

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE
Corso di Laurea in Matematica

Metodi di estrapolazione
per il calcolo
del vettore PageRank

Tesi di Laurea in Analisi Numerica

Relatore:
Chiar.ma Prof.ssa
VALERIA SIMONCINI

Presentata da:
FRANCESCO CONTI

Sessione invernale
Anno Accademico 2017/2018

*"Solo nel silenzio, la parola
solo nelle tenebre, la luce
solo nel morir, la vita.
Fulgido il volo del falco, nel cielo vuoto".
- Ursula K. Le Guin*

Elenco delle figure

2.1	Velocità di convergenza del metodo delle potenze per diversi valori di c	18
2.2	Residuo del metodo delle potenze aumentando c	20
2.3	Zoom sul gradino nel residuo all'aggiornamento di c	20
2.4	Residuo del metodo delle potenze riducendo c nel corso delle iterate.	21
3.1	Confronto del metodo delle potenze standard e accelerato con Aitken per $c = 0.85$	26
3.2	Confronto del metodo potenze standard e accelerato con Aitken per $c = 0.99$	27
3.3	Applicazione periodica dell'estrapolazione di Aitken per diversi valori di c , estrapolando ogni 10 iterate.	28
3.4	Applicazione periodica dell'estrapolazione di Aitken per diversi valori di c , estrapolando ogni 50 iterate.	29
3.5	Confronto del metodo delle potenze standard e accelerato con Aitken per $c = 0.5$	30
3.6	Confronto del metodo delle potenze standard e accelerato con Aitken per $c = 0.99$	31
3.7	Confronto del residuo per tre diverse matrici.	32
4.1	Confronto del metodo delle potenze standard e accelerato con epsilon per $c = 0.85$	38

4.2	Confronto del metodo delle potenze standard e accelerato con epsilon per $c = 0.99$	38
4.3	Applicazione periodica dell'extrapolazione epsilon per diversi valori di c	39
4.4	Confronto dell'extrapolazione epsilon e di Aitken per $c = 0.5$	40
4.5	Confronto dell'extrapolazione epsilon e di Aitken per $c = 0.99$	41
5.1	Confronto del metodo delle potenze standard e accelerato con QE per diversi valori di c	49
5.2	Confronto del metodo delle potenze standard e accelerato con QE per vari valori di c	50
5.3	Confronto del metodo di extrapolazione quadratica, epsilon e Aitken per $c = 0.5$	51
5.4	Confronto del metodo di extrapolazione quadratica, epsilon e Aitken per $c = 0.85$	51
5.5	Confronto del metodo di extrapolazione quadratica, epsilon e Aitken per $c = 0.5$	52
5.6	Confronto del metodo di extrapolazione quadratica, epsilon e Aitken per $c = 0.99$	53
5.7	Confronto del residuo ottenuto extrapolando con diverse frequenze per $c = 0.85$	54
5.8	Confronto del residuo ottenuto extrapolando con diverse frequenze per $c = 0.99$	55

Elenco delle tabelle

2.1	Dataset utilizzato.	17
3.1	Secondi impiegati per arrivare a convergenza.	32
4.1	Secondi impiegati per la convergenza.	41
5.1	Tempi di convergenza per la matrice “Stanford”.	56
5.2	Tempi di convergenza per la matrice “Google”.	56
5.3	Tempi di convergenza per la matrice “LiveJournal”.	57

Indice

Introduzione	1
1 Il calcolo del vettore PageRank	3
1.1 Il vettore PageRank	3
1.2 La matrice di Hyperlink H	5
1.3 Il grafo del Web	7
2 Il metodo delle potenze	11
2.1 Scelta del metodo	11
2.2 Derivazione dell'algoritmo	12
2.3 Il metodo delle potenze per il vettore PageRank	13
2.4 Algoritmo	14
2.5 Costo computazionale	15
2.6 Velocità di convergenza	16
2.7 Dati campione	17
2.8 Risultati sperimentali	18
2.8.1 Il metodo delle potenze incrementando c	19
2.8.2 Il metodo delle potenze riducendo c	21
2.8.3 Conclusioni	22
3 Estrapolazione di Aitken	23
3.1 Il metodo di estrapolazione di Aitken	23
3.2 Algoritmi	24
3.3 Costo computazionale	25

3.4	Risultati sperimentali	26
3.4.1	Estrapolazione periodica di Aitken	28
3.4.2	Iterata ottimale di estrapolazione	30
3.4.3	La sparsità della matrice e il residuo	32
3.4.4	Conclusioni	33
4	Estrapolazione epsilon	35
4.1	Il metodo di estrapolazione epsilon	35
4.2	Algoritmi	36
4.3	Costo computazionale	37
4.4	Risultati sperimentali	37
4.4.1	Estrapolazione periodica epsilon	39
4.4.2	Confronto epsilon e Aitken	39
4.4.3	Tempi di convergenza	41
4.4.4	Conclusioni	42
5	Estrapolazione quadratica	43
5.1	Il metodo di estrapolazione quadratica	43
5.2	Algoritmi	46
5.3	Costo computazionale	47
5.4	Risultati sperimentali	49
5.4.1	Confronto QE, epsilon e Aitken	50
5.4.2	Frequenza di applicazione e residuo	54
5.4.3	Frequenza ottimale di estrapolazione	55
5.4.4	Conclusioni	57
	Conclusioni	59
	Bibliografia	61
	Ringraziamenti	63

Notazione

Introduciamo alcune notazioni che verranno usate nell'elaborato.

Definizione.

$x^{(k)}$ indica l'iterata k -esima, non la derivata di ordine k .

Definizione.

$\mathbf{1}$ è il vettore $(1, \dots, 1)^T \in \mathbb{R}^n$

Definizione.

x^* è il trasposto coniugato.

Introduzione

Questo elaborato ha l'obiettivo di presentare alcuni metodi estrapolativi per il calcolo del vettore PageRank.

