

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE
Corso di Laurea in Matematica

**METODI NUMERICI
PER IL CALCOLO DEL
PAGERANK**

Tesi di Laurea in Analisi Numerica

Relatore:
Chiar.ma Prof.
Valeria Simoncini

Presentata da:
Laura Girometti

II sessione
Anno Accademico 2018-2019

*A chi mi ha accompagnato e sostenuto,
passo dopo passo, in questo percorso.*

Indice

Introduzione	5
1 Il vettore Pagerank	6
1.1 Introduzione	6
1.2 Strategia di calcolo	6
2 Metodo delle potenze e estrapolazione	10
2.1 Il metodo delle potenze	10
2.2 Estrapolazione	12
2.2.1 Derivazione	12
2.2.2 Analisi della convergenza	13
3 Metodi di proiezione per il calcolo di autovalori	15
3.1 Proiettori	15
3.2 Metodi di proiezione	17
3.2.1 Metodi di proiezione ortogonale	18
3.3 Metodi di proiezione sui sottospazi di Krylov	20
3.3.1 Sottospazi di Krylov	20
3.3.2 Metodo di Arnoldi	22
4 Applicazione del metodo di Arnoldi al calcolo del Pagerank	32
4.1 Algoritmo di Arnoldi 'raffinato'	32
4.2 Esperimenti numerici	35
Riferimenti bibliografici	41

Introduzione

Il Pagerank è un sistema di classificazione che colloca in ordine di importanza quantità enormi di pagine Web. L'importanza di una pagina Web, come sottolineato dagli ideatori dell'algoritmo Larry Page e Serge Brin, è un concetto intrinsecamente soggettivo che dipende dagli interessi dei lettori e che, però, può essere reso in qualche modo oggettivo e calcolato meccanicamente. Questo algoritmo ricopre un ruolo importante nella classificazione delle pagine web svolta dai motori di ricerca ma il suo campo di applicazione non è limitato solo a questo: è applicato ad un qualsiasi grafo o rete presenti in diversi ambiti, come le reti neuronali, i social network, la biologia.

In questa tesi mi occupo di analizzare diversi metodi numerici adatti al calcolo del Pagerank. Nel primo capitolo, ho trattato un'introduzione al problema e ho descritto la sua modellizzazione matematica, sfruttando strumenti matematici che permettono di attribuire un unico valore ad ogni pagina web. La prima tecnica utilizzata da Page è il metodo delle potenze, che, nonostante la sua semplicità, permette di lavorare con una grande mole di dati. Nel secondo capitolo, oltre a trattare di tale metodo, ho analizzato risultati di convergenza accelerata ottenuti tramite una particolare estrapolazione. Successivamente, dopo l'introduzione di metodi più innovativi per il calcolo di autovalori, ho analizzato diverse tecniche basate sui metodi di proiezione ortogonale, in particolare sul metodo di Arnoldi, descritto in dettaglio nel terzo capitolo. Per illustrarne l'efficacia, ho svolto esperimenti numerici per confrontare il comportamento dei metodi descritti e analizzare la sensibilità dei risultati ottenuti al variare del valore del parametro del modello.

Capitolo 1

Il vettore Pagerank

1.1 Introduzione

Definizione 1.1.1 (Grafo). Si dice *grafo* una coppia (V, E) , dove V indica l'insieme dei vertici e E denota l'insieme dei lati; possiamo vedere E anche come una relazione sull'insieme dei vertici V , ovvero come un sottoinsieme di $V \times V$ in cui un arco è identificato dalla coppia (u, v) , dove u è l'estremo iniziale e v l'estremo finale; se la relazione è simmetrica, il grafo si dice indiretto (o non orientato), altrimenti si dice diretto (o orientato).

Possiamo identificare il World Wide Web con un grafo diretto in cui ogni pagina web rappresenta un vertice e in cui l'arco tra la pagina P_i e la pagina P_j è determinato dalla presenza di links (o collegamenti) che, dalla pagina P_i , portano alla pagina P_j .

Definizione 1.1.2 (Matrice di adiacenza del Web). La matrice di adiacenza del grafo Web è la matrice $H = (h_{i,j})$ con

$$h_{i,j} = \begin{cases} \frac{1}{deg(j)} & \text{se } P_j \text{ punta a } P_i \\ 0 & \text{altrimenti} \end{cases}$$

dove $deg(j)$ è il numero di outlinks (collegamenti verso l'esterno) della pagina P_j .

Definizione 1.1.3 (Vettore Pagerank). Il Pagerank è un vettore $I = (w_i)_{i=1\dots n}$ di lunghezza uguale al numero totale di pagine web considerate, la cui i -esima componente rappresenta l'importanza della pagina P_i .

1.2 Strategia di calcolo

Strategia iniziale: Sia w_i l'importanza della i -esima pagina web; definiamo

$$w_i = \sum_{j \in B_i} \frac{w_j}{deg(j)}, \quad (1.1)$$

dove B_i rappresenta la famiglia degli indici degli outlinks di P_i .

Il modello proposto da Page per il calcolo del Pagerank si basa sull'ipotesi che w_i , cioè l'importanza della pagina P_i , dipenda dall'importanza delle pagine che puntano alla pagina stessa.

Sotto questa ipotesi, il calcolo del vettore I viene eseguito iterativamente: supponiamo inizialmente che tutte le pagine siano classificate allo stesso modo, ovvero $w_i^{(0)} = \frac{1}{n}$ e applichiamo l'iterazione

$$w_i^{(k+1)} = \sum_{j \in B_i} \frac{w_j^{(k)}}{\deg(j)}. \quad (1.2)$$

In termini matriciali, (1.1) equivale a richiedere che $I = HI$, cioè che I sia autovettore di H associato all'autovalore 1 mentre l'iterazione descritta in (1.2) equivale ad applicare il metodo delle potenze alla matrice H , cioè alla matrice di adiacenza del web, scegliendo come vettore iniziale $\frac{1}{n}\mathbf{1}$.

Il passo successivo rispetto alla strategia appena descritta consiste nel modificare la matrice a cui il metodo viene applicato per risolvere alcune problematiche:

- Non è detto che H abbia autovalore 1;
- H potrebbe avere colonne nulle nel caso in cui ci siano delle pagine prive di outlinks (nodi non connessi);
- L'autovettore associato all'autovalore 1 potrebbe non essere unico nel caso in cui l'autospazio associato abbia dimensione maggiore (ovvero nel caso siano presenti, ad esempio, dei percorsi ciclici);

Innanzitutto modifichiamo H affinché sia una matrice stocastica per colonne:

Definizione 1.2.1. Sia $A \in \mathbb{R}^{n \times n}$; diciamo che A è una matrice stocastica per colonne se $a_{i,j} \geq 0 \forall i, j = 1 \dots n$ e $\sum_{i=1}^n a_{i,j} = 1 \forall j = 1 \dots n$.

Osserviamo che gli autovalori di una matrice stocastica per colonne soddisfano la relazione $1 = \lambda_1 \geq |\lambda_2| \geq \dots \geq |\lambda_n|$.

Sostituiamo quindi ogni colonna nulla con una colonna costante, che indica uguale probabilità di passare ad una qualunque altra pagina, se ci troviamo in un nodo non connesso. Definiamo quindi il vettore

$$d_j = \begin{cases} 1 & \text{se } \deg(j) = 0 \\ 0 & \text{altrimenti} \end{cases}$$

e la matrice

$$\tilde{H} = H + \frac{1}{n}\mathbf{1}d^T.$$

Possiamo anche sostituire il vettore $\frac{1}{n}\mathbf{1}$ con un vettore w tale che $w_i \geq 0 \forall i = 1, \dots, n$ e $\|w\|_1 = 1$, cioè modifichiamo la distribuzione di probabilità di passare da un nodo non connesso ad una qualsiasi altra pagina. Osserviamo che vale

$$\tilde{H}^T \mathbf{1} = H^T \mathbf{1} + \frac{1}{n} d \mathbf{1}^T \mathbf{1} = \mathbf{1}.$$

Questo implica che \tilde{H} è una matrice stocastica per colonne e che $\mathbf{1}$ è autovettore destro di \tilde{H}^T associato all'autovalore 1; in particolare, $\mathbf{1}$ è autovettore sinistro per \tilde{H} sempre relativo all'autovalore 1. Vale anche la relazione

$$\tilde{H} I = I,$$

che equivale a dire che il vettore di Pagerank è proprio l'autovettore destro associato. Possiamo vedere il vettore I anche come una distribuzione stazionaria di probabilità la cui j -esima componente corrisponde alla probabilità che il 'walker' o navigatore si trovi nella pagina P_j a lungo termine.

Definizione 1.2.2 (Matrice riducibile). Sia $A \in \mathbb{R}^{n \times n}$; diciamo che A è riducibile se esiste una matrice di permutazione T tale che

$$T A T^T = \begin{bmatrix} X & Y \\ 0 & Z \end{bmatrix},$$

con X e Z quadrate. Altrimenti diciamo che A è irriducibile.

Proposizione 1.2.1. *Una matrice A è irriducibile se e solo se il suo grafo diretto è fortemente connesso (ovvero $\forall a_{i,j} \neq 0$ esiste un percorso nel grafo che porta da P_i a P_j).*

Per assicurare l'unicità del vettore di Pagerank, modifichiamo la matrice \tilde{H} affinché sia irriducibile e soddisfi le ipotesi del seguente teorema:

Teorema 1.2.1. *Sia A irriducibile e stocastica per colonne; Allora l'autovalore dominante è 1, ha molteplicità 1 e l'autovettore associato ha tutte componenti positive. Inoltre, se A ha elementi tutti positivi, si ha che $|\lambda_i| < 1 \forall i > 2$.*

Definiamo quindi la matrice di Google

$$P = \alpha \tilde{H} + (1 - \alpha) \frac{1}{n} \mathbf{1} \mathbf{1}^T, \text{ con } \alpha \in [0, 1].$$

Questa modifica di rango uno della matrice assicura l'irriducibilità di P : abbiamo aggiunto un collegamento 'fittizio' in modo che il navigatore possa passare da una pagina all'altra in modo casuale con probabilità $1 - \alpha$ (non considerando gli outlinks disponibili). Possiamo anche scegliere un generico vettore di probabilità v al posto di $\frac{1}{n} \mathbf{1}$, in modo da influenzare dall'"esterno" la scelta dell'importanza delle pagine web (v è chiamato vettore di personalizzazione).

Teorema 1.2.2. *Sia A una matrice stocastica per colonne il cui spettro è $\{1, \lambda_2, \dots, \lambda_n\}$; Allora lo spettro della matrice $P = \alpha A + (1 - \alpha)\frac{1}{n}\mathbf{1}\mathbf{1}^T$ è $\{1, \alpha\lambda_2, \dots, \alpha\lambda_n\}$.*

Dimostrazione. Sia $\bar{\mathbf{1}} = \frac{1}{\sqrt{n}}\mathbf{1}$; osserviamo che $P^T\bar{\mathbf{1}} = \alpha\tilde{H}\bar{\mathbf{1}} + (1 - \alpha)\frac{1}{n}\mathbf{1}\mathbf{1}^T\bar{\mathbf{1}} = \bar{\mathbf{1}}$. Sia ora $U = [\bar{\mathbf{1}}, U_0]$ unitaria; allora $U^*AU = \begin{bmatrix} 1 & * \\ 0 & T \end{bmatrix}$.

Di conseguenza, $U^*PU = \alpha \begin{bmatrix} 1 & * \\ 0 & T \end{bmatrix} + (1 - \alpha)(U^*\bar{\mathbf{1}})(\bar{\mathbf{1}}^TU) = \alpha \begin{bmatrix} 1 & * \\ 0 & T \end{bmatrix} + (1 - \alpha)e_1e_1^T = \begin{bmatrix} 1 & * \\ 0 & \alpha T \end{bmatrix}$. □

Questo teorema dimostra quindi che l'autovalore 1 associato alla matrice P è semplice, a prescindere da quale sia la molteplicità algebrica dello stesso autovalore associato alla matrice A .

