 -1; 0 1]; % matrice 4x2
b1=[1; 1; 8; -11; 0];
b2=[1; -1; 0 ; 4];

eq_norm = (A'*b1)*(b2'*B) - A'*(A*X*B')*B;

X = minquad_house_matriciale(A,B,b1,b2);

Xtrue = A \ b1*b2' / B';

Errore = norm(X - Xtrue, 'fro');

residuo = norm(b1*b2' - A*X*B', 'fro');

residuo_vero = norm(b1*b2' - A*Xtrue*B', 'fro');

residuo_normale = norm(eq_norm, 'fro');
```

Eseguito questo script, otteniamo il seguente output:

```
Errore Frobenius: 9.684e-16

Residuo Frobenius: 5.427e+01

Residuo Frobenius backslash: 5.427e+01

Residuo Frobenius Eq. Normale: 1.154e-13
```

Qui:

- **Errore Frobenius** indica la differenza tra la matrice **X** restituita dall'algoritmo implementato e la soluzione **Xtrue** calcolata tramite la funzione **backslash** di MATLAB, che rappresenta una soluzione di riferimento.
- I valori di **Residuo Frobenius** e **Residuo Frobenius backslash** risultano praticamente identici, confermando la correttezza della soluzione approssimata.
- Il valore di **Residuo Frobenius Eq. Normale**, ottenuto dall'equazione normale, è prossimo alla macchina zero, come previsto teoricamente: ciò avvalorava la correttezza dell'approccio tramite fattorizzazione QR, dove la minimizzazione è equivalente alla risoluzione delle equazioni normali.

Capitolo 2

Algoritmo LSQR

Si presenta ora un metodo iterativo per la risoluzione del sistema sovradeterminato $Ax = b$ e il problema (1.1), dove la matrice $A \in \mathbb{R}^{n \times m}$, con $n \geq m$ ($n, m \gg 100$), e sparsa.

Il metodo LSQR si basa sulla procedura di bidiagonalizzazione di Golub-Kahan e genera una successione di iterate $\{x_k\}$ tali che la norma del residuo $\|r_k\|_2$, con $r_k = b - Ax_k$, decresce in modo monotono.

Si evidenzia il fatto che tale metodo è matematicamente equivalente all'applicazione del metodo dei gradienti coniugati al sistema normale associato, ovvero

$$A^T Ax = A^T b$$

nel senso che genera esattamente gli stessi sottospazi di Krylov e quindi anche la stessa iterata x_k al passo k . La differenza principale è che tali sottospazi vengono costruiti in maniera diversa: invece di lavorare esplicitamente con $A^T A$, che può essere mal condizionata e distruggere la sparsità, LSQR costruisce le basi ortonormali utilizzando solo prodotti matrice-vettore con A e A^T , mantenendo quindi una maggiore stabilità numerica e sfruttando l'eventuale struttura sparsa di A .

2.1 Bidiagonalizzazione di Golub-Kahan

Si presentano due varianti della procedura di bidiagonalizzazione, **Bidiag 1** e **Bidiag 2**, che si differenziano essenzialmente per la scelta del vettore iniziale. Si illustrerà in seguito la relazione tra le due diverse procedure.

- **Bidiag 1**

Si sceglie il vettore dei termini noti b come vettore iniziale:

$$\beta_1 u_1 = b, \quad \alpha_1 v_1 = A^T u_1. \quad (2.1)$$

$$\left. \begin{aligned} \beta_{i+1} u_{i+1} &= A v_i - \alpha_i u_i, \\ \alpha_{i+1} v_{i+1} &= A^T u_{i+1} - \beta_{i+1} v_i \end{aligned} \right\} \quad i = 1, 2, \dots \quad (2.2)$$

Gli scalari $\alpha_i \geq 0$ e $\beta_i \geq 0$ sono scelti in modo che $\|u_i\| = \|v_i\| = 1$.
Con le definizioni:

$$\begin{aligned} U_{k+1} &\equiv [u_1, u_2, \dots, u_{k+1}] \in \mathbb{R}^{n \times (k+1)} \\ V_k &\equiv [v_1, v_2, \dots, v_k] \in \mathbb{R}^{m \times k} \end{aligned}, B_k \equiv \begin{bmatrix} \alpha_1 & & & & \\ \beta_2 & \alpha_2 & & & \\ & \beta_3 & \ddots & & \\ & & \ddots & \ddots & \\ & & & \ddots & \alpha_k \\ & & & & \beta_{k+1} \end{bmatrix} \in \mathbb{R}^{(k+1) \times k},$$

le relazioni in (2.2) possono essere riscritte come:

$$U_{k+1}(\beta_1 e_1) = b, \quad (2.3)$$

$$AV_k = U_{k+1}B_k, \quad (2.4)$$

$$A^T U_{k+1} = V_k B_k + \alpha_{k+1} v_{k+1} e_{k+1}^T. \quad (2.5)$$

In aritmetica esatta si avrebbe $U_{k+1}^T U_{k+1} = I_{k+1}$ e $V_k^T V_k = I_k$, ma quanto scritto rimane valido anche entro la precisione macchina.

• Bidiag 2

Si sceglie il vettore $A^T b$ come vettore iniziale:

$$\theta_1 v_1 = A^T b, \quad \rho_1 p_1 = A v_1. \quad (2.6)$$

$$\left. \begin{aligned} \theta_{i+1} v_{i+1} &= A^T p_i - \rho_i v_i, \\ \rho_{i+1} p_{i+1} &= A v_{i+1} - \theta_{i+1} p_i \end{aligned} \right\} \quad i = 1, 2, \dots \quad (2.7)$$

Come sopra, gli scalari $\rho_i \geq 0$ e $\theta_i \geq 0$ sono scelti in modo che $\|p_i\| = \|v_i\| = 1$.
In questo caso se definiamo:

$$\begin{aligned} P_k &\equiv [p_1, p_2, \dots, p_k] \in \mathbb{R}^{n \times k} \\ V_k &\equiv [v_1, v_2, \dots, v_k] \in \mathbb{R}^{m \times k} \end{aligned}, R_k \equiv \begin{bmatrix} \rho_1 & \theta_2 & & & \\ & \rho_2 & \theta_3 & & \\ & & \ddots & \ddots & \\ & & & \rho_{k-1} & \theta_k \\ & & & & \rho_k \end{bmatrix} \in \mathbb{R}^{k \times k},$$

si può riscrivere (2.7) come:

$$V_k(\theta_1 e_1) = A^T b, \quad (2.8)$$

$$AV_k = P_k R_k, \quad (2.9)$$

$$A^T P_k = V_k R_k^T + \theta_{k+1} v_{k+1} e_k^T. \quad (2.10)$$

e in aritmetica esatta si avrebbe allo stesso modo $P_k^T P_k = V_k^T V_k = I_k$. Ognuna delle due procedure si può derivare dall'altra scegliendo l'appropriato vettore di partenza e scambiando A con A^T .

Osservazione 2.1.1. *[Relazione tra le Bidiagonalizzazioni]*

La principale connessione tra le due procedure di bidiagonalizzazione è che le matrici V_k sono le stesse per ognuna, e vale l'identità:

$$B_k^T B_k = R_k^T R_k$$

Si osserva inoltre che la matrice R_k è identica alla matrice che si otterrebbe dalla convenzionale fattorizzazione QR di B_k , infatti:

$$Q_{k+1}^T B_k = R_k \quad (2.11)$$

dove $Q_{k+1} \in \mathbb{R}^{(k+1) \times k}$ ha colonne ortonormali. In presenza di errori di arrotondamento, queste identità non valgono ovviamente più.

2.2 Struttura dell'algoritmo LSQR

Le quantità generate a partire da A e b mediante la procedura **Bidiag 1** saranno ora impiegate per risolvere il problema ai minimi quadrati (1.1). Consideriamo le quantità:

$$x_k = V_k y_k, \quad (2.12)$$

$$r_k = b - Ax_k, \quad (2.13)$$

$$t_{k+1} = \beta_1 e_1 - B_k y_k, \quad (2.14)$$

definite in termini di un vettore y_k tale che

$$r_k = b - Ax_k \stackrel{(2.12)}{=} b - AV_k y_k \stackrel{(2.3)+(2.4)}{=} U_{k+1} \beta_1 e_1 - U_{k+1} B_k y_k = U_{k+1} (\beta_1 e_1 - B_k y_k) = U_{k+1} t_{k+1}. \quad (2.15)$$