In particolare si prenderanno in esame tre metodi: l'estrapolazione di Aitken, l'estrapolazione epsilon e l'estrapolazione quadratica.

L'importanza di accelerare il calcolo del vettore PageRank è fondamentale data la grandezza del problema in questione. Il tempo richiesto per il calcolo di un singolo vettore con il metodo standard può essere infatti estremamente elevata. Inoltre, recenti approcci personalizzati per la ricerca possono richiedere il calcolo di vari vettori PageRank.

È in quest'ottica che si inseriscono pertanto i metodi estrapolativi.

Mostriamo come l'utilizzo di questi metodi riesca ad accelerare la velocità di convergenza del 25 – 300%.

Capitolo 1

Il calcolo del vettore PageRank

1.1 Il vettore PageRank

Definizione 1.1. Il vettore **PageRank** è il vettore che ha come componenti le importanze delle diverse pagine Web.

$$\mathcal{I} = (\mathcal{I}(P_1), \mathcal{I}(P_2), \dots, \mathcal{I}(P_n)) \quad (1.1)$$

dove P_1, \dots, P_n rappresentano le diverse pagine Web e $\mathcal{I}(P_1), \dots, \mathcal{I}(P_n)$ le rispettive importanze.

Affinché si possa determinare il vettore PageRank è quindi necessario determinare ciascuna componente, cioè definire l'importanza di ogni pagina Web.

“Una pagina Web è importante se a lei puntano molte pagine importanti”¹.

Osservazione 1. La dicitura “pagine importanti” è fondamentale perché permette di limitare l'influenza dello *spam*.

Infatti un gran numero di pagine vuote che puntano tutte a una stessa pagina non deve conferire importanza alla pagina puntata.

A partire da questa idea, possiamo quindi definire l'importanza di una pagina Web.

¹Idea di Larry Page, 1996

Definizione 1.2. L'importanza della pagina P_k è:

$$\mathcal{I}(P_k) = \frac{1}{l_1}\mathcal{I}(P_1) + \frac{1}{l_2}\mathcal{I}(P_2) + \cdots + \frac{1}{l_m}\mathcal{I}(P_m) \quad (1.2)$$

dove P_1, \dots, P_m sono le pagine che puntano a P_k , $\mathcal{I}(P_1), \dots, \mathcal{I}(P_m)$ la rispettiva importanza e l_1, \dots, l_m il rispettivo numero di outlinks.

Da un punto di vista matriciale possiamo definire:

Definizione 1.3. La **matrice di Hyperlink** $H = (h_{ij}) \in \mathbb{R}^{n \times n}$ è tale che:

$$h_{ij} = \begin{cases} \frac{1}{l_i} & \text{se } P_i \text{ punta a } P_j \\ 0 & \text{altrimenti} \end{cases}$$

Dalle Definizioni (1.1) e (1.2) si ottiene:

$$\mathcal{I} = H\mathcal{I} \quad (1.3)$$

ovvero un problema agli autovalori con autovalore $\lambda = 1$ e autovettore associato \mathcal{I} , cioè il nostro vettore PageRank.

Esempio 1.1.

$$H = \begin{pmatrix} 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 1 & 0 \\ \frac{1}{3} & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} \mathcal{I}(P_1) \\ \mathcal{I}(P_2) \\ \mathcal{I}(P_3) \\ \mathcal{I}(P_4) \\ \mathcal{I}(P_5) \end{pmatrix} = H \begin{pmatrix} \mathcal{I}(P_1) \\ \mathcal{I}(P_2) \\ \mathcal{I}(P_3) \\ \mathcal{I}(P_4) \\ \mathcal{I}(P_5) \end{pmatrix}$$

che è proprio $\mathcal{I} = H\mathcal{I}$.

Osservazione 2. La matrice H è, nel caso generico, *stocastica per colonna*, ovvero:

- è estremamente sparsa
- ha elementi tutti non negativi
- $\sum_{i=1}^n h_{ij} = 1$

Il problema del calcolo del vettore PageRank può essere studiato da due punti di vista, quello matriciale, con la matrice di Hyperlink H , e quello dei grafi.

Esamineremo separatamente entrambi gli aspetti che, comunque, converranno al medesimo risultato.

1.2 La matrice di Hyperlink H

Riprendiamo la definizione della matrice di Hyperlink H ,

$$H = (h_{ij}) \quad \text{e} \quad h_{ij} = \begin{cases} \frac{1}{l_i} & \text{se } P_i \text{ punta a } P_j \\ 0 & \text{altrimenti} \end{cases}$$

Essa risulta essere stocastica per colonna, tranne in alcuni casi. Queste eccezioni si verificano quando una pagina Web P_i non presenta alcun outlink; in corrispondenza della colonna i -esima, H avrà quindi una colonna di soli zeri. Per eliminare queste problematiche è necessario fare una piccola modifica di H .

Definiamo:

$$d_i = \begin{cases} 1 & \text{se } l_i = 0 \\ 0 & \text{altrimenti} \end{cases} \quad \text{e} \quad \vec{v} = \frac{1}{n} \mathbf{1}.$$

Infine definiamo la nuova matrice:

$$P' = H + \vec{v} \cdot \vec{d}^T, \tag{1.4}$$

cosicché P' è sempre stocastica per colonna.

Osservazione 3. Questa modifica sostituisce una colonna di $\frac{1}{n}$ a ogni colonna di zeri di H , ovvero aggiunge una probabilità uniforme che un navigatore, giunto su una pagina senza outlinks, passi a una qualsiasi altra pagina.

Consideriamo ora il nuovo problema agli autovalori:

$$\mathcal{I} = P'\mathcal{I}$$

che dobbiamo risolvere per l'autovalore $\lambda = 1$.

È possibile che questo autovalore non sia semplice, in questo caso esso ha più autovettori associati e il problema agli autovalori che stiamo cercando di risolvere non ha una soluzione unica. Per garantire l'unicità della soluzione ci servono nuove definizioni.