Possiamo anche dire qualcosa in più riguardo il secondo autovalore della matrice P sopra considerata:

Teorema 1.2.3. *Sia $A \in \mathbb{R}^{n \times n}$ una matrice stocastica per colonne; se A ha almeno 2 sottoinsiemi chiusi irriducibili (che è il caso della matrice \tilde{H} costruita in precedenza), allora il secondo autovalore più grande in modulo della matrice $P = \alpha A + (1 - \alpha)\frac{1}{n}\mathbf{1}\mathbf{1}^T$ è $\lambda_2 = \alpha$.*

Dal punto di vista teorico, la strategia appena analizzata per il calcolo del vettore di Pagerank consiste nel risolvere un problema agli autovalori. Nei capitoli successivi analizzeremo alcuni tra i metodi numerici più adatti per risolvere questo problema, viste le grandi dimensioni della matrice che stiamo considerando.

Capitolo 2

Metodo delle potenze e estrapolazione

2.1 Il metodo delle potenze

Per quanto visto in precedenza, l'autocoppia da calcolare per determinare il vettore di Pagerank è l'autocoppia dominante della matrice di adiacenza e il metodo delle potenze si presta bene a svolgere questo calcolo, visto che nell'algoritmo vengono eseguiti solamente prodotti matrice per vettore, implementati in modo efficiente. In generale infatti, la matrice di adiacenza non è sparsa ed è impensabile utilizzare metodi di fattorizzazione troppo costosi sia in termini di memoria che di costo computazionale.

Il metodo delle potenze è un metodo iterativo che permette di approssimare l'autocoppia dominante di una matrice, ovvero l'autovalore più grande in modulo e l'autovettore associato.

L'idea di base dell'iterazione è molto semplice : si vuole sfruttare il fatto che la direzione del vettore $A^k x^{(0)}$, dove $x^{(0)}$ è il vettore iniziale scelto, tenda alla direzione dell'autovettore dominante da approssimare, sotto opportune ipotesi. Si richiede, infatti, che la matrice A sia diagonalizzabile, che la componente di $x^{(0)}$ rispetto all'autovettore cercato sia non nulla e che l'autovalore dominante λ_1 sia semplice. Sia $B = \{v_1, \dots, v_n\}$ una base di autovettori e siano $\alpha_1, \dots, \alpha_n$ le componenti di $x^{(0)}$ rispetto alla base B .

Si ha che

$$x^{(k)} = A^k x^{(0)} = A^k \sum_{i=1}^n \alpha_i v_i = \sum_{i=1}^n \alpha_i \lambda_i^k v_i.$$

Supponendo che $\alpha_1 \neq 0$, otteniamo

$$x^{(k)} = \alpha_1 \lambda_1^k \left(v_1 + \sum_{i=2}^n \frac{\alpha_i}{\alpha_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k v_i \right). \quad (2.1)$$

Poichè λ_1 è l'autovalore dominante ed è distinto in modulo dai restanti, si ha che $x^{(k)}$ tende ad un multiplo del vettore v_1 . La norma del vettore $x^{(k)}$ dipende dal modulo di

λ_1 : se $|\lambda_1| < 1$ allora $\|x^{(k)}\| \rightarrow 0$ per $k \rightarrow \infty$; se $|\lambda_1| > 1$ allora $\|x^{(k)}\| \rightarrow \infty$ per $k \rightarrow \infty$; per fare in modo che la norma del k -esimo iterato non cresca o decresca troppo, normalizziamo $x^{(k)}$ ad ogni iterazione, ottenendo il seguente algoritmo:

Algorithm 1 Metodo delle potenze

```

for  $k = 0, 1, 2, \dots$  do
   $\lambda^{(k)} = (x^{(k)})^H y$ ;
  if c'è convergenza then
    return
  end if
   $x^{(k+1)} = \frac{y}{\|y\|}$ ;
   $y = Ax^{(k+1)}$ ;
end for

```

Nell'algoritmo descritto, l'autovalore cercato viene approssimato tramite il quoziente di Rayleigh:

$$\lambda^{(k)} = \frac{(x^{(k)})^H Ax^{(k)}}{(x^{(k)})^H x^{(k)}}.$$

Osserviamo che l'implementazione è efficiente: viene infatti calcolato un solo prodotto matrice per vettore per iterazione!

Nel caso del Pagerank, il nostro interesse si concentra sull'autovettore perciò non è necessario calcolare il quoziente di Rayleigh per approssimare l'autovalore; Inoltre, si è notato che è conveniente normalizzare $x^{(k)}$ rispetto alla norma 1 per ragioni di aritmetica finita. Analizziamo ora la velocità di convergenza:

Teorema 2.1.1. *Sia $A \in \mathbb{R}^{n \times n}$ e siano $\lambda_1, \lambda_2, \dots, \lambda_n$ i suoi autovalori ordinati in modo che $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$, dove λ_1 è semplice; sia $x^{(0)} \in \mathbb{R}^n$ e sia α_1 la componente di $x^{(0)}$ rispetto all'autovettore v_1 associato all'autovalore λ_1 ; Se $\alpha_1 \neq 0$, allora esiste una costante $C > 0$ tale che*

$$\|x^{(k)} - v_1\|_2 \leq C \left| \frac{\lambda_2}{\lambda_1} \right|^k, \text{ con } k \geq 1.$$

Questo teorema dà una stima della velocità di convergenza della successione $\{x^{(k)}\}$ all'autovettore v_1 e mostra che tale convergenza è lineare rispetto al rapporto $\left| \frac{\lambda_2}{\lambda_1} \right|$.

Stimando il rapporto $\left| \frac{\lambda_2}{\lambda_1} \right|$ per la matrice di Google, possiamo, quindi, avere un'idea della velocità di convergenza del metodo delle potenze applicato al calcolo del Pagerank:

Dal teorema 1.2.3 abbiamo che :

$$\left| \frac{\lambda_2}{\lambda_1} \right| = \alpha \text{ con } \alpha \in [0, 1].$$

La scelta di quale sia il valore di α più appropriato diventa perciò cruciale perchè da esso dipende sia la velocità di convergenza dell'algoritmo che l'importanza data al grafo Web non modificato: al crescere di α , aumenta infatti il valore dato alla matrice del Web originaria mentre decresce la velocità di convergenza del metodo.

Perciò il passo successivo da compiere è escogitare una modifica dell'algoritmo che permetta di accelerare il metodo per valori di α vicini a 1.

2.2 Estrapolazione

Una delle principali tecniche utilizzate come acceleratore del metodo delle potenze è l'estrapolazione. Fissato un certo valore m , essa consiste nel modificare il vettore $x^{(m)}$ ottenuto alla m -esima iterazione dell'algoritmo, con una combinazione lineare di vettori calcolati nelle iterazioni precedenti. La scelta della combinazione lineare distingue i diversi metodi: ad esempio, sia nel caso dell'estrapolazione quadratica che in un nuovo tipo di estrapolazione proposta da Xueyuan Tan, l'idea alla base è quella di determinare un'approssimazione dell'autovettore sfruttando le proprietà del polinomio caratteristico della matrice e la struttura delle iterate ottenute dal metodo delle potenze. Ciò in cui differiscono essenzialmente i due metodi è il modo in cui approssimano il polinomio caratteristico.

In questa sezione analizziamo il nuovo metodo estrapolativo proposto da Xueyuan Tan in [2].

2.2.1 Derivazione

Sapendo che la matrice di Google ha come autovalore 1, possiamo scrivere il suo polinomio caratteristico come $(\lambda - 1)Q_P(\lambda)$, dove $Q_P(\lambda)$ è un polinomio di grado $n - 1$.

Dal teorema di Hamilton-Cayley, otteniamo che

$$(P - I)Q_P(P) = 0. \tag{2.2}$$

Inoltre, esiste almeno un vettore $x^{(0)}$ tale che $Q_P(P)x^{(0)} \neq 0$. Da (2.2), deduciamo che $Q_P(P)x^{(0)}$ è autovettore di P relativo a 1, ovvero parallelo al vettore di Pagerank. Le grandi dimensioni di P non permettono di calcolare I in questo modo e rendono necessaria una buona approssimazione del polinomio.

Se indichiamo con ℓ il numero di nodi non connessi, abbiamo che la matrice P avrà

almeno ℓ colonne identiche, ovvero autovalore 0 con molteplicità algebrica almeno $\ell - 1$.
Ne segue che

$$Q_P(\lambda) = \lambda^{n-1} - (\mu - 1)\lambda^{n-2} + \alpha_3\lambda^{n-3} + \dots + \alpha_p\lambda^{n-p},$$

dove μ è la traccia di P e $p \leq n - l + 1$.

Possiamo approssimare il polinomio come $\lambda^{n-1} - (\mu - 1)\lambda^{n-2}$ e utilizzare $[P^{n-1} - (\mu - 1)P^{n-2}]x^{(0)}$ come approssimazione del Pagerank. Osserviamo che

$$\begin{aligned} [P^{n-1} - (\mu - 1)P^{n-2}]x^{(0)} &= P^{n-2}[P - (\mu - 1)I]x^{(0)} \\ &= P^{n-m-1}[P - (\mu - 1)I]P^{m-1}x^{(0)} \\ &= P^{n-m-1}[P - (\mu - 1)I]x^{(m-1)} \\ &= P^{n-m-1}[x^{(m)} - (\mu - 1)x^{(m-1)}], \quad (2.3) \end{aligned}$$

con $m < n - 1$.

Il metodo estrapolativo basato sulla traccia della matrice viene definito in questo modo: dopo m passi del metodo delle potenze, approssimiamo il Pagerank I con una combinazione lineare di $x^{(m)}$ e $x^{(m-1)}$ e lo utilizziamo come vettore iniziale per una nuova iterazione del metodo delle potenze. Intuitivamente, (2.3) dice che stiamo scegliendo un nuovo vettore iniziale che porta l'iterazione nella direzione giusta. Notiamo che il calcolo della traccia non è costoso: dalla costruzione di P , è facile vedere che $\mu = 1 + \alpha(\frac{l}{n} - 1)$ e risulta $\mu < 1$. L'introduzione dell'estrapolazione non modifica significativamente il costo computazionale ma accelera la convergenza e, rispetto all'estrapolazione quadratica, ha un costo inferiore sia in termini di memoria che di costo computazionale.

2.2.2 Analisi della convergenza

Analizziamo più approfonditamente la convergenza del metodo sfruttando il seguente teorema:

Teorema 2.2.2.1 ([3]). *Sia $B \in \mathbb{C}^{n \times n}$ e sia p un polinomio di grado $m < n$; Siano $\lambda_1, \dots, \lambda_n$ gli autovalori di B tali che $|p(\lambda_1)| \geq |p(\lambda_2)| \dots \geq |p(\lambda_n)|$. Supponiamo che esista k tale che $|p(\lambda_k)| > |p(\lambda_{k+1})|$ e sia $\rho = |p(\lambda_{k+1})| / |p(\lambda_k)|$. Siano \mathcal{U} e \mathcal{V} gli autospazi di B associati rispettivamente a $\lambda_1, \dots, \lambda_k$ e a $\lambda_{k+1}, \dots, \lambda_n$. Consideriamo l'iterazione stazionaria del sottospazio $\mathcal{S}_i = p(B)\mathcal{S}_{i-1}$, $\mathcal{S}_0 = \mathcal{S}$ con \mathcal{S} sottospazio di dimensione k tale che $\mathcal{S} \cap \mathcal{V} = \{0\}$. Allora $\forall \hat{\rho}$ tale che $\rho < \hat{\rho} < 1$ esiste una costante C tale che*

$$d(\mathcal{S}_i, \mathcal{U}) \leq C\hat{\rho}^i$$

Abbiamo quindi convergenza lineare al sottospazio dominante e invariante rispetto ad A ma dobbiamo osservare che la velocità di convergenza dipende da $\hat{\rho}$, che possiamo scegliere più vicino possibile a ρ , però più $\hat{\rho}$ è vicino a ρ maggiore sarà la costante C , la cui esistenza è assicurata dal teorema.