Considerando che U_{k+1} è limitata e teoricamente avente colonne ortonormali, e considerando che l'obiettivo è mantenere $\|r_k\|$ piccolo, risulta naturale scegliere y_k in modo tale da minimizzare $\|t_{k+1}\|$; passando infatti alle norme si ha:

$$\min_{x_k \in \text{Range}(V_k)} \|r_k\|_2 = \min_{y_k \in \mathbb{R}^k} \|U_{k+1} t_{k+1}\|_2 = \min_{y_k \in \mathbb{R}^k} \|\beta_1 e_1 - B_k y_k\|_2 \quad (2.16)$$

Computazionalmente è vantaggioso risolvere (2.16) usando la fattorizzazione QR standard di B_k , che coincide con la fattorizzazione (2.11) che collega le due procedure di bidiagonalizzazione. Quindi si ha:

$$Q_{k+1}^T \begin{bmatrix} B_k & \beta_1 e_1 \end{bmatrix} = \begin{bmatrix} R_k & f_k \\ 0 & \bar{\phi}_{k+1} \end{bmatrix} = \left[\begin{array}{cccc|c} \rho_1 & \theta_2 & & & \phi_1 \\ & \rho_2 & \theta_3 & & \phi_2 \\ & & \ddots & \ddots & \vdots \\ & & & \rho_{k-1} & \theta_k & \phi_{k-1} \\ \text{---} & \text{---} & \text{---} & \text{---} & \rho_k & \phi_k \\ & & & & & \bar{\phi}_{k+1} \end{array} \right] \quad (2.17)$$

dove $Q_{k+1} \in \mathbb{R}^{(k+1) \times (k+1)}$, $Q_{k+1}^T Q_{k+1} = I_{k+1}$, e $Q_{k+1} \equiv Q_{k,k+1} \dots Q_{2,3} Q_{1,2}$ è il prodotto di rotazioni piane costruite per eliminare gli elementi sottodiagonali $\beta_2, \beta_3, \dots$ di B_k .

Riprendendo (2.16) si ottiene:

$$\begin{aligned} \|r_k\|_2^2 &= \|\beta_1 e_1 - B_k y_k\|_2^2 = \\ &= \left\| Q_{k+1}^T \beta_1 e_1 - \begin{bmatrix} R_k \\ 0 \end{bmatrix} y_k \right\|_2^2 = \left\| \begin{bmatrix} f_k \\ \bar{\phi}_{k+1} \end{bmatrix} - \begin{bmatrix} R_k \\ 0 \end{bmatrix} y_k \right\|_2^2 = \|f_k - R_k y_k\|_2^2 + |\bar{\phi}_{k+1}|^2. \end{aligned} \quad (2.18)$$

Dalla risoluzione del sistema triangolare superiore:

$$y_k = R_k^{-1} f_k \quad (2.19)$$

segue immediatamente che

$$\min_{y_k \in \mathbb{R}^k} \|r_k\|_2 = |\bar{\phi}_{k+1}|.$$

Osservazione 2.2.2.

Per necessità espositiva si evidenzia che dopo aver risolto (2.19), il residuo vettoriale nel sistema originale (cioè il vettore $t_{k+1} = \beta_1 e_1 - B_k y_k$) è ottenuto riapplicando le rotazioni inverse:

$$t_{k+1} = Q_{k+1}^T \begin{bmatrix} 0 \\ \bar{\phi}_{k+1} \end{bmatrix}, \quad (2.20)$$

da cui segue

$$\|t_{k+1}\|_2 = |\bar{\phi}_{k+1}|.$$

Osservazione 2.2.3.

y_k in (2.19) in genere non ha nulla in comune con y_{k-1} , mentre la matrice $[R_k f_k]$ coincide con $[R_{k-1} f_{k-1}]$, estesa di una riga e una colonna. Quindi ritornando all'iterazione $\{x_k\}$, un modo di combinare (2.12) e (2.19) efficientemente è secondo:

$$x_k = V_k R_k^{-1} f_k \equiv D_k f_k, \quad \text{con } f_k = [f_{k-1}; \phi_k] \quad (2.21)$$

dove le colonne di $D_k = [d_1 \ d_2 \ \dots \ d_k]$ si possono calcolare ricorsivamente dal sistema $D_k R_k = V_k$ tramite sostituzioni successive. Posto $d_0 = x_0 = 0$, si ha:

$$d_1 \rho_1 = v_1, \quad d_1 \theta_2 + d_2 \rho_2 = v_2, \quad d_2 \theta_3 + d_3 \rho_3 = v_3, \dots$$

da cui

$$d_k = \frac{1}{\rho_k} (v_k - \theta_k d_{k-1}), \quad (2.22)$$

$$x_k = x_{k-1} + \phi_k d_k. \quad (2.23)$$

Così è necessario salvare in memoria solamente l'ultima iterata aggiornata.

Osservazione 2.2.4. [Rotazioni]

La fattorizzazione QR (2.17) è stata definita costruendo la k -esima rotazione piana $Q_{k,k+1}$

per operare sulle righe k e $(k+1)$ della matrice estesa $[B_k \ \beta_1 e_1]$ per azzerare β_{k+1} . Questo dà la seguente semplice ricorrenza:

$$\begin{bmatrix} c_k & s_k \\ s_k & -c_k \end{bmatrix} \begin{bmatrix} \bar{\rho}_k & 0 & \bar{\phi}_k \\ \beta_{k+1} & \alpha_{k+1} & 0 \end{bmatrix} = \begin{bmatrix} \rho_k & \theta_{k+1} & \phi_k \\ 0 & \bar{\rho}_{k+1} & \bar{\phi}_{k+1} \end{bmatrix}, \quad (2.24)$$

dove $\bar{\rho}_1 \equiv \alpha_1$, $\bar{\phi}_1 \equiv \beta_1$ e $\bar{\rho}_k, \bar{\phi}_k$ sono quantità intermedie che vengono successivamente scambiate con ρ_k, ϕ_k . Gli scalari c_k e s_k che costituiscono la k -esima matrice di rotazione di Givens sono definiti:

$$c_k = \frac{\bar{\rho}_k}{\sqrt{\bar{\rho}_k^2 + \beta_{k+1}^2}}, \quad s_k = -\frac{\beta_{k+1}}{\sqrt{\bar{\rho}_k^2 + \beta_{k+1}^2}}$$

Osservazione 2.2.5. [Stime di $\|r_k\|_2$ e $\|A^T r_k\|_2$]

Da (2.13) e (2.20) abbiamo

$$r_k = \bar{\phi}_{k+1} U_{k+1} Q_{k+1} e_{k+1}^T; \quad (2.25)$$

quindi, assumendo $U_{k+1}^T U_{k+1} = I$, otteniamo la stima

$$\|r_k\|_2 = \bar{\phi}_{k+1} = \beta_1 s_k s_{k-1} \dots s_1, \quad (2.26)$$

dove la scrittura di $\bar{\phi}_{k+1}$ segue da (2.24). LSQR è particolare perché non mantiene esplicitamente il vettore residuo r_k ; tuttavia, si vede che la norma $\|r_k\|$ è comunque disponibile praticamente senza alcun costo aggiuntivo. È chiaro che il prodotto dei seni (dei coefficienti delle rotazioni di Givens) decresce in modo monotono. Convergerebbe a zero se il sistema $Ax = b$ fosse compatibile, altrimenti convergerà a un limite finito positivo.

Per i problemi ai minimi quadrati una quantità più importante, come già evidenziato per il metodo QR, è $A^T r_k$, che sarebbe zero all'ultima iterazione se si lavorasse in aritmetica esatta. Da (2.25), (2.5) e (2.17) abbiamo

$$A^T r_k = \bar{\phi}_{k+1} (V_k B_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T) Q_{k+1}^T e_{k+1} \quad (2.27)$$

$$= \bar{\phi}_{k+1} V_k [R_k^T \ 0] e_{k+1} + \bar{\phi}_{k+1} \alpha_{k+1} (e_{k+1}^T Q_k e_{k+1}) v_{k+1}. \quad (2.28)$$

Il primo termine è nullo e si nota facilmente che l'elemento diagonale $(k+1)$ -esimo di Q_{k+1} è $-c_k$. Quindi otteniamo

$$A^T r_k = -(\bar{\phi}_{k+1} \alpha_{k+1} c_k) v_{k+1}, \quad (2.29)$$

da cui

$$\|A^T r_k\|_2 = \bar{\phi}_{k+1} \alpha_{k+1} |c_k|. \quad (2.30)$$

La stima (2.30) è molto utile per il residuo normale poichè si sfruttano quantità già memorizzate all'interno del ciclo escludendo operazioni matrice-vettore, o nel caso matriciale prodotti tra matrici, molto costosi per matrici di grandi dimensioni.

2.3 La funzione lsqr secondo Paige e Saunders

L'algoritmo LSQR, presentato di seguito, è stato sviluppato da Christopher C. Paige e Michael A. Saunders nel 1982 [1], ed è stato concepito per risolvere sistemi lineari di grandi dimensioni e problemi ai minimi quadrati in modo efficiente, specialmente quando la matrice dei coefficienti è sparsa.

Si riporta di seguito l'algoritmo spiegato passo dopo passo:

```
function [x,restot]=lsqr(A,b,tol,imax)

[n,m]=size(A);
beta=norm(b); u=b/beta; % vettore iniziale
res=beta;
v=A'*u; alfa=norm(v); v=v/alfa;
phi=beta; rho=alfa;

i=0; x=zeros(m,1); d=v; % inizializzazione
```

Si inizializza l'iterazione creando i primi due vettori u_1 e v_1 delle basi ortonormali così come descritto in (2.1); si crea la prima riga della matrice estesa $[B_k \ \beta_1 e_1]$ su cui agirà la prima rotazione:

$$\bar{\rho}_1 = \alpha_1, \quad \bar{\phi}_1 = \beta_1$$

e poi si impostano l'iterata x_0 iniziale e il primo vettore direzione $d_1 \bar{\rho}_1 = v_1$

```
while (i<imax && res/norm(b)>tol) % 2 criteri di arresto

    i=i+1;

% Calcolo dei vettori di base {u_k} e {v_k}
    u = A*v - alfa*u;      % Costruzione u_{k+1}
    beta = norm(u);
    u = u/beta;            % Ortonormalizzazione
    v = A'*u - beta*v;     % Costruzione v_{k+1}
    alfa = norm(v);
    v = v/alfa;            % Ortonormalizzazione
```

Come descritto in (2.2) si costruiscono i vettori u_{k+1} e v_{k+1} delle rispettive basi ortonormali che concorrono al processo di bidiagonalizzazione; si evidenziano anche i due criteri di arresto che si basano sul numero massimo di iterazioni, necessario in algoritmi iterativi per non generare infinite iterazioni, e sul residuo relativo che stima la bontà dell'approssimazione

```
% Nuova Rotazione
rho1 = (rho^2+beta^2)^(0.5);
c = rho/rho1;
s = beta/rho1;
theta = s*alfa;
rho = -c*alfa;
phi1 = c*phi;
phi = s*phi;
```


Si definisce la k -esima rotazione di Givens e si applica alle righe k e $k + 1$ della matrice $[B_k \ \beta_1 e_1]$, si opera ciò che in precedenza si è descritto in (2.24)

```
% Aggiornamento soluzione e vettore direzione
x = x + phi1/rho1*d;
d = v - theta/rho1*d;
```

Si aggiornano l'iterata e il vettore direzione

```
% Calcolo residuo
res = phi;
end
end
```

Infine si calcola il residuo, che servirà all'inizio del ciclo successivo per il criterio di arresto precedentemente descritto.

Capitolo 3

Algoritmo LSQR matriciale

3.1 Prodotto di Kronecker

Una nozione utile nello studio di equazioni di matrici è il *prodotto di Kronecker*, o *prodotto tensoriale*, di matrici.

Definizione 3.1. (*Prodotto di Kronecker*)

Il prodotto di Kronecker di $\mathbf{A} \in \mathbb{R}^{n \times m}$ e $\mathbf{B} \in \mathbb{R}^{p \times q}$, si denota $\mathbf{A} \otimes \mathbf{B}$ e si definisce come la matrice a blocchi:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1m}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{n1}\mathbf{B} & \cdots & a_{nm}\mathbf{B} \end{bmatrix} \in \mathbb{R}^{np \times mq} \quad (3.1)$$

Tra le proprietà fondamentali del prodotto di Kronecker vi sono:

- Bilinearità e Associatività

1. $A \otimes (B + C) = A \otimes B + A \otimes C$
2. $(B + C) \otimes A = B \otimes A + C \otimes A$
3. $(kA) \otimes B = A \otimes (kB) = k(A \otimes B)$
4. $(A \otimes B) \otimes C = A \otimes (B \otimes C)$
5. $A \otimes 0 = 0 \otimes A = 0$

dove A , B e C sono matrici e $k \in \mathbb{R}$.

- Non commutatività

In generale $A \otimes B$ e $B \otimes A$ sono matrici differenti. Tuttavia si può dire che $A \otimes B$ e $B \otimes A$ sono matrici equivalenti per permutazione, nel senso che esistono due matrici di permutazione P e Q , tali che:

$$B \otimes A = P(A \otimes B)Q \quad (3.2)$$

- Relazione con il prodotto matriciale

Siano A , B , C e D matrici di dimensioni tali per cui è lecito svolgere i prodotti AC e BD , allora vale:

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \quad (3.3)$$

- Invertibilità

Siano A e B matrici quadrate invertibili, allora si ha che $A \otimes B$ è invertibile e vale:

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1} \quad (3.4)$$

- Trasposizione

L'operatore di trasposizione è distributivo sul prodotto di Kronecker:

$$(A \otimes B)^T = A^T \otimes B^T \quad (3.5)$$

Definizione 3.2. (*Operazione vec*)

Data una matrice $A \in \mathbb{R}^{n \times m}$, si associa il vettore $\text{vec}(A) \in \mathbb{R}^{nm}$ definito da:

$$\text{vec}(A) = [a_{1,1}, \dots, a_{n,1}, a_{1,2}, \dots, a_{n,2}, \dots, a_{1,m}, \dots, a_{n,m}]^T$$

Osservazione 3.1.1.

Se $x \in \mathbb{R}^n$ e $y \in \mathbb{R}^m$ sono vettori arbitrari, allora il prodotto esterno tra x e y è definito come xy^T . Il prodotto di Kronecker è legato al prodotto esterno dalla relazione:

$$y \otimes x = \text{vec}(xy^T) \quad (3.6)$$

Una delle applicazioni principali del prodotto di Kronecker è la riformulazione di equazioni lineari matriciali in forma vettoriale. A tal proposito, si considera il seguente lemma, centrale per i risultati presentati in questa tesi.

Lemma 3.3. Siano $A \in \mathbb{R}^{n \times m}$, $B \in \mathbb{R}^{p \times q}$ e $C \in \mathbb{R}^{n \times q}$ e sia $X \in \mathbb{R}^{m \times p}$ incognita. Allora l'equazione matriciale:

$$AXB = C \quad (3.7)$$

è equivalente al sistema di qn equazioni in mp incognite dato da:

$$(B^T \otimes A)\text{vec}(X) = \text{vec}(C) \quad (3.8)$$

Dimostrazione. Per una generica matrice Q , denotiamo Q_k la k -esima colonna di Q . Sia $B = (b_{ij})$, allora:

$$\begin{aligned} (AXB)_k &= A(XB)_k = AXB_k = \\ &= A \left(\sum_{i=1}^p b_{ik} X_i \right) = \begin{bmatrix} b_{1k}A & b_{2k}A & \dots & b_{pk}A \end{bmatrix} \text{vec}(X) = (B_k^T \otimes A)\text{vec}(X) \end{aligned}$$

Quindi,

$$\text{vec}(AXB) = \begin{bmatrix} B_1^T \otimes A \\ \vdots \\ B_q^T \otimes A \end{bmatrix} \text{vec}(X)$$

Ma questo prodotto è semplicemente $B^T \otimes A$, poichè la trasposizione di una colonna di B è una riga di B^T ; quindi $\text{vec}(C) = \text{vec}(AXB) = (B^T \otimes A)\text{vec}(X)$, il che conclude la dimostrazione. $\square$

3.2 Algoritmo LSQR matriciale

In questa sezione si estende l'algoritmo LSQR al problema matriciale ai minimi quadrati:

$$\min_{X \in \mathbb{R}^{m_1 \times m_2}} \|\mathbf{b}_1 \mathbf{b}_2^T - \mathbf{A} \mathbf{X} \mathbf{B}^T\|_F$$

dove $\mathbf{A} \in \mathbb{R}^{n_1 \times m_1}$, $\mathbf{B} \in \mathbb{R}^{n_2 \times m_2}$, $n_1 \geq m_1$ e $n_2 \geq m_2$ ($n_1, m_1, n_2, m_2 \gg 100$), $\mathbf{b}_1 \in \mathbb{R}^{n_1}$ e $\mathbf{b}_2 \in \mathbb{R}^{n_2}$. Stante quanto osservato con il prodotto di Kronecker, torniamo a (1.2) e si consideri l'equazione matriciale:

$$\mathbf{A} \mathbf{X} \mathbf{B}^T = \mathbf{b}_1 \mathbf{b}_2^T$$

Per il Lemma precedente questa equazione è equivalente a:

$$(\mathbf{B} \otimes \mathbf{A})\text{vec}(X) = \text{vec}(\mathbf{b}_1 \mathbf{b}_2^T) \stackrel{(3.6)}{=} \mathbf{b}_2 \otimes \mathbf{b}_1$$

Ponendo:

$$\mathbf{D} := \mathbf{B} \otimes \mathbf{A} \in \mathbb{R}^{n_1 n_2 \times m_1 m_2}, \quad \bar{\mathbf{b}} := \mathbf{b}_2 \otimes \mathbf{b}_1 \in \mathbb{R}^{n_1 n_2} \quad \text{e} \quad \mathbf{x} := \text{vec}(X) \in \mathbb{R}^{m_1 m_2},$$

il problema matriciale ai minimi quadrati (1.2) diventa:

$$\min_X \|\mathbf{b}_1 \mathbf{b}_2^T - \mathbf{A} \mathbf{X} \mathbf{B}^T\|_F = \min_X \|\mathbf{b}_2 \otimes \mathbf{b}_1 - (\mathbf{B} \otimes \mathbf{A})\text{vec}(X)\|_2 = \min_x \|\bar{\mathbf{b}} - \mathbf{D}x\|_2 \quad (3.9)$$

che può essere risolto efficientemente mediante l'algoritmo LSQR precedentemente presentato.

Osservazione 3.2.1. Per completezza vediamo come ottenere il risultato della *Proposizione 1.1* usando il formalismo del prodotto di Kronecker. Innanzitutto osserviamo che:

$$\begin{aligned} \mathbf{D} = \mathbf{B} \otimes \mathbf{A} &= \begin{bmatrix} Q_{1B}, Q_{2B} \end{bmatrix} \begin{bmatrix} R_{1B} \\ 0 \end{bmatrix} \otimes \begin{bmatrix} Q_{1A}, Q_{2A} \end{bmatrix} \begin{bmatrix} R_{1A} \\ 0 \end{bmatrix} \stackrel{(3.3)}{=} \left(\begin{bmatrix} Q_{1B}, Q_{2B} \end{bmatrix} \otimes \begin{bmatrix} Q_{1A}, Q_{2A} \end{bmatrix} \right) \\ &\left(\begin{bmatrix} R_{1B} \\ 0 \end{bmatrix} \otimes \begin{bmatrix} R_{1A} \\ 0 \end{bmatrix} \right) = \left(\begin{bmatrix} Q_{1B} \otimes Q_{1A}, Q_{1B} \otimes Q_{2A}, Q_{2B} \otimes Q_{1A}, Q_{2B} \otimes Q_{2A} \end{bmatrix} \right) \begin{pmatrix} \begin{bmatrix} R_{1B} \otimes R_{1A} \\ R_{1B} \otimes 0 \\ 0 \otimes R_{1A} \\ 0 \otimes 0 \end{bmatrix} \end{pmatrix} = \\ &= \mathbf{Q}_D \mathbf{R}_D \end{aligned}$$

Osserviamo che la matrice Q_D è ortogonale, infatti:

$$(Q_B \otimes Q_A)^T (Q_B \otimes Q_A) \stackrel{(3.5)}{=} (Q_B^T \otimes Q_A^T)(Q_B \otimes Q_A) \stackrel{(3.3)}{=} (Q_B^T Q_B \otimes Q_A^T Q_A) = (I_{n_2} \otimes I_{n_1}) = I_{n_1 n_2},$$

si osserva inoltre che la matrice $R_D = R_B \otimes R_A$ mantiene la struttura triangolare superiore. La fattorizzazione ottenuta è esattamente la fattorizzazione QR di D .

Quindi si ha:

$$\begin{aligned} \|\bar{b} - Dx\|_2^2 &= \|\bar{b} - Q_D R_D x\|_2^2 = \|Q_D^T \bar{b} - R_D x\|_2^2 = \left\| \begin{bmatrix} Q_{1B}^T \otimes Q_{1A}^T \\ Q_{1B}^T \otimes Q_{2A}^T \\ Q_{2B}^T \otimes Q_{1A}^T \\ Q_{2B}^T \otimes Q_{2A}^T \end{bmatrix} \bar{b} - \begin{bmatrix} R_{1B} \otimes R_{1A} \\ 0 \\ 0 \\ 0 \end{bmatrix} x \right\|_2^2 = \\ &= \|(Q_{1B}^T \otimes Q_{1A}^T)(b_2 \otimes b_1) - (R_{1B} \otimes R_{1A})x\|_2^2 + \|(Q_{1B}^T \otimes Q_{2A}^T)(b_2 \otimes b_1)\|_2^2 + \\ &\quad + \|(Q_{2B}^T \otimes Q_{1A}^T)(b_2 \otimes b_1)\|_2^2 + \|(Q_{2B}^T \otimes Q_{2A}^T)(b_2 \otimes b_1)\|_2^2 \end{aligned}$$

Siccome R_{1A}, R_{1B} sono invertibili per (3.4) si ha che $R_{1B} \otimes R_{1A}$ è invertibile, da cui:

$$\begin{aligned} x &= (R_{1B} \otimes R_{1A})^{-1} (Q_{1B}^T \otimes Q_{1A}^T)(b_2 \otimes b_1) \stackrel{(3.3)+(3.4)}{=} \\ &= (R_{1B}^{-1} \otimes R_{1A}^{-1})(Q_{1B}^T b_2 \otimes Q_{1A}^T b_1) \stackrel{(3.3)}{=} [R_{1B}^{-1}(Q_{1B}^T b_2)] \otimes [R_{1A}^{-1}(Q_{1A}^T b_1)] = x_2 \otimes x_1 = x_1 x_2^T \end{aligned}$$

Per tale x vale:

$$\min_{x \in \mathbb{R}^{m_1 m_2}} \|\bar{b} - Dx\|_2^2 = \|(Q_{1B}^T \otimes Q_{2A}^T)(b_2 \otimes b_1)\|_2^2 + \|(Q_{2B}^T \otimes Q_{1A}^T)(b_2 \otimes b_1)\|_2^2 + \|(Q_{2B}^T \otimes Q_{2A}^T)(b_2 \otimes b_1)\|_2^2$$

Svolgendo i prodotti seguendo le regole del prodotto di Kronecker si arriva esattamente al risultato trovato in (1.3). Osserviamo che, in analogia con il caso lineare, il residuo ottenuto corrisponde alla proiezione di $b_2 \otimes b_1$ nello spazio ortogonale alle colonne di D , cioè in $\text{Range}(B \otimes A)^\perp = \text{Range}(Q_B \otimes Q_A)^\perp$.

Illustriamo di seguito l'algoritmo LSQR per la risoluzione del problema matriciale (1.2), osservando che, siccome le matrici A e B sono di grandi dimensioni, non è efficiente svolgere le operazioni matrice-vettore con D e D^T , nè tanto meno creare la matrice D che è computazionalmente molto costoso (se $A, B \in \mathbb{R}^{n \times n}$, allora $A \otimes B \approx \mathcal{O}(n^4)$).

Si mette in evidenza che da quanto osservato nel capitolo precedente il residuo non tenderà a zero, mentre il residuo proveniente dalla equazione normale sì, per questo motivo all'interno dell'algoritmo LSQR matriciale, uno dei due criteri di arresto sfrutta la norma del residuo normale. Preciso ciò, procediamo con la spiegazione dell'algoritmo `lsqr_matriciale`:

```

function [X]=lsqr_matriciale(A,B,b1,b2,imax,tol)

[~,m1]=size(A); [~,m2]=size(B);

b=b1*b2';
normb=norm(b,'fro');

beta=normb;
U=b/beta; % matrice iniziale
res=beta; % residuo iniziale

V = A' * U * B; % D'*u = kron(B',A')vec(U) = vec(A'UB)
alpha=norm(V, 'fro');
V=V/alpha;

phi=beta; rho=alpha;

i=0; X=zeros(m1,m2); D=V;

```

L'inizializzazione prevede la creazione della matrice di rango 1, che sarebbe il termine noto del problema, e si prosegue con la creazione delle prime due matrici U_1 e V_1 , che, come nel caso vettoriale, costituiscono i primi due vettori delle rispettive basi ortonormali che si costruiscono nella procedura di bidiagonalizzazione, così come descritto in (2.1). Nonostante nel commento a latere si abbia la formulazione vettoriale con il `vec`, che nell'ambiente MATLAB si sarebbe tradotto con la funzione *reshape*, a livello pratico, quando le matrici da gestire hanno dimensioni molto alte, è più efficiente lavorare con moltiplicazioni matriciali, anzichè vettorializzare matrici molto grandi. Successivamente si lavora come l'algoritmo LSQR lineare. Per il monitoraggio del residuo normale si considera l'equazione (2.30) spiegata nel capitolo precedente.

```

while (i<imax && res/normb>tol)

    i=i+1;

    % Calcolo delle matrici di base {U_{k+1}} e {V_{k+1}}
    % D*v = kron(B,A)vec(V) = vec(AVB')
    U = A * V * B' - alpha*U; % u = D*v - alpha*u
    beta = norm(U,'fro');
    U = U/beta;

    % D'*u = kron(B',A')vec(U) = vec(A'UB)
    V = A' * U * B - beta*V; % v = D'*u - beta*v
    alpha = norm(V,'fro');
    V = V/alpha;

    % Nuova Rotazione
    rho1 = (rho^2+beta^2)^(0.5);
    c = rho/rho1;
    s = beta/rho1;
    theta = s*alpha;
    rho = -c*alpha;
    phi1 = c*phi;

```

```

        phi = s*phi;

% Aggiornamento soluzione e vettore direzione
    X = X + phi1/rho1*D;
    D = V - theta/rho1*D;

```

Il ciclo è identico a quello descritto per il caso lineare

```

% Calcolo residuo
    % res = phi; residuo lsqr
    res = phi*alpha*abs(c); % residuo equazione normale
end

```

Per quanto riguarda il residuo, si considera il residuo normale, come descritto in precedenza, anzichè quello derivante dall'algoritmo LSQR. A prova di ciò, osserviamo con un esempio la differenza nell'utilizzo dei due differenti residui in termini di velocità di convergenza:

```

tol=1e-6; imax=1000;

A=randn(500,300);
B=randn(400,200);
b1=randn(500,1);
b2=randn(400,1);

tic
[residuo_norm,X1]=lsqr_matriciale(A,B,b1,b2,imax,tol);
time_residuo_norm=toc;
tic
[residuo_lsqr,X2]=lsqr_matriciale_standard(A,B,b1,b2,imax,
    tol);
time_residuo_lsqr=toc;

Xtrue= (A\b1)*(b2'/B');

Err_norm = norm(X1-Xtrue,'fro')
Err_lsqr = norm(X2-Xtrue,'fro')

```

Eseguendo questo script, otteniamo il seguente output:

```

Tempo algoritmo LSQR con residuo normale: 1.935e+00
Tempo algoritmo LSQR con residuo standard: 4.447e+00
Errore_convergenza_normale: 5.657e-08
Errore_convergenza_lsqr: 6.639e-15