Definizione 1.4. Una matrice A si dice **irriducibile** se esiste una matrice di permutazione T tale che:

$$TAT^T = \begin{pmatrix} X & Y \\ 0 & Z \end{pmatrix}$$

Teorema 1.2.1. Teorema di Perron-Frobenius.

Sia $A \in \mathbb{R}^{n \times n}$ irriducibile e stocastica per colonna.

Allora l'autovalore dominante è uguale a uno ed è semplice.

Inoltre, il suo unico autovettore associato è l'unico ad avere componenti tutte non negative.

Ecco quindi che, grazie a questo teorema, il problema agli autovalori ha sempre un'unica soluzione. D'altra parte, la matrice P' non è a priori irriducibile. Per garantire anche questa proprietà, dobbiamo introdurre una nuova modifica della matrice. Definiamo:

$$E = \mathbf{1} \cdot \bar{v}^T$$

$$P'' = cP' + (1 - c)E \quad c \in (0, 1) \quad (1.5)$$

La matrice P'' così definita è irriducibile.

Osservazione 4. Con la modifica in (1.5) stiamo scalando la matrice P' del valore $c \in (0, 1)$ e aggiungiamo $\frac{1-c}{n}$ a ogni termine di P' . Cioè introduciamo la possibilità che un utente salti da una pagina a qualsiasi altra con una

probabilità di $(1 - c)$. Questa probabilità è detta *teletrasporto*.

Chiaramente, se c è molto vicino a 1, la modifica del problema originario è limitata. Sembrerebbe quindi intuitivo voler tenere c molto vicino a uno per dare più rilevanza alla struttura Web originaria.

Teorema 1.2.2. *Sia $\text{spec}(H) = \{1, \lambda_2, \dots, \lambda_n\}$. Allora:*

$$\text{spec}(P'') = \{1, c\lambda_2, \dots, c\lambda_n\}$$

Dimostrazione. Sia $\mathbf{1}^* = \frac{1}{\sqrt{n}}\mathbf{1}$.

Allora $(P'')^T \mathbf{1}^* = cH^T \mathbf{1}^* + (1 - c)\frac{1}{n}\mathbf{1}\mathbf{1}^T \mathbf{1}^* = \mathbf{1}^*$.

Sia ora $U = [\mathbf{1}^*, U_0]$ unitaria, allora $U^* H U = \begin{pmatrix} 1 & 0 \\ * & T \end{pmatrix}$.

Per cui: $U^* P U = \alpha \begin{pmatrix} 1 & 0 \\ * & T \end{pmatrix} + (1 - \alpha)\frac{1}{n}n e_1 \mathbf{1}^{*T} U = \begin{pmatrix} 1 & 0 \\ * & \alpha T \end{pmatrix} \quad \square$

Quindi, se anche H avesse più di un autovalore 1, P'' ne avrebbe comunque soltanto uno per $c < 1$, perché $c \cdot 1 < 1$.

Perciò il vettore PageRank è l'autovettore di P'' associato all'autovalore 1. Ponendo $A = (P'')^T$, il vettore PageRank diventa l'autovettore destro di A associato sempre all'autovalore 1.

Avendo quindi garantito l'esistenza e l'unicità del vettore PageRank, rimane da risolvere il problema agli autovalori:

$$\mathcal{I} = A\mathcal{I}. \quad (1.6)$$

1.3 Il grafo del Web

Analizziamo ora lo stesso problema dal punto di vista dei grafi.

Sia quindi G il grafo diretto del Web, dove i nodi sono rappresentati dalle diverse pagine Web e i cammini orientati come gli inlinks e gli outlinks.

L'importanza di ciascun nodo è data dalla probabilità che un navigatore, al tempo $k > K$, si trovi sul nodo P_i .

Si consideri ora il cammino casuale di un navigatore sul grafo G , dove la probabilità di passare da un nodo all'altro è quella uniforme secondo i cammini uscenti. Esso induce una catena di Markov.

Definizione 1.5. Un processo stocastico discreto si dice di **Markov** se:

$$P(X_{n+1} = i_{n+1} | X_0 = i_0, \dots, X_n = i_n) = P(X_{n+1} = i_{n+1} | X_n = i_n)$$

Ovvero lo stato successivo del sistema dipende soltanto dallo stato immediatamente precedente.

A questa catena di Markov è quindi possibile associare una matrice di transizione $P = (p_{ij})$ dove $p_{ij} = 1/\text{deg}(i)$ con $\text{deg}(i)$ il numero di cammini uscenti dal nodo P_i .

Definizione 1.6. Una **matrice di transizione** per una catena di Markov è la matrice generata dalla probabilità di transizione in k passi.

È chiaro che la matrice P così definita coincide con la matrice di Hyperlink H . Ancora una volta, P risulta effettivamente di transizione soltanto se ogni nodo del grafo ha almeno un cammino uscente.

Possiamo quindi definire la matrice P' come in (1.4) per garantire questa proprietà.

Da questo punto di vista, determinare il vettore PageRank, ovvero determinare l'importanza di ciascuna pagina Web, consiste nel calcolare la distribuzione di probabilità associata alla catena di Markov.

Vedremo in seguito come questa distribuzione, sotto opportune ipotesi, risulta essere unica.

Analizziamo ora l'analogo per i grafi delle matrici irriducibili.

Definizione 1.7. Un grafo diretto si dice **fortemente connesso** se per ogni coppia di nodi (P_i, P_j) esiste un cammino che porta l'uno nell'altro.

Ecco quindi che grafi fortemente connessi e matrici irriducibili sono legati dal seguente teorema.

Teorema 1.3.1. *Una matrice A è irriducibile se e solo se il suo grafo diretto è fortemente connesso.*

Presentiamo infine il rispettivo Teorema di Perron-Frobenius.

Teorema 1.3.2. Teorema ergodico per catene di Markov.