Dal punto di vista dell' iterazione del sottospazio, il nuovo algoritmo genera una sequenza di sottospazi \mathcal{S}_i :

$$\mathcal{S}_i = p(A)\mathcal{S}_{i-1} \text{ con } \mathcal{S}_0 = \text{span}(x^{(0)}) \text{ e } p(A) = A^{m-1} [A - (\mu - 1)I]$$

Possiamo quindi applicare il teorema alla matrice di Google : abbiamo che $\frac{p(\alpha)}{p(1)}$ è < 1 , scegliendo come polinomio di grado $m < n$ il polinomio $p(\lambda) = \lambda^m - (\mu - 1)\lambda^{m-1}$. Infatti $p(1) = 2 - \mu$ e $p(\alpha) = \alpha^m - (\mu - 1)\alpha^{m-1} \Rightarrow$

$$\left| \frac{p(\alpha)}{p(1)} \right| = \left| \frac{\alpha^{m-1} [\alpha - (\mu - 1)]}{2 - \mu} \right| < \alpha^{m-1} < 1.$$

In particolare, in questo caso \mathcal{U} è l'autospazio associato a 1 e \mathcal{V} è l'autospazio associato a $\lambda_2, \dots, \lambda_n$.

Il teorema assicura la convergenza lineare dei sottospazi costruiti dall'algoritmo all'autospazio $\text{span}(I)$, con una velocità di convergenza accelerata, dipendente da $\hat{\rho} \in (\rho, \alpha^{m-1})$

Capitolo 3

Metodi di proiezione per il calcolo di autovalori

Analizziamo ora un'altra classe di metodi per il calcolo degli autovalori che sono risultati essere utili nel calcolo del vettore di Pagerank.

3.1 Proiettori

Definizione 3.1.1. Un proiettore P è una trasformazione lineare da \mathbb{C}^n a \mathbb{C}^n che è idempotente, cioè tale che $P^2 = P$.

Proprietà:

- Se P è un proiettore, allora anche $I - P$ è un proiettore e vale che $Ker(P) = Range(I - P)$;
- $Ker(P) \cap Range(P) = \{0\}$;
Infatti, se $x \in Range(P)$ allora $Px = x$; se $x \in Ker(P)$ allora $Px = 0$; Di conseguenza l'intersezione è costituita soltanto dal vettore nullo;
- Si ha, inoltre, che $\mathbb{C}^n = Ker(P) \oplus Range(P)$; infatti ogni elemento $x \in \mathbb{C}^n$ si può scrivere come $x = Px + (I - P)x$ e, essendo l'intersezione nulla, vale che la somma è diretta;
- Viceversa, ogni coppia di sottospazi M e S , che formano una somma diretta di \mathbb{C}^n , definiscono un unico proiettore P tale che $Range(P) = M$ e $Ker(P) = S$; ogni elemento x può essere decomposto in modo unico come $x = x_1 + x_2$, con $x_1 \in M$ e $x_2 \in S$; perciò la trasformazione lineare associata al proiettore mappa ogni elemento x in x_1 , ovvero nella componente rispetto al sottospazio M .

Definizione 3.1.2. Siano M e S due sottospazi che formano una somma diretta di \mathbb{C}^n ; se S è il complemento ortogonale di M , diciamo che il proiettore P associato è *ortogonale* su M , ovvero $\text{Ker}(P) = \text{Range}(P)^\perp$ e

$$Px \in M \text{ e } (I - P)x \perp M.$$

Proposizione 3.1.3. *Un proiettore P è ortogonale \Leftrightarrow è hermitiano.*

Dimostrazione. (\Leftarrow) Sia P hermitiano; allora vale che

$$\langle P^H x, y \rangle = \langle x, Py \rangle \quad \forall x, \forall y.$$

Da qui deduciamo che

$$\text{Ker}(P^H) = \text{Range}(P)^\perp \tag{3.1}$$

$$\text{Ker}(P) = \text{Range}(P^H)^\perp. \tag{3.2}$$

Vale quindi che $\text{Ker}(P) = \text{Range}(P)^\perp$ e possiamo concludere che P è ortogonale.

(\Rightarrow) Supponiamo P proiettore ortogonale; allora P^H è un proiettore ortogonale. Vale infatti che $(P^H)^2 = (P^2)^H = P^H$. Da (3.1) deduciamo che $\text{Ker}(P) = \text{Ker}(P^H)$ mentre da (3.2) deduciamo che $\text{Range}(P) = \text{Range}(P^H)$. Poichè un proiettore P è identificato univocamente da $\text{Ker}(P)$ e $\text{Range}(P)$, abbiamo che $P = P^H$. \square

Definizione 3.1.4 (Rappresentazione matriciale di un proiettore ortogonale). Sia P un proiettore ortogonale; Sia V una matrice avente colonne ortogonali che formano una base per $M = \text{Range}(P)$; allora possiamo rappresentare il proiettore P tramite la matrice VV^H , che è idempotente e soddisfa la caratterizzazione precedente.

Osserviamo che la rappresentazione matriciale non è unica: ogni base ortonormale del sottospazio M dà una rappresentazione del proiettore diversa ma, per ogni V_1 e V_2 basi ortonormali di $\text{Range}(P)$, abbiamo che $V_1 V_1^H = V_2 V_2^H$.

Proposizione 3.1.1. *Sia P un proiettore ortogonale su un sottospazio S ; allora per ogni $x \in \mathbb{C}^n$ si ha che*

$$\min_{y \in S} \|x - y\|_2 = \|x - Px\|_2.$$

.

Dimostrazione. Consideriamo un vettore $y \in S$; si ha che

$$\|x - y\|_2 = \|x - Px + (Px - y)\|_2 = \|x - Px\|_2 + \|Px - y\|_2,$$

dato che il vettore $x - Px$ è ortogonale al sottospazio M , a cui appartiene il vettore $Px - y$. Da qui abbiamo che $\|x - y\|_2 \geq \|x - Px\|_2$ e per $y = Px$ raggiungiamo il minimo richiesto. \square

Definizione 3.1.5. Un proiettore P che non è ortogonale è detto *obliquo*. Sia $S = Ker(P)$ e sia $M = Range(P)$; Consideriamo L il complemento ortogonale di S , che ha perciò la stessa dimensione di M ; Osserviamo che dire che $(I - P)x \in Ker(P)$ equivale a dire che appartiene al complemento ortogonale di L . Caratterizziamo il proiettore P in questo modo:

$$Px \in M \text{ e } (I - P)x \perp L.$$

Diciamo che P è un proiettore su M e ortogonale a L .

Definizione 3.1.6 (Rappresentazione matriciale di un proiettore obliquo). Per la rappresentazione matriciale di un proiettore obliquo abbiamo bisogno di due basi: consideriamo una matrice V , le cui colonne formano una base per $M = Range(P)$, e una matrice W , le cui colonne formano una base per L , il complemento ortogonale di S ; diremo che le due basi sono biortogonali se

$$\langle v_i, w_j \rangle = \delta_{ij}.$$

Quindi, data una coppia di basi biortogonali V, W , possiamo rappresentare il proiettore P come $P = VW^H$.

3.2 Metodi di proiezione

La maggior parte degli algoritmi per il calcolo degli autovalori contiene al suo interno un metodo di proiezione. Esso può essere sia il corpo centrale dell'algoritmo che una tecnica aggiunta per migliorarne l'efficienza: ad esempio, nel caso in cui la matrice sia reale mentre l'autovettore dominante sia complesso, una tecnica di proiezione sullo spazio generato dagli iterati x_{j+1}, x_j , calcolati tramite il metodo delle potenze, permette di ottenere una buona approssimazione dell'autocoppia.

Un metodo di proiezione consiste nell'approssimare l'esatto autovettore u tramite un vettore \tilde{u} , che appartiene ad un certo sottospazio K , chiamato sottospazio di approssimazione o sottospazio destro, imponendo che il vettore residuo di \tilde{u} sia ortogonale ad un qualche sottospazio L , chiamato sottospazio sinistro. Ci sono due grandi classi di metodi di proiezione: i metodi di proiezione ortogonali, in cui si sceglie $K = L$, e i metodi di proiezione obliqui, in cui K e L sono distinti e possono essere anche non correlati tra loro.

Si pone ora il problema della bontà dell'approssimazione: è chiaro che, se nessun vettore di K è sufficientemente vicino a u , nessun metodo di proiezione sarà efficiente. Però nel caso in cui esista un vettore $\tilde{u} \in K$, che dista ϵ dal vettore esatto u , vogliamo capire con quale accuratezza il metodo è in grado di calcolarlo.

3.2.1 Metodi di proiezione ortogonale

Siano $A \in \mathbb{C}^{n \times n}$ e K un sottospazio di \mathbb{C}^n di dimensione m ; consideriamo il problema agli autovalori associato ad A consistente nel trovare $u \in \mathbb{C}^n$ e $\lambda \in \mathbb{C}$ tale che

$$Au = \lambda u. \quad (3.3)$$

Un metodo di proiezione ortogonale sul sottospazio K consiste nel cercare un'autocoppia approssimata $(\tilde{\lambda}, \tilde{u})$, con $\tilde{u} \in K$ e $\tilde{\lambda} \in \mathbb{C}$ che risolva il problema (3.3) imponendo che

$$A\tilde{u} - \tilde{\lambda}\tilde{u} \perp K \quad (3.4)$$

o, equivalentemente,

$$\langle A\tilde{u} - \tilde{\lambda}\tilde{u}, v \rangle = 0 \quad \forall v \in K. \quad (3.5)$$

Ora, se è disponibile una base ortonormale $\{v_1, \dots, v_m\}$ del sottospazio K , definiamo la matrice $V = [v_1 \ \dots \ v_m]$ e risolviamo il problema traducendolo in questa base: sia

$$\tilde{u} = Vy.$$

Possiamo riscrivere l'equazione (3.5) :

$$\langle AVy - \tilde{\lambda}Vy, v_j \rangle = 0 \quad \forall j = 1, \dots, m. \quad (3.6)$$

Otteniamo che y , $\tilde{\lambda}$ devono soddisfare:

$$B_m y = \tilde{\lambda} y \quad \text{con } B_m = V^H AV.$$

Infatti, (3.6) equivale a richiedere

$$V^H(AVy - \tilde{\lambda}Vy) = 0 \Leftrightarrow V^H AVy = \tilde{\lambda}y.$$

Osserviamo che, definendo la trasformazione lineare $A_m = P_k A P_k$, dove P_k è un proiettore ortogonale sullo spazio K , la matrice B_m rappresenta la restrizione dell'operatore A_m al sottospazio K rispetto alla base V scelta. Vediamo meglio questo fatto analizzando il problema dal punto di vista dei proiettori ortogonali. Dato P_k proiettore ortogonale sullo spazio K , esprimiamo la condizione (3.4) in questi termini:

$$P_k(A\tilde{u} - \tilde{\lambda}\tilde{u}) = 0$$

o,

$$P_k A \tilde{u} = \tilde{\lambda} \tilde{u} \quad \text{con } \tilde{\lambda} \in \mathbb{C} \text{ e } \tilde{u} \in K.$$

Come abbiamo visto prima, abbiamo trasformato il problema (3.3) in un problema agli autovalori della trasformazione lineare $P_k A$ ristretta al sottospazio K . Possiamo riformulare l'equazione estendendo la trasformazione a tutto lo spazio:

$$P_k A P_k \tilde{u} = \tilde{\lambda} \tilde{u} \quad \text{con } \tilde{\lambda} \in \mathbb{C} \text{ e } \tilde{u} \in \mathbb{C}^n.$$

Oltre agli autovalori di B_m , A_m ha anche autovalore 0 con molteplicità multipla e gli autovettori associati sono i vettori che appartengono al complemento ortogonale di K . Analizziamo cosa succede quando il sottospazio K è invariante sotto l'azione di A .

Proposizione 3.2.1.1. *Se il sottospazio K è invariante sotto A , ogni autocoppia approssimata, calcolata tramite il metodo di proiezione ortogonale, è esatta.*

Dimostrazione. Sia $\tilde{\lambda} \in \mathbb{C}, \tilde{u} \in \mathbb{C}^n$ un'autocoppia approssimata; allora vale che

$$P_k(A\tilde{u} - \tilde{\lambda}\tilde{u}) = 0.$$

Poichè K è invariante sotto l'azione di A , otteniamo che $P_k A\tilde{u} = A\tilde{u}$. Questo dimostra che l'autocoppia $(\tilde{\lambda}, \tilde{u})$ è esatta. \square

Un'importante quantità, che gioca un ruolo cruciale nell'analisi della convergenza dei metodi di proiezione, è la distanza tra l'autovettore esatto u e il sottospazio K , ovvero $\|(I - P_k)u\|_2$. È chiaro che, se questa quantità non è piccola, l'autovettore non può essere ben approssimato, dato che $\|\tilde{u} - u\|_2 \geq \|(I - P_k)u\|_2$. Possiamo interpretare $\|(I - P_k)u\|_2$ anche come il seno dell'angolo acuto compreso tra il vettore u e il sottospazio K .