Rango di X1: 190
Rango di X2: 1

```

Dalla stima dell'errore si osserva che la soluzione ottenuta con il residuo LSQR standard risulta più accurata, come confermato anche dal rango delle matrici calcolate. Il grafico di convergenza (3.1) mostra che il residuo LSQR non tende a zero, come previsto dalla teoria: esso rappresenta infatti il residuo del problema ai minimi quadrati, che si annulla

solo in presenza di un sistema compatibile. L'algoritmo termina quindi per raggiungimento del numero massimo di iterazioni, non a causa di una lentezza di convergenza, ma perché il criterio basato sul residuo standard non è il più adatto a monitorare la convergenza in questo contesto. Il residuo normale, invece, tende correttamente a zero, indicando che la soluzione ottenuta soddisfa le equazioni normali con la precisione numerica attesa.

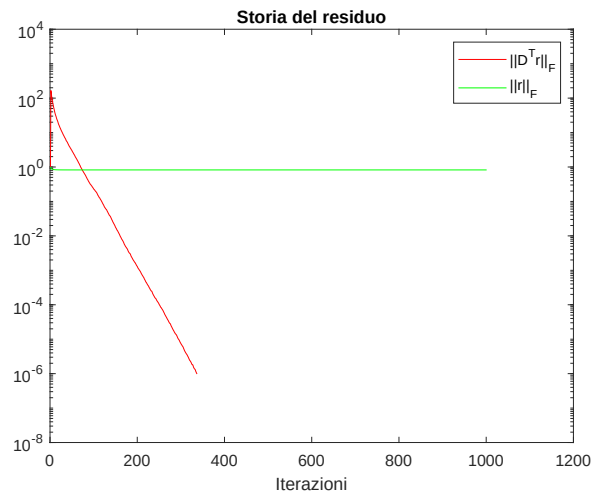


Figura 3.1: Convergenza dei due algoritmi LSQR

Capitolo 4

Applicazione dei Minimi Quadrati all'Information Retrieval

4.1 Cos'è l'Information Retrieval?

L'*Information Retrieval* (IR) si occupa della rappresentazione, della ricerca e della manipolazione di grandi raccolte di testi elettronici. I sistemi e i servizi di IR sono ormai diffusi, con milioni di persone che vi fanno affidamento quotidianamente per facilitare attività di business, istruzione e intrattenimento. I motori di ricerca web – Google, Bing e altri – sono di gran lunga i servizi di IR più popolari e utilizzati, poiché forniscono accesso a informazioni tecniche aggiornate, consentono di trovare persone e organizzazioni, riassumono notizie ed eventi e semplificano il confronto tra prodotti. I sistemi di biblioteche digitali aiutano i ricercatori in ambito medico e accademico a conoscere nuovi articoli scientifici e presentazioni a conferenze relative ai loro campi di ricerca. I consumatori si rivolgono ai servizi di ricerca locale per trovare rivenditori che offrono i prodotti e servizi desiderati. All'interno delle grandi aziende, i sistemi di ricerca aziendale fungono da archivi per e-mail, promemoria, relazioni tecniche e altri documenti aziendali, fornendo una memoria aziendale grazie alla conservazione di tali documenti e consentendo l'accesso alle conoscenze in essi contenute. I sistemi di ricerca desktop permettono agli utenti di cercare tra le proprie e-mail, documenti e file personali.

4.2 Un esempio pratico

Si presenta di seguito uno dei metodi standard per la rappresentazione di documenti in un sistema di IR, chiamato modello vettoriale, dove sia le *query*, cioè le richieste dell'utente, che i documenti, sono rappresentati come vettori in uno spazio vettoriale ad alta dimensione, in cui ciascun componente del vettore corrisponde alla frequenza di un termine del vocabolario della collezione e vale $k \in \mathbb{N}$ se il termine è presente k volte, 0 altrimenti.

Consideriamo il seguente esempio preso da [3]:

Parola chiave	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5
eigenvalue	0	0	0	1	0
England	0	0	0	0	1
FIFA	0	0	0	0	1
Google	1	0	1	0	0
Internet	1	0	0	0	0
Link	0	1	0	0	0
matrix	1	0	1	1	0
page	0	1	1	0	0
rank	0	0	1	1	1
Web	0	1	1	0	0

Figura 4.1: Matrice termine-documento

Chiamiamo A la matrice "termine-documento" sopra rappresentata. Sulle righe si ha la collezione di termini, mentre sulle colonne la collezione dei documenti che compongono il *dataset*:

- Doc 1: The **Google matrix** P is a model of the **Internet**
- Doc 2: $P_{i,j}$ is nonzero if there is a **link** from **Web page** j to i
- Doc 3: The **Google matrix** is used to **rank** all **Web pages**
- Doc 4: The **ranking** is done by solving a **matrix eigenvalue** problem
- Doc 5: **England** dropped out of the top 10 in the **FIFA ranking**

Simuliamo ora una richiesta dell'utente:

1. "**ranking of Web pages**", a cui corrisponde il *query vector* $b_1^T = [0, 0, 0, 0, 0, 0, 0, 1, 1, 1]^T$
2. "**complains in England about FIFA**", a cui corrisponde il *query vector* $b_2^T = [0, 1, 1, 0, 0, 0, 0, 0, 0, 0]^T$

La richiesta è considerata come un documento e il problema di capire quanto la richiesta possa essere rappresentata dai documenti disponibili si formalizza in un problema ai minimi quadrati: Per ogni b_i , $i = 1, 2$, determina x_i tale che

$$\min_x \|b_i - Ax_i\|_2$$

Osservando le soluzioni:

	x_1	x_2
	-4.4444e-01	2.2222e-01
	2.2222e-01	1.6162e-01
	6.6667e-01	-2.4242e-01
	8.7591e-17	-1.8182e-01
	1.1111e-01	8.0808e-01
$\frac{\ b_i - Ax_i\ }{\ b_i\ }$	3.8490e-01	4.3809e-01

Figura 4.2: Soluzioni relative a queries diverse

Dal momento che $b_1 \approx \text{Doc1}(x_1)_1 + \text{Doc2}(x_1)_2 + \text{Doc3}(x_1)_3 + \text{Doc4}(x_1)_4 + \text{Doc5}(x_1)_5$, notiamo subito che il Doc 4 non è pertinente alla *query* espressa, mentre il Doc 3 è quello che meglio la rappresenta.

L'obiettivo di questo elaborato non è quello di produrre un *ranking* dei documenti né di definire misure di similarità tra query e documenti, come avviene nei tradizionali sistemi di IR, bensì lo scopo è dimostrare come un problema ai minimi quadrati generalizzato possa essere applicato per rappresentare una query rispetto a due *dataset* tematicamente distinti. I capitoli precedenti hanno fornito gli strumenti numerici necessari — in particolare la variante dell'algoritmo LSQR — che qui vengono impiegati unicamente come mezzo per modellare la pertinenza in termini di proiezione nei rispettivi spazi termine-documento, senza alcuna finalità di *clustering* o ordinamento dei risultati.