La catena di Markov definita dalla matrice P' ha un'unica distribuzione di probabilità stazionaria se P' è aperiodica e irriducibile.

Quindi, il vettore PageRank associato alla matrice di transizione P' è unico soltanto se essa è irriducibile.

Ancora, la modifica in P'' , presentata in precedenza, soddisfa le ipotesi e garantisce l'unicità della soluzione.

Definendo nuovamente $A = (P'')^T$, rimane da risolvere il problema agli autovalori $\mathcal{I} = A\mathcal{I}$.

Capitolo 2

Il metodo delle potenze

2.1 Scelta del metodo

Il metodo delle potenze, nonostante sia il più vecchio per il calcolo degli autovettori di una matrice, risulta il più adatto per il problema in questione. Tra le motivazioni che ci spingono ad adottare questo metodo, figura sicuramente la dimensione stessa del problema; la matrice di Hyperlink ha dimensioni $n \times n$, dove n è il numero di siti Web presenti in rete.

Si stima che in Internet ci siano più di 8 miliardi di siti.¹

Metodi diretti per il calcolo degli autovettori, come ad esempio la fattorizzazione QR , falliscono per problemi di dimensioni così elevate.

La motivazione principale che ci spinge a utilizzare il metodo delle potenze e non metodi più recenti, come il metodo di Lanczos, riguarda la memoria utilizzata. Volendo calcolare soltanto l'autovettore dominante, di cui conosciamo peraltro l'autovalore associato, il metodo delle potenze richiede la memorizzazione di un solo vettore, quello che convergerà alla soluzione. Chiaramente, il vettore in questione avrà dimensioni $n \times 1$. Avendo il problema dimensioni enormi, memorizzare più vettori richiederebbe un utilizzo esagerato della memoria. Ecco quindi che il metodo delle potenze risulta il più adatto per il calcolo del vettore PageRank.

¹Secondo il sito <http://www.worldwidewebsize.com>

2.2 Derivazione dell'algoritmo

Il metodo delle potenze è un metodo iterativo per il calcolo dell'autovalore dominante di una matrice A e del relativo autovettore associato.

Avremo quindi bisogno di una iterata iniziale $x^{(0)} \in \mathbb{R}^n$ tale che $\|x^{(0)}\|_2 = 1$. Se supponiamo $A \in \mathbb{R}^{n \times n}$ diagonalizzabile, i suoi autovettori u_1, \dots, u_n costituiscono una base per $\mathbb{R}^{n \times n}$, quindi $x^{(0)}$ può essere scritto come combinazione lineare di questi autovettori:

$$x^{(0)} = \sum_{i=1}^n \alpha_i u_i$$

Definiamo poi l'iterata successiva come:

$$x^{(k+1)} = Ax^{(k)} \quad (2.1)$$

In questo modo possiamo definire:

$$\lambda^{(k+1)} = (x^{(k+1)})^* Ax^{(k+1)} \quad (2.2)$$

Il metodo delle potenze determina quindi due successioni di iterate:

$$\{x^{(k)}\}_{k \in \mathbb{N}}, \quad \{\lambda^{(k)}\}_{k \in \mathbb{N}}$$

tali che, per $|\lambda_1|$ semplice,

$$x^{(k)} \xrightarrow{k \rightarrow \infty} x_1, \quad \lambda^{(k)} \xrightarrow{k \rightarrow \infty} \lambda_1$$

con x_1 autovettore associato a λ_1 .

Verifichiamo che le due iterate convergano effettivamente a x_1 e λ_1 .

Dalla definizione (2.1) otteniamo che:

$$x^{(k)} = Ax^{(k-1)} = A^k x^{(0)}$$

che possiamo anche scrivere come:

$$\frac{x^{(k)}}{\lambda_1^k} = \frac{A^k x^{(0)}}{\lambda_1^k}$$

Avendo supposto A diagonalizzabile, $A = X\Lambda X^{-1}$, da cui

$$A^k = (X\Lambda X^{-1})^k = X\Lambda^k X^{-1} \quad \text{dove } \Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} \quad (2.3)$$

Allora

$$\begin{aligned} A^k &= \frac{X\Lambda^k X^{-1}}{\lambda_1^k} x^{(0)} = \frac{1}{\lambda_1^k} \begin{pmatrix} x_1 & \dots & x_n \end{pmatrix} \begin{pmatrix} \lambda_1^k & & \\ & \ddots & \\ & & \lambda_n^k \end{pmatrix} \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix} = \\ &= \frac{1}{\lambda_1^k} \begin{pmatrix} x_1 & \dots & x_n \end{pmatrix} \begin{pmatrix} \lambda_1^k z_1 \\ \vdots \\ \lambda_n^k z_n \end{pmatrix} = \frac{1}{\lambda_1^k} \sum_{j=1}^n x_j \lambda_j^k z_j = \\ &= \frac{1}{\lambda_1^k} x_1 \lambda_1^k z_1 + \sum_{j=2}^n x_j \frac{\lambda_j^k}{\lambda_1^k} z_j \xrightarrow{k \rightarrow \infty} x_1 z_1 \quad (2.4) \end{aligned}$$

poiché $\frac{\lambda_j^k}{\lambda_1^k} \xrightarrow{k \rightarrow \infty} 0$.

Teorema 2.2.1. *Sia $A \in \mathbb{R}^{n \times n}$ e $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$ i suoi autovalori ordinati. Allora $\exists c > 0$ tale che:*

$$\|x^{(k)} - x_1\|_2 \leq c \left| \frac{\lambda_2}{\lambda_1} \right|^k \quad (2.5)$$

2.3 Il metodo delle potenze per il vettore PageRank

Vediamo ora come si applica il metodo delle potenze per il calcolo del vettore PageRank.