Stabiliamo ora una stima del residuo dell'autocoppia esatta rispetto alla trasformazione lineare approssimata A_m , ovvero quanto è 'distante' la coppia esatta (λ, u) dall'essere autocoppia della matrice A_m . Questa stima è importante perchè, avendo trasformato il problema agli autovalori di A in un problema agli autovalori di A_m , ci permette di capire quanto l'autocoppia approssimata sia lontana dall'autocoppia cercata, supponendo i problemi considerati ben condizionati.

Teorema 3.2.1.1. *Sia $\gamma = \|P_k A(I - P_k)\|_2$. Allora le norme dei residui delle coppie $\lambda, P_k u$ e λ, u per l'operatore lineare A_m soddisfano rispettivamente le seguenti disuguaglianze:*

$$\|(A_m - \lambda I)P_k u\|_2 \leq \gamma \|(I - P_k)u\|_2 \quad (3.7)$$

$$\|(A_m - \lambda I)u\|_2 \leq \sqrt{\lambda^2 + \gamma^2} \|(I - P_k)u\|_2. \quad (3.8)$$

Dimostrazione. Per la prima disuguaglianza, utilizziamo la definizione di A_m e otteniamo:

$$\begin{aligned} \|(A_m - \lambda I)P_k u\|_2 &= \|P_k(A - \lambda I)(u - (I - P_k)u)\|_2 \\ &= \|P_k(A - \lambda I)(I - P_k)u\|_2 \\ &= \|P_k(A - \lambda I)(I - P_k)(I - P_k)u\|_2 \\ &\leq \gamma \|(I - P_k)u\|_2. \end{aligned}$$

Per la seconda disuguaglianza, scomponiamo il vettore u sul sottospazio K e sul suo ortogonale:

$$(A_m - \lambda I)u = (A_m - \lambda I)P_k u + (A_m - \lambda I)(I - P_k)u = (A_m - \lambda I)P_k u - \lambda(I - P_k)u.$$

Dato che i due vettori della scomposizione sono ortogonali, abbiamo che:

$$\begin{aligned} \|(A_m - \lambda I)u\|_2^2 &= \|(A_m - \lambda I)P_k u\|_2^2 + \|\lambda(I - P_k)u\|_2^2 \\ &\leq (\gamma^2 + |\lambda|^2) \|(I - P_k)u\|_2^2. \end{aligned}$$

□

Osserviamo che γ è limitata superiormente da $\|A\|_2$. Se $\|(I - P_k)u\|_2$ è piccolo, il metodo di proiezione ortogonale può raggiungere una buona approssimazione dell'autocoppia, supponendo il problema considerato ben condizionato. Infatti, un residuo piccolo non assicura una buona accuratezza dell'autocoppia proprio per le possibili difficoltà derivanti dal condizionamento dell'autovalore.

Riassumiamo la procedura per calcolare numericamente le approssimazioni delle auto-coppie di A , nota come la procedura di Rayleigh-Ritz:

Algorithm 2 La procedura di Rayleigh-Ritz

1. Calcolare una base ortonormale $\{v_i\}_{i=1,\dots,m}$ del sottospazio K ; Sia $V = [v_1 \ \dots \ v_m]$;
 2. Calcolare $B_m = V^H A V$;
 3. Calcolare gli autovalori di B_m e selezionare i k desiderati $\tilde{\lambda}_i$ con $i = 1, \dots, k$ e $k \leq m$;
 4. Calcolare gli autovettori y_i di B_m associati agli autovalori $\tilde{\lambda}_i$, con $i = 1, \dots, k$, e calcolare i corrispondenti autovettori approssimati di A , $\tilde{u}_i = V y_i$, per $i = 1, \dots, k$;
-

Osserviamo solamente che è possibile sostituire gli autovettori con i vettori di Schur, che possono essere calcolati in modo stabile, per ottenere dei vettori di Schur della matrice approssimati, meno soggetti ad errori di arrotondamento rispetto agli autovettori.

3.3 Metodi di proiezione sui sottospazi di Krylov

Analizziamo ora una classe importante di metodi di proiezione per matrici di grandi dimensioni; ciò che li accomuna è la scelta del sottospazio K di approssimazione, o sottospazio destro, noto come sottospazio di Krylov; tra questi ci soffermeremo, in particolare, sul metodo di Arnoldi e sul suo utilizzo per il calcolo di autovalori.

3.3.1 Sottospazi di Krylov

Sia $A \in \mathbb{C}^{n \times n}$ e $v \in \mathbb{C}^n$, indichiamo con $K_m(A, v)$ il sottospazio di Krylov individuato da A e v e lo definiamo nel modo seguente:

$$K_m = \text{span}\{v, Av, A^2v, \dots, A^{m-1}v\}$$

Richiamiamo ora alcune proprietà dei sottospazi di Krylov, ricordando che il polinomio minimo di un vettore v è il polinomio monico non nullo di grado minore tale che $p(A)v = 0$. Tale grado è chiamato il grado di v rispetto ad A .

Proposizione 3.3.1.1. *Il sottospazio di Krylov K_m è il sottospazio che contiene tutti i vettori $x \in \mathbb{C}^n$ che possono essere scritti come $x = p(A)v$, dove p è un polinomio di grado non superiore a $m - 1$.*

Proposizione 3.3.1.2. *Sia μ il grado del polinomio minimo di v rispetto ad A ; Allora K_μ è invariante sotto l'azione di A e vale che $K_m = K_\mu \forall m \geq \mu$.*

Proposizione 3.3.1.3. *Il sottospazio di Krylov K_m ha dimensione $m \Leftrightarrow$ il grado di v rispetto ad A è maggiore di $m - 1$.*

Dimostrazione. I vettori $v, Av, A^2v, \dots, A^{m-1}v$ sono linearmente indipendenti, e quindi formano una base per K_m , se e solo se per ogni m -upla complessa $(\alpha_0, \dots, \alpha_{m-1})$, dove almeno un α_i è non nullo, la combinazione lineare $\sum_{i=0}^{m-1} \alpha_i A^i v$ è diversa da 0. Questo equivale a dire che non c'è un polinomio di grado minore o uguale di $m - 1$ per cui $p(A)v = 0$. \square

Proposizione 3.3.1.4. *Sia Q_m un qualsiasi proiettore su K_m e sia $B_m = Q_m A|_{K_m}$, cioè la sezione di A su K_m ; Allora per ogni polinomio q di grado non superiore a $m - 1$, si ha che $q(A)v = q(B_m)v$ e per ogni polinomio di grado minore o uguale di m si ha che $Q_m q(A)v = q(B_m)v$.*

Dimostrazione. Proviamo prima che $q(A)v = q(B_m)v$ per ogni polinomio q di grado $\leq m - 1$. E' sufficiente provare questa proprietà per i polinomi monici del tipo $q_i(t) = t^i$ per $i = 0, \dots, m - 1$. Dimostriamo la proprietà per induzione sul grado. Per il polinomio $q_0(t) = 1$ la proprietà è vera; supponiamo che per t^i valga

$$q_i(A)v = q_i(B_m)v$$

Per mostrare che la proprietà è vera per t^{i+1} , moltiplichiamo entrambi i membri per A ottenendo:

$$q_{i+1}(A)v = Aq_i(B_m)v$$

Se $i+1 \leq m-1$, il vettore a sinistra appartiene a K_m e perciò se moltiplichiamo entrambi i membri per Q_m abbiamo:

$$q_{i+1}(A)v = Q_m Aq_i(B_m)v$$

Osserviamo anche che, per ipotesi induttiva, $q_i(B_m)v$ appartiene a K_m e quindi

$$q_{i+1}(A)v = Q_m A|_{K_m} q_i(B_m)v = q_{i+1}(B_m)v$$

Ci manca da mostrare la proprietà per $i+1 = m$: sapendo che $q_{m-1}(A)v = q_{m-1}(B_m)v$, moltiplichiamo entrambi i membri per $Q_m A$ e otteniamo il risultato. \square

Vogliamo ora dare un'interessante caratterizzazione dei metodi di proiezione ortogonale su K_m in termini del polinomio caratteristico del problema approssimato, ovvero il polinomio caratteristico della matrice $V_m^H A V_m$, dove le colonne della matrice V_m formano una base ortonormale per il sottospazio K_m . Osserviamo che la definizione è indipendente dalla scelta della base di K_m .

Teorema 3.3.1.1. *Sia \bar{p}_m il polinomio caratteristico del problema approssimato risultante dal metodo di proiezione ortogonale su K_m ; Allora \bar{p}_m minimizza la norma $\|p(A)v\|_2$ tra tutti i polinomi monici p di grado m .*

Dimostrazione. Indichiamo con P_m il proiettore ortogonale su K_m e con B_m la sezione corrispondente di A ; Per il teorema di Hamilton-Cayley, abbiamo che $\bar{p}_m(B_m) = 0$ e perciò

$$\langle \bar{p}_m(B_m)v, w \rangle = 0, \forall w \in K_m. \quad (3.9)$$

Per la proposizione precedente, possiamo riscrivere l'equazione (3.9):

$$\langle P_m \bar{p}_m(A)v, w \rangle = 0, \forall w \in K_m.$$

Utilizzando ora il fatto che i proiettori sono autoaggiunti, abbiamo che

$$\langle \bar{p}_m(A)v, P_m w \rangle = 0, \forall w \in K_m$$

o equivalentemente

$$\langle \bar{p}_m(A)v, A^j v \rangle = 0, j = 0, \dots, m-1.$$

Scegliendo un opportuno polinomio q di grado $\leq m-1$, possiamo scrivere $\bar{p}_m(t) = t^m - q(t)$ e ottenere

$$\langle A^m v - q(A)v, A^j v \rangle = 0, j = 0, \dots, m-1.$$

Possiamo concludere osservando che le precedenti equazioni corrispondono alle equazioni normali per minimizzare la norma euclidea di $A^m v - s(A)v$ tra tutti i polinomi s di grado minore o uguale di $m-1$. \square

L'utilità di questa proprietà, caratterizzante i metodi di proiezione ortogonale su K_m , risiede nella possibilità di stabilire equivalenze matematiche tra metodi apparentemente differenti e non viene utilizzata per scopi computazionali.

3.3.2 Metodo di Arnoldi

Il metodo di Arnoldi viene visto come un metodo di proiezione ortogonale sul sottospazio di Krylov per generiche matrici non hermitiane. La procedura introdotta da Arnoldi inizia col costruire una base ortonormale del sottospazio di Krylov K_m , di cui proponiamo una variante dell'algoritmo in aritmetica esatta.

Algorithm 3 Arnoldi

1. Sia v_1 un vettore di norma unitaria.
 2. **for** $j = 1, 2, \dots, m$ **do**
 - for** $i = 1, \dots, j$ **do**
 - $h_{ij} = \langle Av_j, v_i \rangle$
 - $w_j = Av_j - \sum_{i=1}^j h_{ij}v_i$
 - end for**
 - $h_{j+1,j} = \|w_j\|_2$
 - if** $h_{j+1,j} = 0$ **then**
 - stop
 - end if**
 - $v_{j+1} = \frac{w_j}{h_{j+1,j}}$
 - end for**
-

In realtà, l'algoritmo proposto da Arnoldi nel 1951 aveva come scopo principale quello di ridurre una matrice piena in forma di Hessenberg ed era solo una supposizione quella di poter calcolare, nel caso di interruzione anticipata del processo, una buona approssimazione di alcuni autovalori della matrice.

Tuttavia, è stato poi dimostrato che ciò è sempre possibile e lo stop anticipato, che avviene quando il vettore w_j calcolato nel ciclo si annulla, fornisce semplicemente dei vantaggi in termini di accuratezza.

Osserviamo che il processo è una forma di ortogonalizzazione di Gram-Schmidt classica : il vettore Av_j viene calcolato e ortogonalizzato rispetto ai precedenti vettori v_i calcolati. Analizziamo ora alcune proprietà importanti dell'algoritmo.