4.3 Il dataset Reuters-21578

I dati contenuti nel corpus “Reuters-21578” consistono in una delle raccolte più influenti e ampiamente utilizzate di articoli di agenzia provenienti dal servizio di notizie finanziarie Reuters apparse nel 1987. I dati sono stati originariamente etichettati da Carnegie Group, Inc. e Reuters, Ltd. e sono stati successivamente raccolti e formattati da David Lewis con l'aiuto di diverse altre persone. Una versione precedente della raccolta, nota come Reuters-22173, è stata utilizzata in una serie di studi pubblicati fino al 1996, quando una revisione della raccolta ha portato alla correzione di diversi altri errori e alla rimozione di 595 duplicati dal set originale di 22.173 documenti, lasciando così i 21.578 documenti che ora compongono Reuters-21578.

Il dataset Reuters-21578, è un punto di riferimento essenziale per i sistemi di Information Retrieval. Questo ampio archivio fornisce una serie di preziose informazioni sugli argomenti trattati frequentemente dalle pubblicazioni finanziarie ed è disponibile in più sezioni per un'esplorazione ottimale tramite machine learning.

Il corpus Reuters-21578 contiene 21578 documenti divisi in 135 categorie. Forniamo qui la versione ModApté. I documenti con più etichette vengono scartati, cioè i documenti appartenenti a più ambiti tematici. Il risultato è 8293 documenti divisi in 65 categorie. Caricando il file contenente il dataset, otteniamo i seguenti dati:

- **fea** matrice documento-termine sparsa di dimensioni 8293×18933 , dove ogni riga rappresenta un documento e ogni colonna un termine del vocabolario;

- **gnd** vettore 8293 x 1 che indica il gruppo tematico di appartenenza di ogni documento.

Per l'analisi, selezioniamo i primi due gruppi tematici, che insieme contano più della metà dei documenti, a cui rispettivamente sono associate le label 'earn', che sta per 'earnings' (i cui documenti si riferiscono a profitti aziendali, risultati finanziari trimestrali, bilanci) e 'acq', che sta per 'acquisitions' (i cui documenti si riferiscono a fusioni, acquisizioni, vendite di aziende o asset), li denotiamo come **g1** e **g2** ottenendo, tramite trasposizione, le rispettive matrici termine-documento:

$$\mathbf{g1} = \text{fea}(1:3713, :)', \quad \mathbf{g2} = \text{fea}(3714:5768, :)'$$

dove gli indici di riga sono stati scelti in base al vettore **gnd**, che indica l'appartenenza dei documenti a ciascun gruppo tematico.

Successivamente, fissiamo un numero limitato di documenti osservabili in ciascuno dei due gruppi, denotati con $n = 25$ e $m = 25$, e definiamo A e B come le matrici termine-documento ottenute tramite la trasposizione delle matrici **g1** e **g2** (stiamo considerando i primi 25 documenti di **g1** e i primi 25 di **g2**).

Per risolvere il problema ai minimi quadrati generalizzato, scegliamo un documento arbitrario all'interno di **fea** che denotiamo c , per fungere da query. Analizzeremo diverse scelte di c e i rispettivi risultati.

Infine, fissiamo una tolleranza adeguata per garantire una soluzione verosimile e scegliamo un numero arbitrario di iterazioni, in modo che l'algoritmo non termini per raggiungimento del numero massimo di iterazioni.

Procediamo a risolvere il problema generalizzato ai minimi quadrati:

$$\min_X \|\mathbf{c}\mathbf{c}^T - \mathbf{A}\mathbf{X}\mathbf{B}^T\|_F \quad (4.1)$$

ed esaminiamo le seguenti casistiche:

Caso 0 : c un documento del range di A (analogo con B)

Caso 1 :

1a : c un documento di **g1**, non appartenente al range di A

1b : c un documento di **g2**, non appartenente al range di B

Caso 2 : c un documento non appartenente ai gruppi tematici **g1** e **g2**

Lo strumento numerico che usiamo per verificare se X sia più proiettata verso A che verso B o viceversa, è la decomposizione SVD di X ; poichè X tenderà a una matrice di rango 1, è sufficiente approssimare X con la prima tripletta della sua decomposizione SVD, cioè $X \approx u_1 \sigma_1 v_1^T$. Con questa approssimazione otteniamo che

$$\mathbf{A}\mathbf{X}\mathbf{B}^T - \mathbf{c}\mathbf{c}^T \approx \mathbf{A}u_1\sigma_1v_1^T\mathbf{B}^T - \mathbf{c}\mathbf{c}^T = \mathbf{A}x_1x_2^T\mathbf{B}^T - \mathbf{c}\mathbf{c}^T = \begin{bmatrix} \mathbf{A}x_1 & c \end{bmatrix} \begin{bmatrix} x_2^T\mathbf{B}^T \\ -\mathbf{c}^T \end{bmatrix} \quad (4.2)$$

¹ $x_1 = u_1\sqrt{\sigma_1}, x_2 = v_1\sqrt{\sigma_1}$

Dalla precedente espressione e da quanto spiegato nei capitoli precedenti, risolvere (4.1) è equivalente alla risoluzione dei due problemi vettoriali ai minimi quadrati e al conseguente prodotto esterno delle soluzioni, quindi, se ad esempio x_1 è una buona approssimazione di c tramite la matrice termine-documento A , mentre x_2 non lo è tramite B , allora questo implicherà che la matrice risultante X sarà maggiormente condizionata dalle componenti di x_1 più grandi in modulo rispetto a quelle di x_2 ; in altre parole l'utilizzo della SVD troncata alla prima tripletta su X permette di recuperare i vettori soluzione dei due rispettivi problemi ai minimi quadrati (che a meno di moltiplicazione della costante $\sqrt{\sigma_1}$ sono esattamente i primi vettori singolari) e valutarne le rispettive componenti, permette di identificare i documenti più importanti di ciascuna matrice termine-documento (così come nel precedente esempio) e soprattutto permette di verificare se la query è meglio rappresentata dai documenti di una collezione o dell'altra.