Consideriamo la matrice A definita nel capitolo 1. Prendendo come iterata iniziale $x^{(0)} = \mathbf{1}$, si hanno già ipotesi sufficienti per garantire la convergenza. Supponiamo inoltre, per semplicità, che $x^{(0)}$ appartenga al sottospazio

generato dagli autovettori di A . Allora possiamo scrivere:

$$x^{(0)} = u_1 + \alpha_2 u_2 + \cdots + \alpha_n u_n \quad (2.6)$$

Siccome $\lambda_1 = 1$, otteniamo che l'iterata successiva $x^{(1)}$:

$$x^{(1)} = Ax^{(0)} = \lambda_1 u_1 + \alpha_2 \lambda_2 u_2 + \cdots + \alpha_n \lambda_n u_n = u_1 + \alpha_2 \lambda_2 u_2 + \cdots + \alpha_n \lambda_n u_n \quad (2.7)$$

Da cui la generica iterata k -esima:

$$x^{(k)} = Ax^{(k-1)} = A^k x^{(0)} = u_1 + \alpha_2 \lambda_2^k u_2 + \cdots + \alpha_n \lambda_n^k u_n \quad (2.8)$$

Poiché 1 è l'autovalore dominante, $|\lambda_n| \leq \cdots \leq |\lambda_2| < 1$ da cui:

$$x^{(k)} \xrightarrow{k \rightarrow \infty} u_1 \quad (2.9)$$

Cioè il metodo delle potenze converge al vettore PageRank per $k \rightarrow \infty$.

2.4 Algoritmo

Possiamo implementare quanto detto sopra in un algoritmo.

Algoritmo 1

```

function  $x^{(n)} = \text{MetodoPotenze}(kmax, \epsilon, v)$ 
 $x^{(0)} = v;$ 
 $k = 1, \delta = 1;$ 
while  $\delta > \epsilon$  and  $k < kmax$ 
   $x^{(k)} = Ax^{(k-1)};$ 
   $\delta = \|x^{(k)} - x^{(k-1)}\|_1;$ 
   $k = k + 1;$ 
end
end

```

Osservazione 5. Stiamo utilizzando la norma-1 per economicità.

Infatti, siccome A è stocastica per colonna, le iterate $x^{(k)}$ hanno norma unitaria e non sono necessari calcoli per ottenere la norma-1.

Osservazione 6. Il residuo così definito è, in realtà, il classico residuo:

$$\|Ax^{(k)} - x^{(k-1)}\|.$$

Infatti, sostituendo a $x^{(k)}$ la definizione data in (2.1), si ritrova la formula usuale. Stiamo ancora una volta limitando il costo computazionale.

Osservazione 7. In una eventuale implementazione del metodo, sarebbe opportuno normalizzare le iterate. Questo avviene perché, nonostante in aritmetica esatta le iterate abbiano effettivamente norma unitaria, in aritmetica finita questo potrebbe non verificarsi.

2.5 Costo computazionale

Il costo computazionale di una singola iterazione del metodo delle potenze consiste nel prodotto matrice-vettore: $Ax^{(k)}$.

Generalmente, questo prodotto ha un costo di $\mathcal{O}(n^2)$, ma nel nostro caso possiamo sfruttare la definizione di A per limitarlo.

Ricordiamo che A era stata definita come segue:

$$A = cP' + (1 - c)E$$

dove P' è una matrice estremamente sparsa. Vediamo quindi come, sfruttando la sparsità della matrice P' , si può implementare opportunamente il prodotto per ridurre il costo computazionale.

Algoritmo 2

$$y = cP^T x;$$

$$\omega = \|x\|_1 - \|y\|_1$$

$$y = y + \omega v$$

dove v è il vettore di personalizzazione.

Usando l'algoritmo 2 si riesce a ridurre il costo computazionale del prodotto matrice-vettore $Ax^{(k)}$ a $\mathcal{O}(n)$.

2.6 Velocità di convergenza

La velocità di convergenza del metodo delle potenze è dato da:

$$\frac{|\lambda_2(A)|}{|\lambda_1(A)|}$$

Osservazione 8. Nel nostro caso, però, abbiamo che:

$$\frac{|\lambda_2(A)|}{|\lambda_1(A)|} = |\lambda_2(A)| \leq c$$

Per questo motivo, valori piccoli di c assicurano una buona velocità di convergenza, mentre valori grandi di c danno più importanza alla struttura Web. Ecco che abbiamo trovato delle motivazioni per cui si potrebbe preferire tenere c basso. Si tratterà di trovare il giusto compromesso tra c grande e piccolo.

Osservazione 9. Il valore scelto originariamente da *Google* era $c = 0.85$. Per $\epsilon = 10^{-8}$ si ha infatti che:

$$|\lambda|^k \leq c^k \leq \epsilon \quad \text{per} \quad k > \frac{\log \epsilon}{\log c} \approx 113$$

Questo valore di c è un buon compromesso perché non modifica troppo la matrice iniziale, tuttavia assicura già una buona velocità di convergenza.

Osservazione 10. L'intervallo $1 - |\lambda_2|$ corrisponde alla probabilità di teletrasporto $1 - c$.

Perciò, se la probabilità di teletrasporto è grande e il vettore di personalizzazione v è uniforme, il metodo delle potenze funziona bene.

In questo caso, però, l'effetto dello *spam* è maggiore, stiamo infatti aggiungendo l'importanza $1 - c$ a tutte le pagine Web, anche a quelle che sarebbero normalmente poco importanti. Ne segue che una pagina puntata da molte pagine di poca importanza, proprio per via di questa aggiunta, guadagnerà un'importanza ingiustificatamente alta.

Nel caso limite in cui $1 - c = 1$ si ha che l'importanza di ogni pagina Web è la stessa.

2.7 Dati campione

Nei risultati sperimentali che andremo ad analizzare, saranno impiegate tre diverse matrici come campione della matrice A .

Queste tre matrici si differenziano per le dimensioni, cioè il numero di pagine Web prese in considerazione, e per la sparsità, nonostante siano chiaramente tutte molto sparse.