Proposizione 3.3.2.1. *I vettori v_1, \dots, v_m formano una base ortonormale del sottospazio $K_m = \text{span}\{v_1, Av_1, A^2v_1, \dots, A^{m-1}v_1\}$.*

Dimostrazione. Notiamo che, per costruzione, i vettori v_1, \dots, v_m sono ortonormali. Per mostrare che generano lo spazio K_m , mostriamo per induzione su j , $j = 1, \dots, m$, che ogni vettore v_j si può scrivere come $q_{j-1}(A)v_1$, dove q_{j-1} è un polinomio di grado $j-1$. Per $j = 1$, scegliamo $q_0(A) = 1$ e chiaramente $v_1 = q_0(A)v_1$. Supponiamo il risultato vero per tutti gli interi minori o uguali a j e consideriamo v_{j+1} :

$$h_{j+1}v_{j+1} = Av_j - \sum_{i=1}^j h_{ij}v_i = Aq_{j-1}(A)v_1 - \sum_{i=1}^j h_{ij}q_{i-1}(A)v_1.$$

Da qui concludiamo che possiamo esprimere v_{j+1} come $q_j(A)v_1$, con q_j polinomio di grado j . \square

Proposizione 3.3.2.2. Denotando con V_m la matrice di dimensioni $n \times m$, le cui colonne sono i vettori v_1, \dots, v_m , e con H_m la matrice di Hessenberg $m \times m$ definita nell'algoritmo, valgono le seguenti relazioni:

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^H \quad (3.10)$$

$$V_m^H AV_m = H_m. \quad (3.11)$$

Dimostrazione. Dall'algoritmo è facile derivare la seguente relazione:

$$Av_j = \sum_{i=1}^{j+1} h_{ij} v_i \quad j = 1, \dots, m,$$

da cui si arriva alla forma matriciale di (3.10).

Per ottenere (3.11) è sufficiente moltiplicare entrambi i membri per V^H e utilizzare le relazioni di ortogonalità tra i vettori. \square

Proposizione 3.3.2.3. L'algoritmo di Arnoldi si ferma al passo j se e solo se il polinomio minimo di v_1 è di grado j . Inoltre, in questo caso il sottospazio K_j è invariante e gli autovalori e autovettori approssimati sono esatti.

Dimostrazione. (\Leftarrow) Supponiamo che il polinomio minimo di v_1 abbia grado j e che $w_j \neq 0$. Allora possiamo definire il vettore v_{j+1} e il sottospazio K_{j+1} ha dimensione $j+1$. Dalla proposizione 3.3.1.2 abbiamo che il grado del polinomio minimo deve essere maggiore o uguale di $j+1$, che è assurdo, e possiamo concludere che $w_j=0$.

(\Rightarrow) Supponiamo che $w_j=0$; per quanto appena visto, il grado del polinomio minimo di v_1 deve essere $\leq j$. Osserviamo che se fosse $\mu < j$, allora per l'implicazione precedente $w_\mu = 0$ e l'algoritmo si sarebbe dovuto fermare prima al passo μ . La seconda parte dell'enunciato deriva dalla proposizione 3.2.1.1. \square

Gli autovalori approssimati $\lambda_i^{(m)}$, calcolati tramite il metodo di proiezione su K_m , sono gli autovalori della matrice H_m e sono chiamati valori di Ritz. L'autovettore associato a $\lambda_i^{(m)}$, chiamato vettore di Ritz, è definito da $u_i^{(m)} = V_m y_i^{(m)}$, dove $y_i^{(m)}$ è autovettore di H_m associato all'autovalore $\lambda_i^{(m)}$. Di solito, un certo numero dei valori di Ritz forniscono una buona approssimazione per i corrispondenti autovalori della matrice A e la qualità dell'approssimazione cresce all'aumentare di m . Direttamente proporzionale al crescere di m , è anche però il costo dell'algoritmo sia in termini di computazione che di memoria, che va perciò preso in considerazione. Dobbiamo infatti tenere in memoria m vettori di lunghezza n più una matrice di Hessenberg e per ogni step j dobbiamo eseguire un prodotto matrice per vettore e ortogonalizzare rispetto ai precedenti vettori v_i .

Proposizione 3.3.2.4. Sia $y_i^{(m)}$ un autovettore di H_m associato all'autovalore $\lambda_i^{(m)}$ e sia $u_i^{(m)} = V_m y_i^{(m)}$; Allora

$$(A - \lambda_i^{(m)} I) u_i^{(m)} = h_{m+1,m} e_m^H y_i^{(m)} v_{m+1}$$

e perciò,

$$\left\| (A - \lambda_i^{(m)} I) u_i^{(m)} \right\|_2 = h_{m+1,m} \left| e_m^H y_i^{(m)} \right|.$$

Dimostrazione. Abbiamo che

$$\begin{aligned} A u_i^{(m)} &= A V_m y_i^{(m)} = V_m H_m y_i^{(m)} + h_{m+1,m} e_m^H y_i^{(m)} v_{m+1} \\ &= \lambda_i^{(m)} V_m y_i^{(m)} + h_{m+1,m} e_m^H y_i^{(m)} v_{m+1}. \end{aligned}$$

Da cui,

$$A V_m y_i^{(m)} - \lambda_i^{(m)} V_m y_i^{(m)} = h_{m+1,m} e_m^H y_i^{(m)} v_{m+1}.$$

□

Dalla proposizione, deduciamo che la norma del residuo è facilmente calcolabile in termini di costo, anche se non sempre indicativa di errori effettivi, ma può essere utile nel derivare criteri di arresto.

Osservazione 3.3.2.1. E' importante fare una distinzione tra l'implementazione del metodo e il metodo stesso : infatti, il metodo, descritto come processo matematico, può essere implementato in diversi modi, che sono però tutti matematicamente equivalenti. E' questo il caso del metodo di Arnoldi : una volta descritto in aritmetica esatta, ci dobbiamo preoccupare degli eventuali problemi connessi all'aritmetica finita, in particolare, delle cancellazioni dovute alle sottrazioni effettuate nel ciclo. Per ovviare a queste problematiche, si può ad esempio utilizzare la norma del vettore w , calcolato nel ciclo, come spia di eventuali cancellazioni avvenute e si può procedere ad una seconda ortogonalizzazione, per esempio, con Householder, che è uno dei metodi più affidabili.

Il limite dell'implementazione di Arnoldi è l'alto costo, al crescere di m , sia in termini di memoria che di computazione. Supponiamo di essere interessati a calcolare un'auto-coppia di A , per esempio l'autovalore con la più grande parte reale, un modo per aggirare la difficoltà, soprattutto in termini di memoria, è fare ripartire l'algorithm, ovvero fare un 'restart' : dopo m passi dell'algorithm di Arnoldi, calcoliamo l'autovettore approssimato e lo usiamo come vettore iniziale per una nuova esecuzione del metodo.

Algorithm 4 Iterative Arnoldi

1. Sia v_1 un vettore iniziale e sia m un intero;
 2. Eseguiamo m passi del metodo di Arnoldi;
 3. Calcoliamo l'autovettore destro approssimato $u_1^{(m)}$ associato all'autovalore $\lambda_1^{(m)}$. Se c'è convergenza, ci fermiamo altrimenti poniamo $v_1 = u_1^{(m)}$ e torniamo al passo 2.
-

Ci occupiamo ora di analizzare la velocità di convergenza all'autocoppia esatta dell'autocoppia approssimata, ottenuta tramite Arnoldi, stimando la distanza di un particolare autovettore u_i dal sottospazio K_m . Assumiamo per semplicità che A sia diagonalizzabile e, indicando con \mathbb{P}_{m-1}^* l'insieme dei polinomi di grado non superiore a $m - 1$ tali che $p(\lambda_i) = 1$, definiamo

$$\epsilon_i^{(m)} = \min_{p \in \mathbb{P}_{m-1}^*} \max_{\lambda \in \Lambda(A) - \lambda_i} |p(\lambda)|.$$

Lemma 3.3.2.1. *Sia $A \in \mathbb{C}^{n \times n}$ diagonalizzabile e sia v_1 il vettore iniziale del metodo di Arnoldi; supponiamo che v_1 si scriva come $v_1 = \sum_{k=1}^n \alpha_k u_k$, dove $\{u_1, \dots, u_n\}$ è una base di autovettori di norma unitaria e che $\alpha_i \neq 0$. Allora vale la seguente disuguaglianza :*

$$\|(I - P_m)u_i\|_2 \leq \xi_i \epsilon_i^{(m)},$$

dove

$$\xi_i = \sum_{k=1, k \neq i}^n \frac{|\alpha_j|}{|\alpha_i|}.$$

Dimostrazione. Dalla precedente relazione tra K_m e \mathbb{P}_{m-1} abbiamo che

$$\begin{aligned} \|(I - P_m)\alpha_i u_i\|_2 &= \min_{q \in \mathbb{P}_{m-1}} \|\alpha_i u_i - q(A)v_1\|_2 \\ &\leq \min_{q \in \mathbb{P}_{m-1}, q(\lambda_i)=1} \|\alpha_i u_i - q(A)v_1\|_2. \end{aligned}$$

Indicando con p il polinomio che realizza il minimo e ricordando l'espansione del vettore v_1 rispetto alla base di autovettori, otteniamo

$$\|(I - P_m)\alpha_i u_i\|_2 \leq \left\| \sum_{j=1, j \neq i}^n \alpha_j p(\lambda_j) u_j \right\|_2 \leq \max_{j \neq i} |p(\lambda_j)| \sum_{j \neq i} |\alpha_j|$$

Dividendo entrambi i membri per $|\alpha_i|$, otteniamo il risultato desiderato. \square

Ora però è necessario dare una stima della quantità $\epsilon_i^{(m)}$ per avere un'idea migliore di quanto valga il residuo dell'autocoppia esatta rispetto al problema approssimato (proiettato su K_m). Ci concentriamo sul determinare la quantità $\epsilon_1^{(m+1)}$ invece di $\epsilon_1^{(m)}$ per una pura convenienza notazionale (abbiamo quindi fissato l'autovalore cercato). Osserviamo che stimare $\epsilon_1^{(m)}$ equivale a determinare una stima dall'alto della distanza, in norma uniforme, tra la funzione $f(z) = 1$ e i polinomi di grado $\leq m$ della forma $p(z) = (z - \lambda_1)q(z)$, ovvero determinare il polinomio, avente come radice λ_1 , che meglio approssima la funzione $f(z) = 1$ in norma uniforme. Infatti, vale che

$$\epsilon_1^{(m+1)} = \min_{q \in \mathbb{P}_{m-1}} \|1 - (z - \lambda_1)q(z)\|_\infty.$$

Ricordiamo che un sottospazio S di funzioni continue definite su un insieme compatto Ω , generato da k funzioni ϕ_1, \dots, ϕ_k , soddisfa la condizione di Haar se ogni funzione in S ha al massimo $k - 1$ radici distinte. Questo significa che ogni combinazione lineare delle ϕ_i , $i = 1, \dots, k$, si annulla se e solo se ha k radici distinte in Ω . Sia f una funzione continua e sia p^* la miglior approssimazione uniforme della funzione f su Ω ; la differenza $f - p^*$ raggiunge il suo modulo massimo in un certo numero di punti estremi. Descriviamo meglio questa proprietà nel seguente teorema.