Alle casistiche sopra citate si allegano i seguenti grafici:

- (a): Confronto in magnitudine tra il primo vettore singolare sinistro e il primo vettore singolare destro di X eseguito con la seguente modalità: ad ogni iterazione si calcola la prima tripletta della SVD di X , si considerano le tre componenti maggiori in valore assoluto di u_1 e v_1 e si calcola la norma di questi vettori 3×1 ; salvando i risultati per ogni iterazione si determina il grafico che illustrerà se X sia meglio rappresentata da A oppure da B in base a quanto detto in precedenza;
- (b): Confronto dei primi due valori singolari all'aumentare delle iterazioni con lo scopo di verificare se l'algoritmo stia effettivamente creando una matrice di rango 1 (si prevede che durante l'algoritmo il secondo valore singolare σ_2 tenda ad annullarsi);
- (c): Convergenza del residuo normale a zero per verificare se X è di fatto soluzione del sistema normale associato, il che è equivalente alla soluzione di (4.1);
- (d): Plot di $|u_1| = (|u_{1_1}|, |u_{1_2}|, \dots)$ e di $|v_1|$ per evinanziare i documenti più pertinenti di A e di B in base alla query scelta c .

4.3.1 Caso 0

Consideriamo il caso in cui la query appartiene al range di una delle due matrici termine-documento. Poniamo dunque $c = A_1$ e calcoliamo la matrice X soluzione del problema (4.1).

I risultati ottenuti sono illustrati nei grafici seguenti:

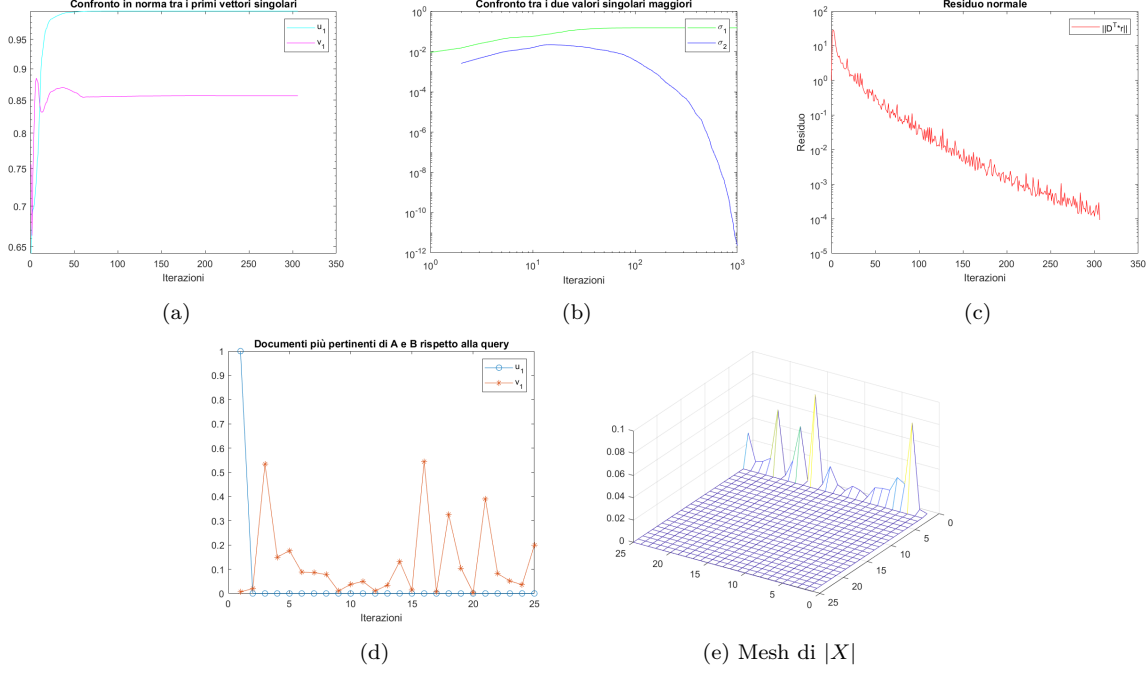


Figura 4.3: $c = A_1$

Con l'approssimazione di rango 1 $X \approx u_1 \sigma_1 v_1^T$ si osserva che $u_1 \approx e_1$, ciò implica che la prima riga della matrice X è la sola significativamente diversa da zero, come si evince dal grafico (4.3e); il comportamento di u_1 riflette infatti il ruolo dominante del primo documento di A nella rappresentazione della query scelta.

La figura (4.3a) evidenzia la maggiore importanza di u_1 rispetto a v_1 nella determinazione di X , come previsto dalla trattazione, e come ugualmente previsto, da (4.3b) si nota che il secondo valore singolare σ_2 tende a zero, ossia che X tende a una matrice di rango 1. Nonostante X sia più proiettata verso A , osserviamo da (4.3d) che tra i documenti presenti in B , quelli più pertinenti alla scelta della query sono il 3°, il 16°, il 18° e il 21°.

4.3.2 Caso 1

- **Caso 1a:** consideriamo il caso in cui la query appartiene allo stesso gruppo tematico da cui si è estrapolata la matrice A , ma non appartenente al range di A . Poniamo dunque $c = g1(n+1, :)$ e riportiamo i grafici risultanti:

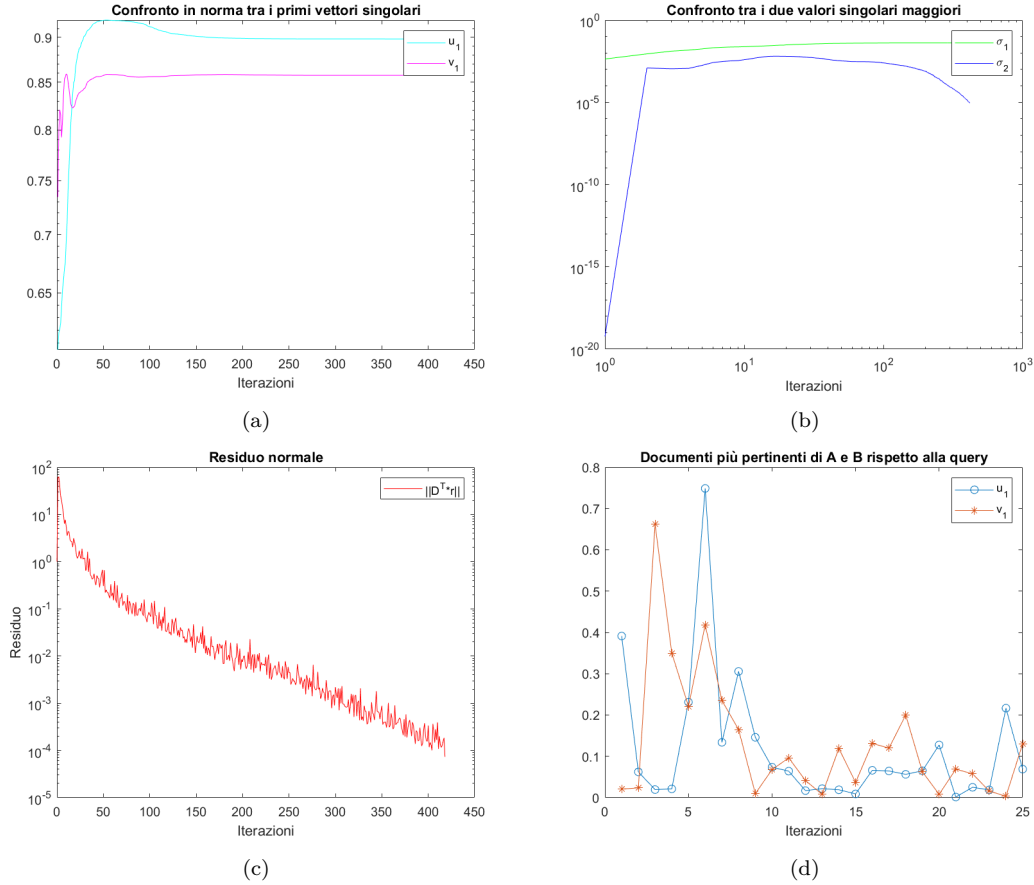


Figura 4.4: $c = g1(n+1, :)$

Da (4.4a) si può notare che la soluzione X è più proiettata verso A , il che è coerente con il fatto che si sta prendendo come query un documento appartenente allo stesso gruppo tematico da cui si è definita la matrice termine-documento A ; si sottolinea comunque che non c'è una netta dominanza verso u_1 rispetto a v_1 , infatti considerando figura (4.4d) si nota come il 3° documento di B sia molto pertinente alla query c , ciò che rende la matrice A più pertinente alla query in esame è il 6° documento. Osservando la figura (4.4b), si evidenzia la regolare convergenza della matrice X verso una matrice di rango 1.

- **Caso 1b:** consideriamo il caso in cui la query appartiene allo stesso gruppo tematico da cui si è estrapolata la matrice B , ma non appartenente al range di B . Poniamo dunque $c = g2(m+1, :)$ e calcoliamo la matrice X .

I risultati ottenuti sono illustrati nei grafici seguenti:

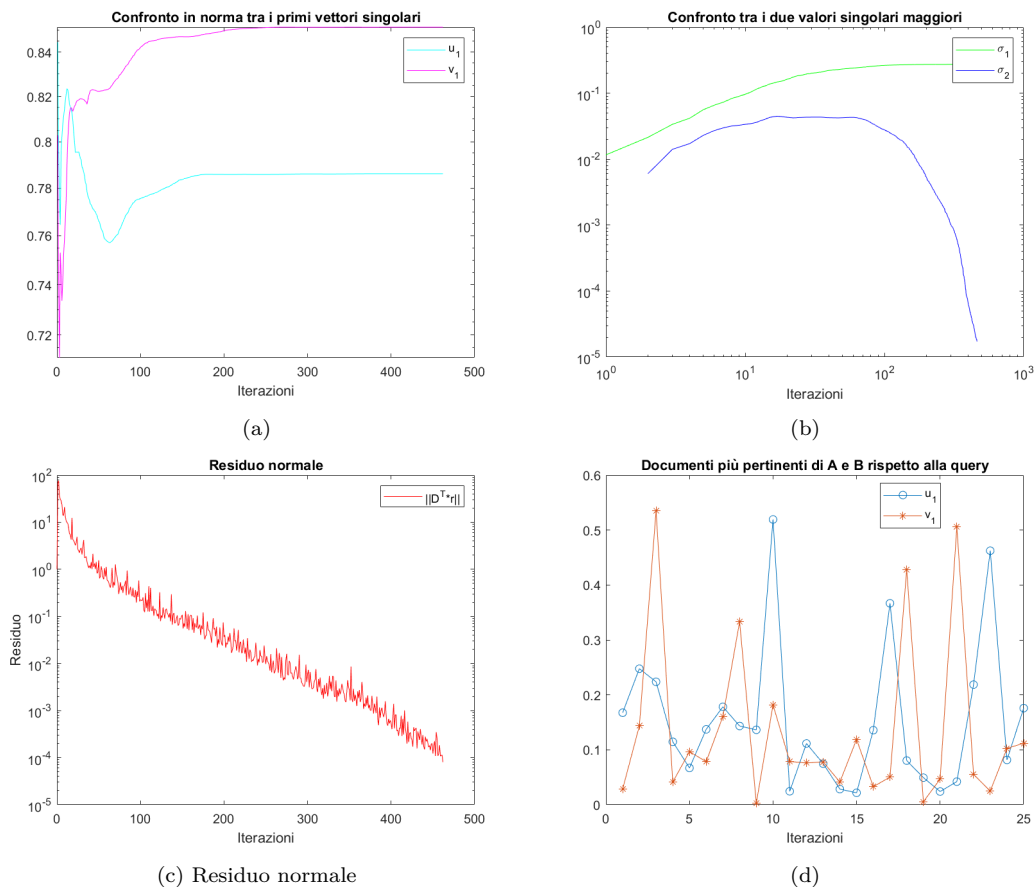


Figura 4.5: $c = g2(m+1, :)$

Rispetto al caso precedente, qui si è in una situazione opposta, dove la query c è stata presa dallo stesso gruppo tematico da cui si è definita la matrice termine-documento B , e come atteso, la matrice X è più proiettata verso B (4.5a). Dalla figura (4.5d) sappiamo che i documenti che hanno reso la matrice B più pertinente alla query espressa sono il 3°, l'8°, il 18° e il 21°, nonostante il fatto che i documenti di A come il 10°, il 17° e il 23° fossero ugualmente molto rappresentativi.

4.3.3 Caso 2

Consideriamo ora il caso in cui la query c non appartenga a nessuno dei due gruppi tematici e per completezza vedremo due risultati, basati sulla scelta arbitraria della query:

- **Caso 2a:** indicato con $\mathbf{g3}=\mathbf{fea}(5769:6089,:)$ il terzo gruppo tematico (dove gli indici sono stati trovati dal vettore $\mathbf{gnd}$) relativo alla label 'crude', cioè 'crude oil'; prendiamo $c = \mathbf{g3}(1,:)$ e calcoliamo la matrice X , ottenendo i seguenti risultati:

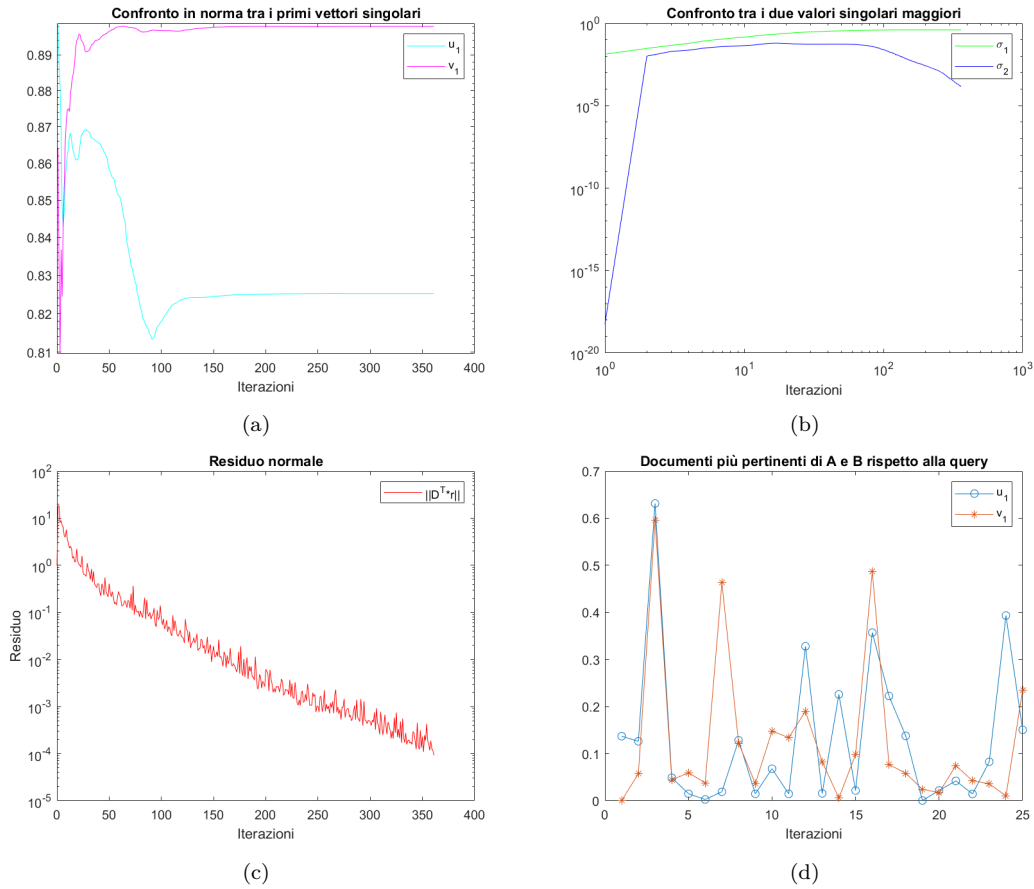


Figura 4.6: $c = \mathbf{g3}(1,:)$

La label 'crude' contiene documenti legati alle notizie su produzione, prezzi ed esportazioni di petrolio e anche accordi commerciali e investimenti tra aziende o stati produttori, quindi tematicamente usano un vocabolario simile a quello della label 'acq'; infatti possiamo vedere dal grafico (4.6a) come la query sia descritta meglio dalla matrice B e come, nonostante il 3° documento di A sia il più pertinente alla query, la combinazione lineare del 3°, 7° e del 16° documento di B orienti la direzione della matrice X (4.6d).