Matrice:	Dimensioni	% non sparsità
“Stanford”	685,000 × 685,000	$2 \cdot 10^{-5}\%$
“Google”	916,000 × 916,000	$3 \cdot 10^{-6}\%$
“LiveJournal”	$\approx 5M \times 5M$	$6 \cdot 10^{-6}\%$

Tabella 2.1: Dataset utilizzato.

La matrice “Stanford” ha le dimensioni più ridotte, ma anche la meno sparsa. Vedremo quanto la sparsità influenzerà il tempo di convergenza.

La matrice “Google” è quella mediana, sia come sparsità che come dimensioni.

La matrice “LiveJournal”, infine, ha le dimensioni maggiori rispetto alle altre, ed è anche la più sparsa.

Il residuo utilizzato sarà 10^{-10} e il numero di iterazioni massime 1200.

2.8 Risultati sperimentali

Esaminiamo ora i risultati sperimentali ottenuti dal metodo delle potenze sulle matrici presentate nella sezione precedente.

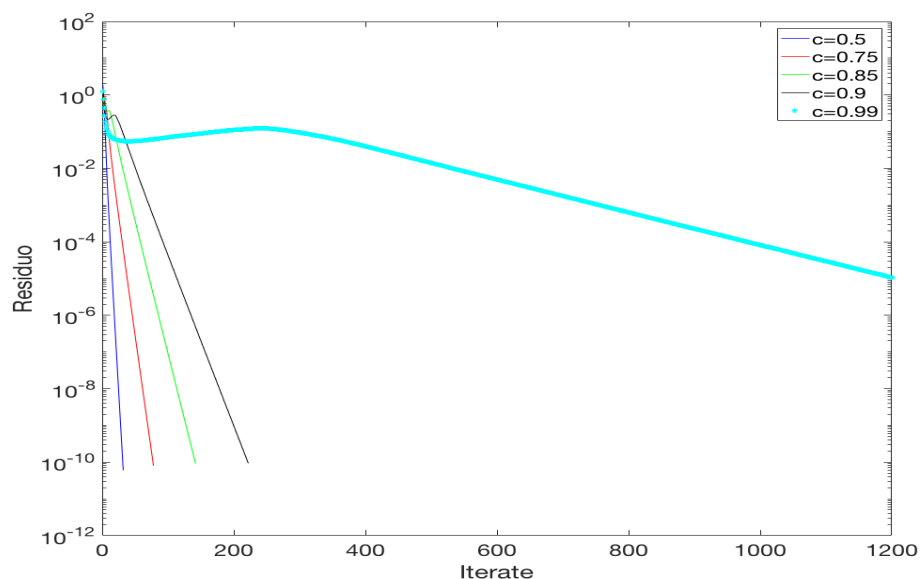


Figura 2.1: Velocità di convergenza del metodo delle potenze per diversi valori di c .

La figura 2.1 mostra la velocità di convergenza del metodo delle potenze. Notiamo come, al crescere di c , cresce anche il numero di iterate richieste per la convergenza.

Sull'asse delle ascisse è riportata l'iterata corrente, mentre sull'asse delle ordinate la grandezza del residuo. Questi risultati sono ottenuti applicando il metodo delle potenze alla matrice "Stanford" per diversi valori di c .

Se non diversamente specificato, i risultati esposti in questo elaborato seguiranno la stessa traccia della figura 2.1.

Un tentativo per migliorare il metodo delle potenze potrebbe essere quello di aggiornare il valore di c al crescere delle iterate. Per farlo, possiamo seguire due strade.

Partendo da un valore di c basso, e aumentandolo, si sfrutta l'iniziale velocità di convergenza, per poi accrescere la precisione della ricerca nel corso delle iterate.

Partendo da c alto, invece, si dà subito importanza alla struttura del Web, per poi aumentare la velocità di convergenza al crescere delle iterate.

2.8.1 Il metodo delle potenze incrementando c

Applichiamo quindi il metodo delle potenze alla matrice “Stanford” con i seguenti valori di c :

- $c_1, \dots, c_3 = 0.5,$
- $c_4, \dots, c_{10} = 0.75,$
- $c_{11}, \dots, c_{20} = 0.85,$
- $c_{20}, \dots, c_{40} = 0.9,$
- $c_{90}, \dots, c_{1200} = 0.99.$

dove c_i rappresenta il valore di c all'iterata i -esima.

La figura 2.2 mostra come ogni cambiamento del valore di c provochi una brusca impennata nel residuo. Ciò va a vanificare gran parte dell'accelerazione ottenuta inizialmente, rendendo quasi inutile questo approccio. Sebbene il residuo finale sia sensibilmente più basso di quello ottenuto mantenendo $c = 0.99$ costante, i risultati ottenuti non sono minimamente soddisfacenti, poiché la tolleranza non è stata raggiunta nel numero massimo di iterate.

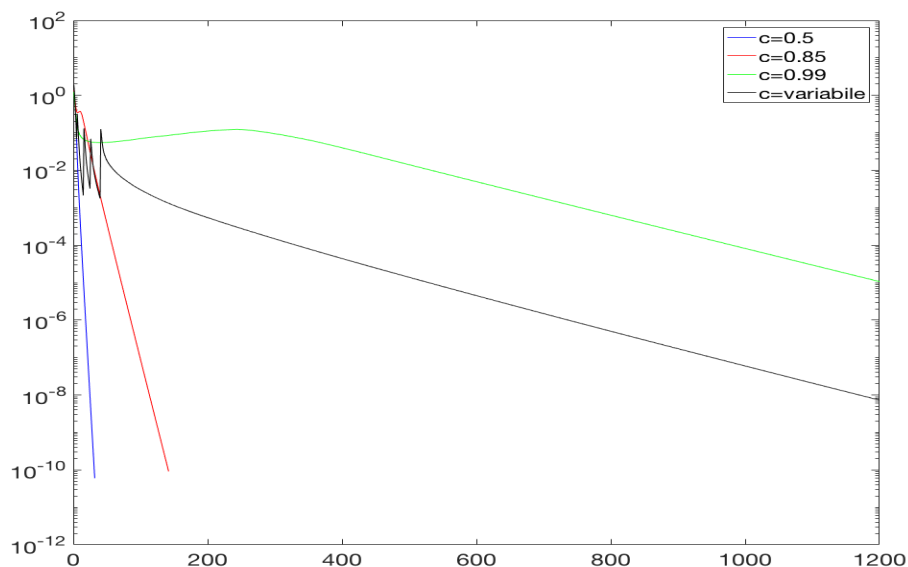


Figura 2.2: Residuo del metodo delle potenze aumentando c .