Teorema 3.3.2.1. *Sia f una funzione continua e S un sottospazio di dimensione k dello spazio delle funzioni continue su Ω , che soddisfa la condizione di Haar. Allora $p^* \in S$ è la miglior approssimazione in norma uniforme di f su Ω se e solo se esistono r punti estremi z_i , $i = 1, \dots, r$ in Ω e r numeri positivi μ_1, \dots, μ_r , con $k + 1 \leq r \leq 2k + 1$, tali che*

$$\sum_{i=1}^r \mu_i \overline{[f(z_i) - p^*(z_i)]} \phi(z_i) = 0 \quad \forall \phi \in S.$$

Applichiamo questo risultato alla nostra situazione. Sia $\Omega = \Lambda(A) - \lambda_1$ e $S = \text{span}\{\phi_j(z)\}_{j=1, \dots, m}$, dove $\phi_j(z) = (z - \lambda_1)z^{j-1}$. Queste funzioni generano il sottospazio S delle funzioni polinomiali di grado $\leq m$ definite su Ω che si annullano in λ_1 . S ha perciò dimensione m e il teorema afferma che esistono r autovalori, con $m + 1 \leq r \leq 2m + 1$ tali che

$$\sum_{k=1}^r \mu_k \overline{[1 - (\lambda_{k+1} - \lambda_1)q^*(\lambda_{k+1})]} \phi_j(\lambda_{k+1}) = 0 \quad \text{per } j = 1, \dots, m.$$

In generale, non sappiamo quanti punti estremi esistono, tuttavia possiamo esprimere il miglior polinomio approssimante selezionandone m tra gli r determinati e numerandoli da 2 a $m + 1$. Sia $p^* = (z - \lambda_1)q^*(z)$; possiamo scrivere $1 - p^*(\lambda_k)$ in ogni punto estremo come

$$1 - p^*(\lambda_k) = \rho e^{i\theta_k},$$

dove ρ è un numero reale positivo e θ_k è un numero reale. Conoscendo il valore che il polinomio $1 - p^*(z)$ di grado m assume in $m + 1$ punti, possiamo determinarlo univocamente utilizzando la formula di Lagrange e otteniamo che

$$1 - p^*(z) = \sum_{k=2}^{m+2} \rho e^{i\theta_k} l_k(z), \quad (3.12)$$

dove $l_k(z)$ è il polinomio di Lagrange

$$l_k(z) = \prod_{j=2, j \neq k}^{m+2} \frac{z - \lambda_j}{\lambda_k - \lambda_j}.$$

Inoltre, sapendo che $1 - p^*(\lambda_1) = 1$, dalla relazione (3.12) otteniamo che ρ è l'inverso di $\sum_{k=2}^{m+2} e^{i\theta_k} l_k(\lambda_1)$. Questo ci permette di scrivere nel seguente modo il polinomio desiderato:

$$1 - p^*(z) = \frac{\sum_{k=2}^{m+2} e^{i\theta_k} l_k(z)}{\sum_{k=2}^{m+2} e^{i\theta_k} l_k(\lambda_1)}. \quad (3.13)$$

Quanto appena visto ci permette di dimostrare il seguente teorema.

Teorema 3.3.2.2. *Ci sono r autovalori in $\Omega = \Lambda(A) - \lambda_1$, dove $m + 1 \leq r \leq 2m + 1$, in cui il polinomio ottimizzante $1 - p^*(z)$ raggiunge il suo massimo valore. Inoltre, scelti $m + 1$ autovalori $\lambda_2, \dots, \lambda_{m+2}$ tra gli r determinati, possiamo scrivere $1 - p^*(z)$ come in (3.13). In particolare,*

$$\epsilon_1^{(m+1)} = \frac{1}{\sum_{k=2}^{m+2} e^{i\theta_k} \prod_{j=2, j \neq k}^{m+2} \frac{\lambda_1 - \lambda_j}{\lambda_k - \lambda_j}}.$$

Dimostrazione. Basta osservare che il modulo massimo di $1 - p^*(z)$, assunto nei punti estremi, vale ρ , da cui deduciamo la formula per $\epsilon_1^{(m+1)}$. \square

Possiamo in realtà determinare anche una formula più pratica per $\epsilon_1^{(m+1)}$ in modo tale che dipenda solo dalla distribuzione degli autovalori della matrice. E' stato, infatti, provato il seguente teorema:

Teorema 3.3.2.3. *Sia p^* il polinomio che meglio approssima in norma uniforme la funzione $f(z) = 1$ su un insieme compatto Ω costituito da almeno $m + 1$ punti e tale che $p(\lambda_1) = 0$. Siamo $\lambda_2, \dots, \lambda_{m+2}$ $m + 1$ punti estremi in Ω . Allora vale che*

$$\epsilon_1^{(m+1)} = \frac{1}{\sum_{k=2}^{m+2} \prod_{j=2, j \neq k}^{m+2} \frac{|\lambda_j - \lambda_1|}{|\lambda_k - \lambda_j|}}. \quad (3.14)$$

Osserviamo quindi che gli autovalori che si trovano nella parte più esterna dello spettro, cioè più isolati rispetto agli altri, vengono ben approssimati dal metodo. Infatti, in questo caso la quantità $|\lambda_k - \lambda_1|$ è maggiore di $|\lambda_k - \lambda_j|$, come vediamo in figura 3.1, e perciò il prodotto considerato sarà grande cosiccome la somma. Di conseguenza, essendo $\epsilon_i^{(m)}$ l'inverso di tale quantità, sarà sufficientemente piccolo da garantire una buona convergenza.

Mostriamo meglio questo fatto attraverso alcuni esempi.

Esempio 1.

Supponiamo che gli autovalori abbiano una distribuzione uniforme su un segmento, ovvero una distribuzione del tipo

$$\lambda_k = \frac{k-1}{n-1}, \quad k = 1, 2, \dots, n.$$

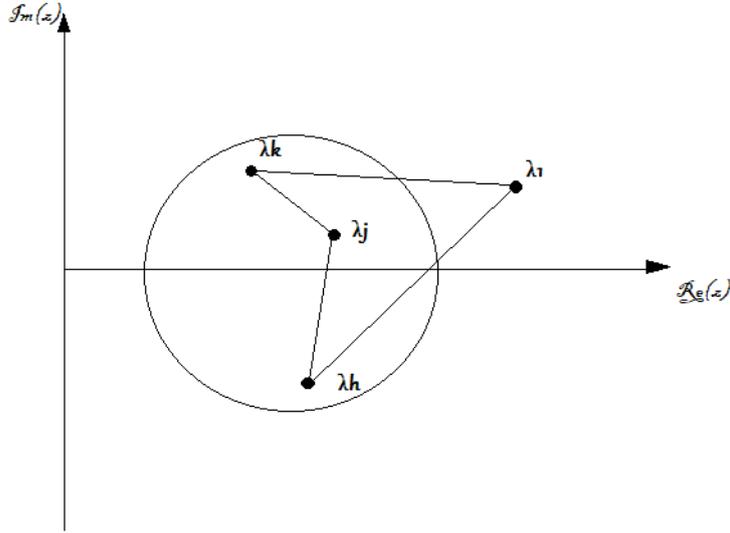


Figura 3.1: λ_1 si trova nella parte esterna dello spettro

Consideriamo il caso particolare in cui $m = n - 1$. Ciò implica che non abbiamo possibilità di scelta degli $m + 1$ autovalori del teorema ma selezioniamo tutti i rimanenti togliendo λ_1 . Otteniamo che

$$\begin{aligned}
 (\epsilon_1^{(m)})^{-1} &= \sum_{j=2}^{m+1} \prod_{k=2, k \neq j}^{m+1} \frac{|k-1|}{|k-j|} \\
 &= \sum_{j=2}^{m+1} \frac{m!}{(j-1)!(m-(j-1))!} \\
 &= \sum_{j=1}^m \binom{m}{j} = 2^m - 1.
 \end{aligned}$$

Quindi

$$\epsilon_1^{(m)} = \frac{1}{2^m - 1}.$$

Esempio 2.

Consideriamo ora una distribuzione uniforme di autovalori sulla circonferenza unitaria, ovvero del tipo

$$\lambda_k = e^{i \frac{2(k-1)\pi}{n}}, \quad k = 1, 2, \dots, n.$$

Assumiamo anche in questo caso che $m = n - 1$ e mostriamo che

$$\epsilon_1^{(m)} = \frac{1}{m}.$$

Nominiamo gli autovalori del teorema come $\lambda_2, \dots, \lambda_n$ e definiamo $w = e^{\frac{2\pi i}{n}}$. Possiamo scrivere ogni termine prodotto della sommatoria come

$$\begin{aligned} \prod_{k=2, k \neq j}^{m+1} \frac{|w^{k-1} - 1|}{|w^{k-1} - w^{j-1}|} &= \prod_{k=1, k \neq j}^m \frac{|w^k - 1|}{|w^k - w^j|} \\ &= \left[\prod_{k=1}^m |w^k - 1| \right] \left[|1 - w^j| \prod_{k=1, k \neq j}^m |w^k - w^j| \right]^{-1}. \end{aligned}$$

Ricordando che w^k sono le potenze dell' n -esima radice di 1, rinominando i prodotti al denominatore, abbiamo che il numeratore e il denominatore hanno lo stesso modulo e perciò il prodotto vale $1 \forall j$. Di conseguenza, la sommatoria vale m , da cui concludiamo. Abbiamo analizzato due casi apparentemente simili di distribuzioni uniformi di autovalori che però si comportano in modo diverso per quanto riguarda la convergenza: gli autovalori complessi distribuiti sul cerchio convergono molto più lentamente e ciò può succedere anche se la matrice è normale perchè la problematica è causata univocamente dalla distribuzione degli autovalori.

Possiamo anche dare una più semplice e esplicita stima dall'alto di $\epsilon_i^{(m)}$, utilizzando risultati di ottimizzazione importanti sui polinomi di Chebyshev.

Proposizione 3.3.2.5. *Sia $C(c, \rho)$ un cerchio di centro c e raggio ρ che racchiude tutti gli autovalori di A escluso λ_1 . Allora*

$$\epsilon_i^{(m)} \leq \left(\frac{\rho}{|\lambda_1 - c|} \right)^{m-1}.$$

Dimostrazione. Utilizzando il polinomio $q(z) = (z - c)^{m-1} / (\lambda_1 - c)^{m-1}$, otteniamo un limite superiore per $\epsilon_i^{(m)}$:

$$\epsilon_i^{(m)} \leq \max_{j=2,3,\dots,n} \left(\frac{|\lambda_j - c|}{|\lambda_1 - c|} \right)^{m-1} \leq \rho^{m-1} / |\lambda_1 - c|^{m-1}.$$

□

Teorema 3.3.2.4. *Supponiamo che tutti gli autovalori di A tranne λ_1 si trovino all'interno dell'ellisse di centro c , con i fuochi in $c + e$ e $c - e$ e semiasse maggiore pari ad a . Allora*

$$\epsilon_i^{(m)} \leq \frac{C_{m-1} \left(\frac{a}{e} \right)}{|C_{m-1} \left(\frac{\lambda_1 - c}{e} \right)|},$$

dove C_{m-1} è il polinomio di Chebyshev di grado $m - 1$ del primo tipo. Inoltre, la differenza relativa tra il membro di sinistra e il membro di destra tende a 0 quando m tende all'infinito.

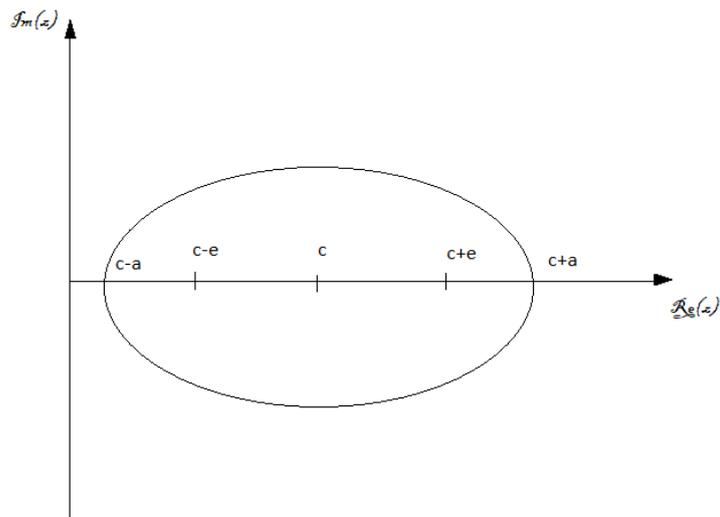


Figura 3.2: Ellisse contenente lo spettro di A

Capitolo 4

Applicazione del metodo di Arnoldi al calcolo del Pagerank

Nel capitolo precedente, ci siamo soffermati sui metodi di proiezione per il calcolo di autovalori di generiche matrici, in particolare sul metodo di Arnoldi e la sua modifica iterativa. Osserviamo che l'importanza di questo algoritmo risiede nel fatto che, al suo interno, compaiono solo prodotti matrice per vettore e può essere perciò applicato anche a matrici di grandi dimensioni, come la matrice di Google, risultando adatto per il calcolo del Pagerank. Prima di analizzare eventuali modifiche dell'algoritmo per migliorarne la prestazione nell'ambito di nostro interesse, descriviamo un nuovo algoritmo 'raffinato' proposto da Zhongxiao Jia in [5], che migliora la convergenza degli autovettori approssimati agli autovettori esatti.