- **Caso 2b:** indicato con $\mathbf{g12}=\mathbf{fea}(7373:7435,:)$ il dodicesimo gruppo tematico relativo alla label 'gnp', cioè 'gross national product'; prendiamo $\mathbf{c} = \mathbf{g12}(1,:)$ e calcoliamo la matrice $\mathbf{X}$, ottenendo i seguenti risultati:

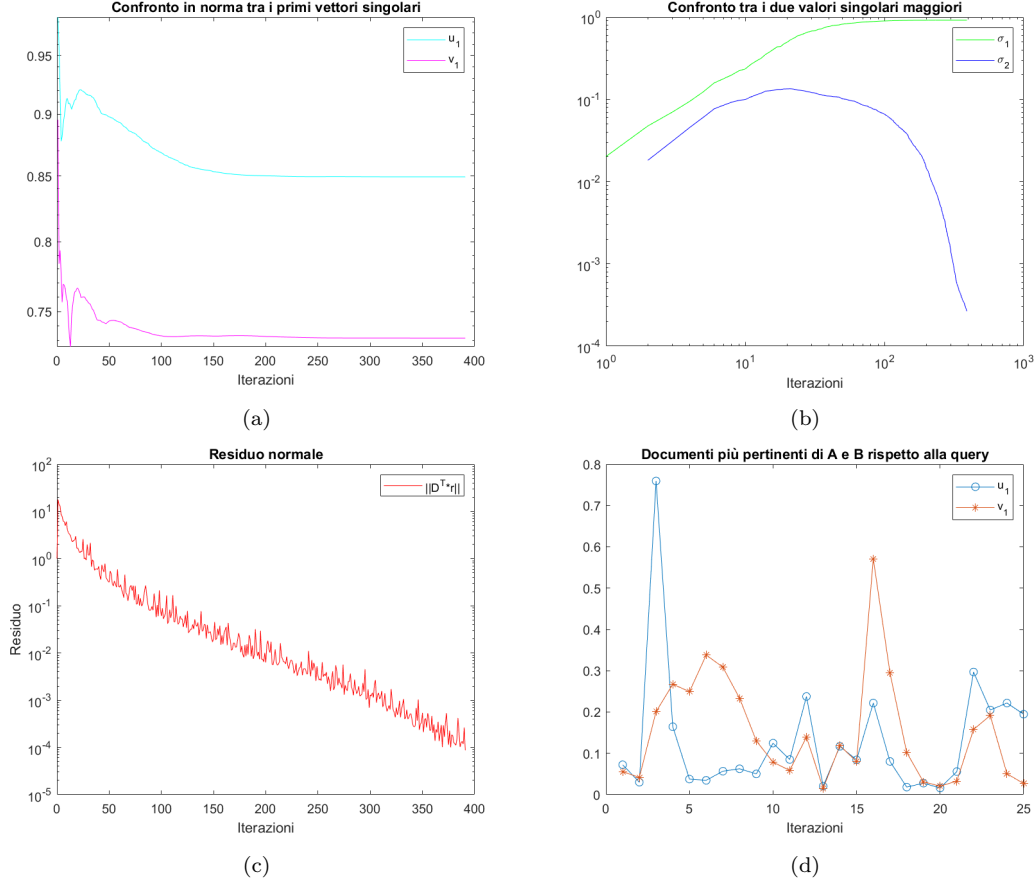


Figura 4.7: $\mathbf{c} = \mathbf{g12}(1,:)$

A livello prettamente tematico il 'gnp', cioè il prodotto nazionale lordo, è un indicatore per valutare l'andamento economico di un paese, quindi si prevede una maggiore affinità al primo gruppo legato agli andamenti economici aziendali; infatti dalla figura (4.7a) si nota questa maggiore affinità, soprattutto con il 3° di A osservando figura (4.7d).

In generale, come si evince dalle casistiche sopra illustrate, il residuo normale decresce rapidamente indipendentemente dalla scelta della query e così anche la convergenza della matrice $\mathbf{X}$, rendendo la trattazione robusta dal punto di vista numerico. L'utilizzo della decomposizione SVD di $\mathbf{X}$, è uno strumento geometrico importante perchè mostra che la direzione principale della matrice si orienta alternativamente verso A o B , a seconda della descrivibilità migliore della query rispetto all'una o all'altra collezione.

Per completezza si riporta di seguito la lista delle etichette associate a ciascun gruppo, i dati utilizzati sono stati scaricati dalla pagina ufficiale di Deng Cai [6] (per un riferimento più dettagliato sui dataset, si rimanda a [7]):

1	earn	23	nat-gas	45	yen
2	acq	24	veg-oil	46	potato
3	crude	25	tin	47	tea
4	trade	26	cotton	48	income
5	money-fx	27	bop	49	instal-debt
6	interest	28	wpi	50	silver
7	ship	29	pet-chem	51	nickel
8	sugar	30	livestock	52	dlr
9	coffee	31	gas	53	platinum
10	gold	32	orange	54	jet
11	money-supply	33	retail	55	cpu
12	gnp	34	strategic-metal	56	groundnut
13	cpi	35	housing	57	inventories
14	cocoa	36	zinc	58	rand
15	alum	37	lumber	59	coconut
16	grain	38	fuel	60	wool
17	copper	39	carcass	61	propane
18	jobs	40	heat	62	l-cattle
19	reserves	41	lei	63	rice
20	rubber	42	oilseed	64	hog
21	iron-steel	43	lead	65	naphtha
22	ipi	44	meal-feed		

Conclusioni

In questo lavoro abbiamo affrontato il problema matriciale ai minimi quadrati nell'ambito dell'Information Retrieval, analizzandolo da un punto di vista sia teorico che pratico. Sono stati messi a punto e studiati due metodi principali per risolvere il problema matriciale: il primo, basato sulla generalizzazione della fattorizzazione QR, risulta particolarmente adatto per matrici di dimensioni contenute; il secondo, che estende l'algoritmo iterativo LSQR al caso matriciale, si è dimostrato efficace per matrici di grandi dimensioni e strutturalmente sparse.

Dal punto di vista teorico, l'analisi ha evidenziato la solidità numerica e la precisione delle soluzioni, grazie alle proprietà intrinseche della decomposizione QR e alla stabilità caratteristica dell'algoritmo LSQR. Un ruolo chiave è stato svolto dall'impiego del prodotto di Kronecker, fondamentale per riformulare il problema in una forma compatibile con metodi noti, facilitando così implementazioni computazionali efficienti.

L'applicazione concreta al dataset Reuters-21578 ha valutato la bontà del modello, mostrando come la matrice soluzione ottenuta sia in grado di rappresentare in maniera efficace la pertinenza delle query rispetto alle diverse raccolte tematiche. L'uso di una decomposizione SVD troncata ha inoltre permesso di interpretare geometricamente le soluzioni, mettendo in luce come il metodo riesca a distinguere tra gruppi tematici associati a differenti matrici termine-documento.

Nonostante ciò, dal punto di vista numerico, entrambi i metodi risultano vincolati dall'assunzione di rango massimo delle matrici coinvolte. Considerando che, nella realtà dei fatti, si opera principalmente con matrici a rango ridotto, le decomposizioni QR e LSQR possono manifestare problemi di stabilità, perdita di ortogonalità e soluzioni non uniche o sensibili al condizionamento. Situazioni di questo tipo richiederebbero l'impiego di pseudo-inverse o la messa in atto di tecniche di regolarizzazione per garantire la robustezza numerica e l'accuratezza anche in presenza di dati ridondanti.

Bibliografia

- [1] Paige, C. C. & Saunders, M. A. (1982). *LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares*, *ACM Transactions on Mathematical Software*, vol. 8, no. 1, pp. 43–71.
- [2] Horn, R. A. & Johnson, C. R. (1991). *Topics in matrix analysis*. Cambridge University Press.
- [3] Eldén, L. (2007). *Matrix Methods in Data Mining and Pattern Recognition (Fundamentals of Algorithms)*. Society for Industrial and Applied Mathematics.
- [4] Debole, F. & Sebastiani, F. (2003). *An analysis of the relative hardness of Reuters-21578 subsets*. Istituto di Scienza e Tecnologie dell’Informazione, Consiglio Nazionale delle Ricerche.
- [5] Thoma, M. (2017, July). *The Reuters Dataset*. Recuperato da: <https://martin-thoma.com/nlp-reuters>
- [6] Deng, C. (s.d.). *Text data collections*. Recuperato da: <http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html>
- [7] Deng, C., Wang, Q. & Han, J. (2011). *Locally consistent concept factorization for document clustering*. *IEEE Transactions on Knowledge and Data Engineering*, 23(6), 902–913.