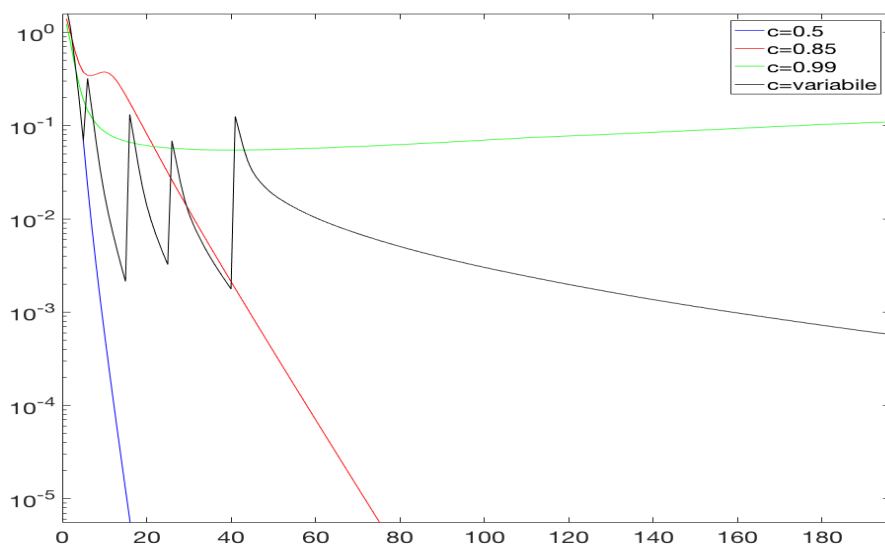


Figura 2.3: Zoom sul gradino nel residuo all'aggiornamento di c .

La figura 2.3 mostra più nel dettaglio il gradino che si forma a ogni aggiornamento di c . Essendo così grande, i benefici ottenuti sono nulli: il valore del residuo alla terza iterata è praticamente lo stesso di quello alla quarantesima.

2.8.2 Il metodo delle potenze riducendo c

Applichiamo ora il metodo delle potenze con i seguenti valori di c :

- $c_1, \dots, c_{200} = 0.99$,
- $c_{201}, \dots, c_{225} = 0.9$,
- $c_{226}, \dots, c_{235} = 0.85$,
- $c_{246}, \dots, c_{250} = 0.75$,
- $c_{251}, \dots, c_{1200} = 0.5$.

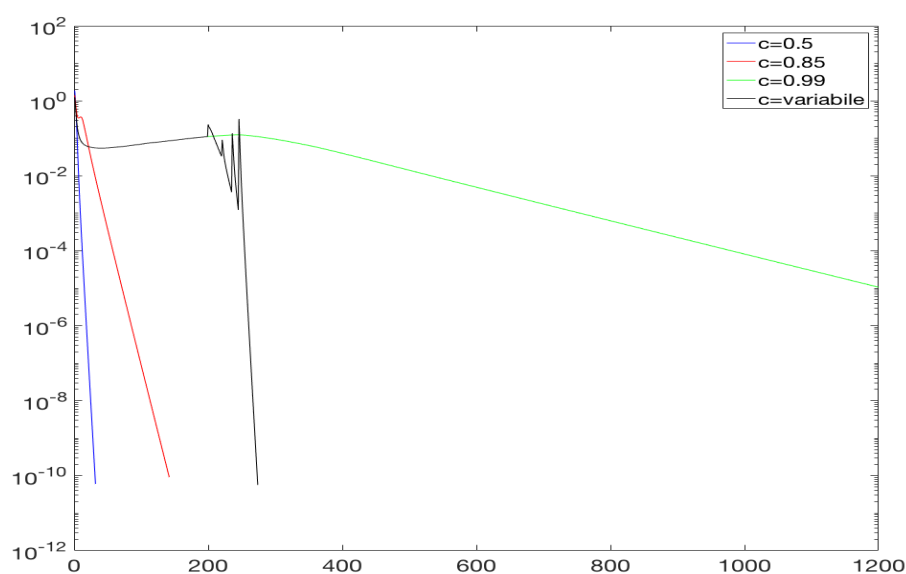


Figura 2.4: Residuo del metodo delle potenze riducendo c nel corso delle iterate.

La figura 2.4 mostra come i risultati ottenuti diminuendo c nel corso delle iterate provochino effetti altrettanto insoddisfacenti. Ogni volta che c viene aggiornato, il residuo subisce un significativo aumento. In questo caso, però, l'accelerazione successiva riesce a compensarlo.

In particolare notiamo come il picco del residuo sia in relazione al valore di c . Sappiamo infatti che se c è alto stiamo modificando di poco la matrice

originaria. L'approssimazione sarà perciò migliore e il residuo ottenuto più limitato. I primi gradini che si formano sono pertanto anche i più ridotti.

Diminuendo invece il valore di c , ci discostiamo molto dal problema originario e, a fronte di una maggiore velocità di convergenza, si presenta un gradino molto più grande, proprio per via dell'approssimazione grossolana che stiamo compiendo. I gradini finali sono perciò i più marcati.

Anche questa strategia di miglioramento non è granché soddisfacente, poiché abbiamo perso molta della precisione iniziale, nonostante il numero di iterate richieste sia ugualmente molto alto.