4.1 Algoritmo di Arnoldi 'raffinato'

Analisi precedenti di Zhongxiao Jia hanno mostrato che la scelta dei vettori di Ritz come autovettori approssimati, calcolati tramite metodi di proiezione, non sia sempre ottimale. Infatti la convergenza dei valori di Ritz agli autovalori della matrice non è condizione sufficiente affinché convergano anche gli autovettori di Ritz.

Per gli autovalori approssimati, vale la seguente stima:

$$\left| \lambda_i - \lambda_i^{(m)} \right| \leq \left\| P_i^{(m)} \right\| \gamma_m \|(I - P_m)u_i\|,$$

dove $P_i^{(m)}$ è il proiettore spettrale associato a $\lambda_i^{(m)}$.

Abbiamo già studiato la convergenza della quantità $\|(I - P_m)u_i\|$ al crescere di m ; perciò se $\left\| P_i^{(m)} \right\|$ è limitata, concludiamo che $\lambda_i^{(m)}$ è una buona approssimazione per λ_i . Non possiamo dire la stessa cosa per quanto riguarda gli autovettori di Ritz perchè entrano in gioco altre quantità che, a priori, per matrici non simmetriche, non sono limitate. Questo

succede, ad esempio, nel caso in cui il problema agli autovalori di A è ben condizionato ma non lo è quello di H_m .

L'idea è perciò quella di utilizzare altri vettori come approssimazione degli autovettori cercati, che chiamiamo vettori 'raffinati' :

$\forall \lambda_i^{(m)}$, cerchiamo $\varphi_i^{(m)} \in K_m(A, v)$ vettore di norma unitaria che soddisfi

$$\left\| (A - \lambda_i^{(m)} I) \varphi_i^{(m)} \right\| = \min_{u \in K_m(A, v), \|u\|=1} \left\| (A - \lambda_i^{(m)} I) u \right\|. \quad (4.1)$$

Per questa particolare scelta di vettori, la convergenza è garantita dalla convergenza dei valori di Ritz agli autovalori:

Proposizione 4.1.1. *Sia $v_1 = \sum_{j=1}^n \alpha_j u_j$ con $\alpha_i \neq 0$; sia $S = (u_1 \dots u_n)$ e $\xi_i^{(m)} = \sum_{i \neq j} |\lambda_j - \lambda_i^{(m)}| \frac{|\alpha_j|}{|\alpha_i|}$; Allora*

$$\left\| (A - \lambda_i^{(m)}) \varphi_i^{(m)} \right\| \leq k(S) \left(|\lambda_i - \lambda_i^{(m)}| + \xi_i^{(m)} \min_{p \in \mathbb{P}_{m-1}^*} \max_{j \neq i} |p(\lambda_j)| \right).$$

Dobbiamo osservare che la rapidità con cui $\left\| (A - \lambda_i^{(m)}) \varphi_i^{(m)} \right\|$ converge a 0 dipende anche da $k(S)$, e perciò la tolleranza desiderata può essere raggiunta, in alcuni casi, per un valore di m grande, che richiede un costo computazionale troppo elevato. Il nostro obiettivo è quello di avere convergenza entro un valore di m sufficientemente piccolo e una buona strategia può essere quella di eseguire un metodo di Arnoldi iterativo. Un'appropriata combinazione lineare dei vettori $\varphi_i^{(m)}$ calcolati, risulta essere una buona scelta del nuovo vettore iniziale v_1 , che permette di amplificare i termini $|\alpha_i|$ associati agli autovettori cercati e ridurre i termini $|\alpha_j|$ associati agli autovettori non desiderati. Ci occupiamo ora di risolvere il problema di minimo in (4.1).

Proposizione 4.1.2. *Siano $z_i^{(m)}$ i vettori destri associati a $\sigma_{\min}(H_{m+1, m} - \lambda_i^{(m)} \tilde{I})$, dove $H_{m+1, m}$ è ottenuta aggiungendo a H_m una riga la cui unica componente non nulla è $h_{m+1, m}$ (se non si ha interruzione anticipata di Arnoldi), e \tilde{I} rappresenta la matrice identità di dimensioni $(m+1) \times m$; Allora, posto $\varphi_i^{(m)} = V_m z_i^{(m)}$, vale*

$$\left\| (A - \lambda_i^{(m)} I) \varphi_i^{(m)} \right\| = \sigma_{\min}(H_{m+1, m} - \lambda_i^{(m)} \tilde{I}).$$

Dimostrazione.

$$\begin{aligned} \left\| (A - \lambda_i^{(m)} I) \varphi_i^{(m)} \right\| &= \min_{\|z\|=1} \left\| (A - \lambda_i^{(m)} I) V_m z \right\| \\ &= \min_{\|z\|=1} \left\| V_{m+1} (H_{m+1, m} - \lambda_i^{(m)} \tilde{I}) z \right\| \\ &= \min_{\|z\|=1} \left\| (H_{m+1, m} - \lambda_i^{(m)} \tilde{I}) z \right\| \\ &= \sigma_{\min}(H_{m+1, m} - \lambda_i^{(m)} \tilde{I}). \end{aligned}$$

□

Abbiamo già osservato che, nel caso in cui il metodo di Arnoldi si ferma prima, le auto-coppie approssimate sono, in realtà, esatte e vale proprio che $\varphi_i^{(m)} = u_i^{(m)}$. Non ha senso perciò utilizzare l'algoritmo 'raffinato'. Ciò può succedere anche quando $h_{k+1,k} \leq tol$ per qualche $k \leq m$ e i vettori di Ritz risultano essere buone approssimazioni, rendendo non necessario il calcolo della decomposizione in valori singolari, il cui costo è comunque trascurabile confrontato al costo del processo di Arnoldi.

Algorithm 5 Refined iterative Arnoldi

1. Siano r il numero di autocoppie richieste, m il numero di steps di Arnoldi, tol la tolleranza richiesta, v_1 vettore iniziale reale unitario;
 2. Dopo m steps di Arnoldi, otteniamo H_m e V_m ;
 3. Calcoliamo $\lambda_1^{(m)}, \dots, \lambda_m^{(m)}$ autovalori di H_m ; selezioniamo gli r autovalori approssimanti gli autovalori desiderati;
 4. Se $h_{m+1,m} = 0$ scegliamo i vettori di Ritz $u_i^{(m)}$ come approssimazione di u_i ; altrimenti calcoliamo $\varphi_i^{(m)}$; Se $\left\| (A - \lambda_i^{(m)} I) \varphi_i^{(m)} \right\| < tol$ allora stop; altrimenti continua
 5. Costruiamo un nuovo vettore iniziale v_1 , combinazione lineare di $\varphi_i^{(m)}$ per $i = 1, \dots, r$
-

Occupiamoci ora dell'applicazione al calcolo del Pagerank. E' importante osservare che, per valori di α non troppo vicini a 1, il metodo delle potenze è efficiente sia in termini di convergenza sia di costo ed è perciò preferibile a metodi tipo Arnoldi che hanno un costo computazionale più elevato. Vogliamo però analizzare i casi in cui α è vicino a 1, ovvero in cui il metodo delle potenze ha una lenta velocità di convergenza.

In [6], Golub e Greif si sono concentrati sulle problematiche effettive presenti nell'algoritmo di Arnoldi iterativo e 'raffinato' per il calcolo del Pagerank. Hanno infatti notato che il valore di Ritz più grande in modulo della matrice H_m , utilizzato per approssimare l'autovalore più grande in modulo della matrice di Google P , poteva essere complesso e non è desiderabile utilizzare l'aritmetica complessa in problemi di così larga scala. Inoltre, quando gli autovalori non sono ben separati, la convergenza di tale valore di Ritz a 1 può essere irregolare, condizionando quindi anche la convergenza dell'autovettore. Un'alternativa efficiente può essere quella di imporre che il più grande valore di Ritz sia 1 senza calcolarlo, dato che è noto che 1 è l'autovalore di P da approssimare e otteniamo il seguente algoritmo:

Algorithm 6 Arnoldi per il Pagerank

1. Sia m intero, v_1 vettore iniziale di norma unitaria e $\sigma_m = 1$
 2. **while** $\sigma_m < \epsilon$ **do**
 - $[V_{m+1}, H_{m+1,m}] = \text{Arnoldi}(A, v_1, m);$
 - $H_{m+1,m} - \tilde{I} = U\Sigma V^T;$
 - $\sigma_m = \Sigma(m, m);$
 - $v = V[:, m];$
 - $v_1 = V_m v;$**end while**
-

La miglior convergenza è dovuta al fatto che il più piccolo valore singolare non nullo della matrice $H_{m+1,m} - \tilde{I}$ converge a 0 regolarmente. Inoltre, l'algoritmo è un po' meno costoso rispetto ad Arnoldi 'raffinato' perchè non viene calcolato nessun valore di Ritz. Per quanto visto prima, il criterio di arresto implementato è equivalente ad utilizzare la quantità $\|Av_1 - v_1\|$ ma è più economico. In generale, rispetto al metodo delle potenze, l'algoritmo presentato ha un costo più elevato per iterazione ma la speranza è che converga in un numero di passi minore nei casi in cui i valori di α sono molto vicini ad 1.

4.2 Esperimenti numerici

Testiamo ora il metodo delle potenze e l'algoritmo di tipo Arnoldi, di cui proviamo diversi valori di k , su alcune matrici disponibili sul sito <http://www.cise.ufl.edu/research/sparse/matrices>. L'obiettivo è confrontare il numero di iterazioni necessarie alla convergenza e il valore del residuo ottenuto, al crescere del parametro α . Scegliamo come vettore iniziale il vettore $\frac{1}{n}\mathbf{1}$, dove n indica il numero di righe/colonne della matrice; implementiamo come criterio di arresto il valore del residuo in norma 1, che è la scelta naturale per il metodo delle potenze, ovvero $\|Ax^{(k)} - x^{(k)}\|_1$.

Matrici utilizzate:

- web-Google, $n = 916428$ con 176974 nodi non connessi
- Stanford, $n = 281903$ con 172 nodi non connessi
- Stanford-Berkeley, $n = 683446$ con 68062 nodi non connessi

Confrontiamo graficamente la convergenza del metodo delle potenze, del metodo estrapolativo e del metodo di Arnoldi con $k = 4$ per differenti valori di α .

α	Power	<i>Power - tr</i>	k=4	k=8	k=16
0.85	71	68	80	56	48
0.9	107	101	116	88	80
0.95	215	197	220	168	128
0.99	1065	882	996	696	288

Tabella 4.1: Prodotti matrice per vettore corrispondenti a valori di α diversi scegliendo come tolleranza 10^{-7} per la matrice Google

Nella tabella abbiamo riportato il numero di prodotti matrice per vettore calcolati dai diversi algoritmi : per $\alpha = 0.85, 0.9$ abbiamo un leggero miglioramento del metodo delle potenze apportato dall'estrapolazione, eseguita ogni 60 iterazioni, e da Arnoldi, soprattutto per valori di k più elevati, che comportano però un maggior costo in termini di memoria; per valori di α più vicini a 1, il miglioramento di Arnoldi, rispetto al metodo delle potenze, è notevole in termini di costo computazionale ma il prezzo da pagare è sempre una maggiore memoria richiesta.

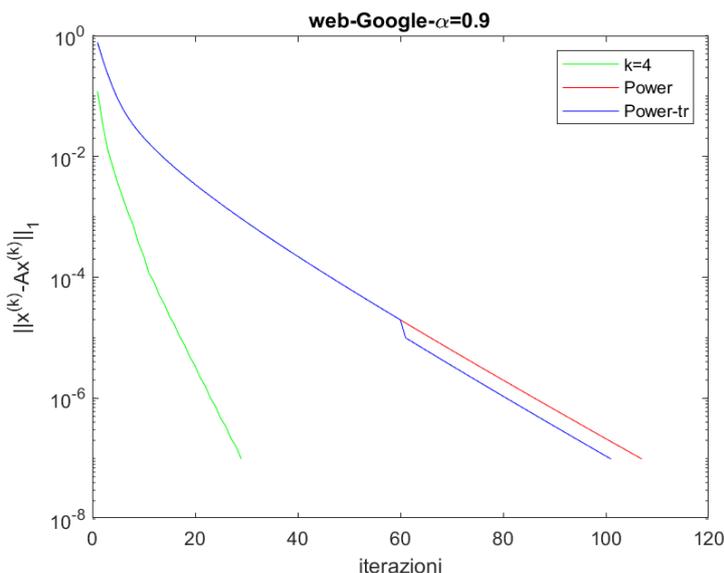


Figura 4.1: Analisi della convergenza per la matrice di Google con $\alpha = 0.9$

Nel grafico riportiamo il valore del residuo calcolato ad ogni iterazione per i diversi algoritmi, avendo scelto esso come criterio di arresto e avendo richiesto una tolleranza uguale a 10^{-7} . Osserviamo che il numero di iterazioni per Arnoldi con $k = 4$ è inferiore ma bisogna tenere in considerazione il maggiore costo computazionale per iterazione. Nei grafici successivi riportiamo sempre la decrescita del residuo in funzione del numero di iterazioni per valori di α sempre più vicini a 1. Per $\alpha = 0.999$ e una tolleranza di 10^{-5} , la scelta di Arnoldi con $k = 4$ è di gran lunga preferibile rispetto al metodo delle potenze

: confrontandoli in termini di numero di prodotti matrice per vettore, il metodo delle potenze ne richiede circa 6000 mentre Arnoldi circa la metà.

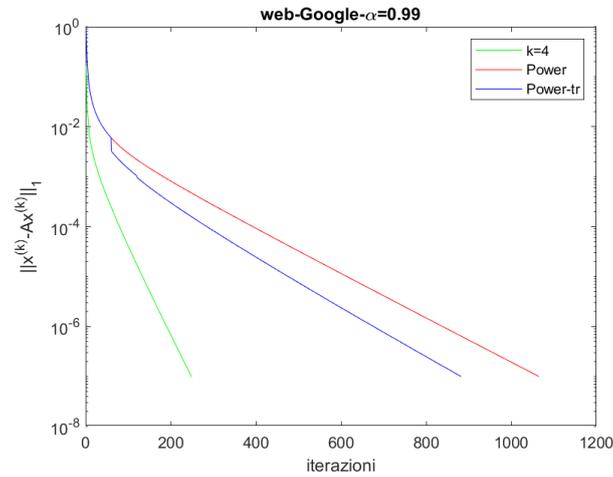


Figura 4.2: Analisi della convergenza per la matrice di Google con $\alpha = 0.99$

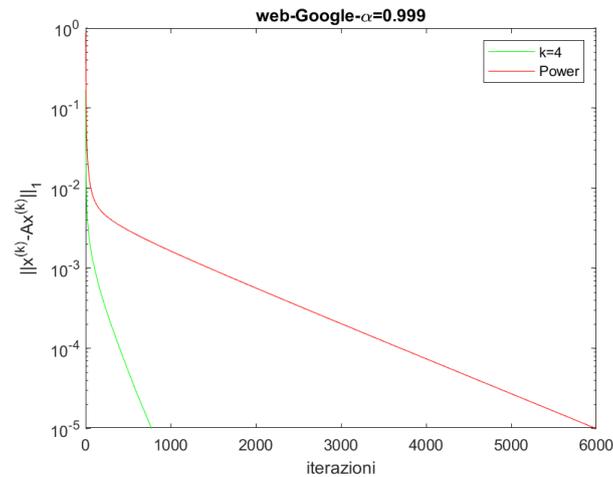


Figura 4.3: Analisi della convergenza per la matrice di Google con $\alpha = 0.999$

Eseguiamo la stessa analisi per la matrice di Stanford, che ha dimensioni inferiori rispetto alla matrice web-Google. Riportiamo nella tabella il numero di prodotti matrice per vettore, che rappresentano il costo più elevato nell'applicazione dei metodi e ci permettono di confrontarli meglio : il numero di iterazioni per raggiungere un certo valore del residuo non è molto indicativo dato che gli algoritmi hanno costi computazionali diversi per iterazione.

α	Power	<i>Power - tr</i>	k=4	k=8	k=16
0.85	77	68	80	64	64
0.9	117	100	112	96	80
0.95	236	193	188	136	144
0.99	1166	789	708	488	336

Tabella 4.2: Prodotti matrice per vettore corrispondenti a valori di α diversi scegliendo come tolleranza 10^{-7} per la matrice Stanford

Eseguiamo l'estrapolazione ogni 60 iterazioni e possiamo osservare che il comportamento dei tre metodi è molto simile a quello visto per la matrice web-Google. Nel grafico riportiamo il valore del residuo in funzione delle iterazioni per $\alpha = 0.95$.

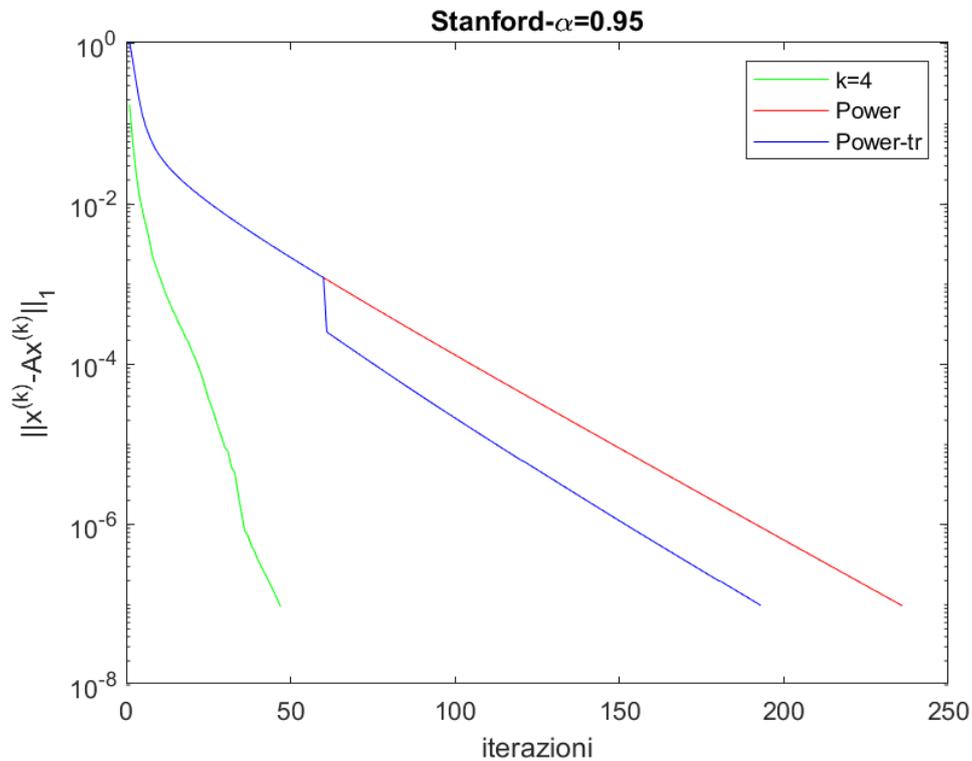


Figura 4.4: Analisi della convergenza per la matrice di Stanford con $\alpha = 0.95$

Riportiamo i risultati del tutto simili ottenuti per la matrice Stanford-Berkeley, soffermandoci sulla convergenza del più piccolo valore singolare che, come sostenuto da Golub e Greif in [6], è più regolare della convergenza del più grande valore di Ritz all'autovalore 1.

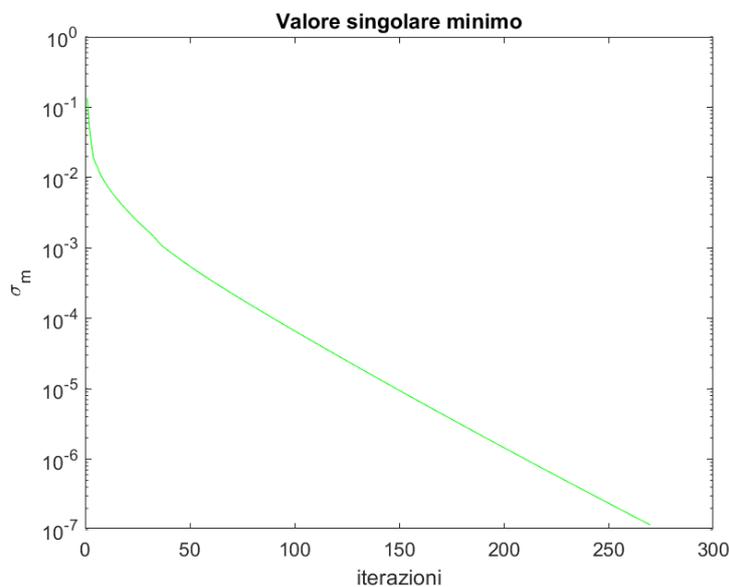


Figura 4.5: Convergenza del più piccolo valore singolare per la matrice Stanford-Berkeley con $k = 4$ e $\alpha = 0.99$

Per valori di k più elevati, il miglioramento della prestazione dato da Arnoldi è indipendente dal valore di α , escludendo la maggiore memoria richiesta. Per valori di k più bassi, in questo caso l'applicazione dell'estrapolazione porta vantaggi leggermente migliori.

α	Power	$Power - tr$	k=4	k=8	k=16
0.85	78	69	84	72	64
0.9	119	103	128	104	96
0.95	241	194	244	184	160
0.99	1202	822	1080	816	688

Tabella 4.3: Prodotti matrice per vettore corrispondenti a valori di α diversi scegliendo come tolleranza 10^{-7} per la matrice Stanford-Berkeley

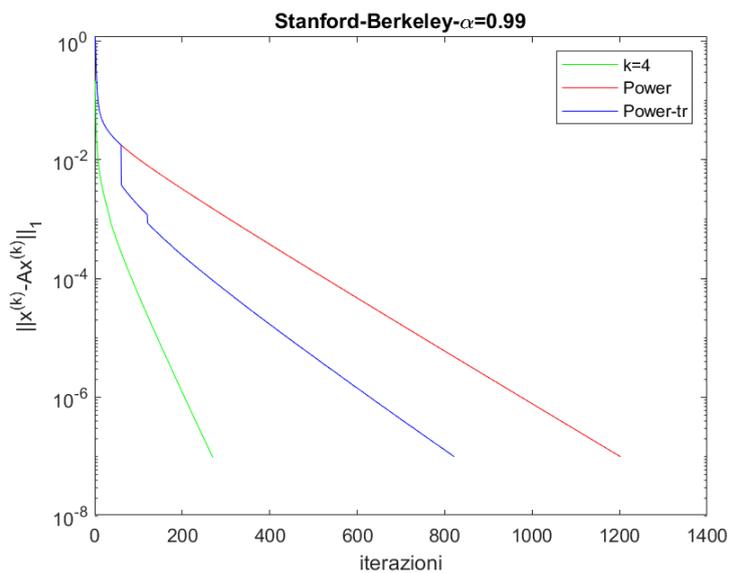


Figura 4.6: Analisi della convergenza per la matrice di Stanford-Berkeley con $\alpha = 0.99$

Nel grafico riportiamo il valore del residuo in funzione del numero di iterazioni per $\alpha = 0.99$.

Bibliografia

- [1] V.Simoncini, *Dispense per il corso di Data Mining*
- [2] Xueyuan Tan, *A new extrapolation method for Pagerank Computation, J.Comput.Appl.Math 313(2017) 383-392*
- [3] D. Watkins, *The matrix Eigenvalue Problem : GR and Krylov Subspace Methods,SIAM,Philadelphia,PA,2003*
- [4] Y.Saad, *Numerical Methods for Large Eigenvalue Problems, Revised,SIAM,2011*
- [5] Z. Jia, *Refined iterative algorithms based on Arnoldi's process for large unsymmetric eigenproblems, Linear Algebra Appl. 259(1997) 1-23*
- [6] G.Golub, C.Greif, *An Arnoldi-type algorithm for computing Pagerank, BIT 46(2006) 759-771*