2.8.3 Conclusioni

I risultati sperimentali mostrati hanno messo in evidenza i limiti del metodo delle potenze. Sebbene esso sia valido per calcolare il vettore PageRank per valori di c limitati, la sua utilità si riduce drasticamente al crescere di questi valori.

Una prima idea per migliorare il metodo, cioè quella di aumentare o ridurre il valore di c nel corso delle iterate, non si è rivelata soddisfacente, poiché il gradino nel residuo che si viene a formare ogni volta che c viene aggiornato fa perdere tutti i benefici, ovvero la velocità di convergenza nel primo caso, la precisione della ricerca nel secondo.

È in questo contesto che si introducono i metodi estrapolativi che, in appoggio al metodo delle potenze, sono volti ad accelerare il calcolo del vettore PageRank, soprattutto per quei valori di c per cui l'algoritmo originario fallisce.

Capitolo 3

Estrapolazione di Aitken

3.1 Il metodo di estrapolazione di Aitken

L'extrapolazione di Aitken è il primo metodo di estrapolazione presentato per accelerare il calcolo del vettore PageRank.

L'idea alla base¹ è che l'iterata $x^{(k-2)}$ possa essere espressa come combinazione lineare dei primi due autovettori di A . Partendo da questa assunzione, si riuscirà a risolvere u_1 in forma chiusa. Chiaramente stiamo compiendo un'approssimazione, ma vedremo che l'accuratezza aumenterà al crescere di n .

Assumiamo quindi che $x^{(k-2)}$ si possa scrivere come:

$$x^{(k-2)} = u_1 + \alpha_2 u_2 \quad (3.1)$$

Siccome $\lambda_1 = 1$, possiamo scrivere le due successive iterate come:

$$x^{(k-1)} = Ax^{(k-2)} = u_1 + \alpha_2 \lambda_2 u_2 \quad (3.2)$$

$$x^{(k)} = Ax^{(k-1)} = u_1 + \alpha_2 \lambda_2^2 u_2 \quad (3.3)$$

¹S. D. Kamvar, T.H. Haveliwala, C. D. Manning, G. H. Golub, *Extrapolation methods for accelerating PageRank computations*, Proceedings of the twelfth international World Wide Web conference, May 2003.

Definiamo adesso i vettori g e h mediante le loro componenti²:

$$g_i := (x_i^{(k-1)} - x_i^{(k-2)})^2 \quad i = 1, \dots, n \quad (3.4)$$

$$h_i := x_i^{(k)} - 2x_i^{(k-1)} + x_i^{(k-2)} \quad i = 1, \dots, n \quad (3.5)$$

Per cui, usando le Definizioni (3.2) e (3.3):

$$g_i = \alpha_2^2 (\lambda_2 - 1)^2 (u_2)_i^2, \quad h_i = \alpha_2 (\lambda_2 - 1)^2 (u_2)_i$$

Definiamo infine il vettore $f := \alpha_2 u_2$, cioè:

$$f_i := \frac{g_i}{h_i} = \frac{\alpha_2^2 (\lambda_2 - 1)^2 (u_2)_i^2}{\alpha_2 (\lambda_2 - 1)^2 (u_2)_i} = \alpha_2 (u_2)_i \quad i = 1, \dots, n \quad (3.6)$$

Ecco quindi che da (3.1) riusciamo ad ottenere una soluzione in forma chiusa per u_1 :

$$u_1 = x^{(k-2)} - \alpha_2 u_2 = x^{(k-2)} - f \quad (3.7)$$

L'estrapolazione di Aitken così definita può essere dunque introdotta nel metodo delle potenze per accelerare la velocità di convergenza.

3.2 Algoritmi

Mostriamo ora una possibile implementazione del metodo esposto sopra negli algoritmi 3 e 4.

Algoritmo 3

```
function x = Aitken(x(k-2), x(k-1), x(k))
g = (x(k-1) - x(k-2)).2;
h = (x(k) - 2x(k-1) + x(k-2));
x = x(k-2) - g./h;
end
```

²C. Brezinski, M. Redivo-Zaglia, *The PageRank vector: properties, computation, approximation and acceleration*, SIMA 2006

L'Algoritmo 3 calcola l'iterata successiva mediante estrapolazione di Aitken. La procedura va introdotta all'interno del metodo delle potenze standard per ottenere un'accelerazione della convergenza.

Algoritmo 4

```

function  $x^{(n)} = \text{AitkenPotenze}(kmax, \epsilon, v)$ 
 $x^{(0)} = v;$ 
 $k = 1, \delta = 1;$ 
while  $\delta > \epsilon$  and  $k < kmax$ 
   $x^{(k)} = Ax^{(k-1)};$ 
   $\delta = \|x^{(k)} - x^{(k-1)}\|_1;$ 
  All'iterata  $l$ :  $x^{(l)} = \text{Aitken}(x^{(l-2)}, x^{(l-1)}, x^{(l)})$ 
   $k = k + 1;$ 
end
end

```

L'Algoritmo 4 mostra come viene modificato il metodo delle potenze standard per ottenere i benefici dell'extrapolazione di Aitken. L'extrapolazione avviene all'iterata l -esima.

Vedremo nei risultati sperimentali i benefici di scegliere l grande o piccolo.

3.3 Costo computazionale

Affinché il metodo di Aitken sia effettivamente utile, il costo computazionale aggiuntivo al metodo delle potenze standard deve essere minimo.

Osservando gli Algoritmi 3 e 4, è chiaro che le operazioni introdotte hanno tutte un costo di $\mathcal{O}(n)$, cioè dello stesso ordine di una singola iterazione del metodo delle potenze standard. Possiamo quindi affermare che stiamo aggiungendo un costo computazionale minimo.

3.4 Risultati sperimentali

Esponiamo ora i risultati che si ottengono introducendo l'estrapolazione di Aitken all'interno del metodo delle potenze.

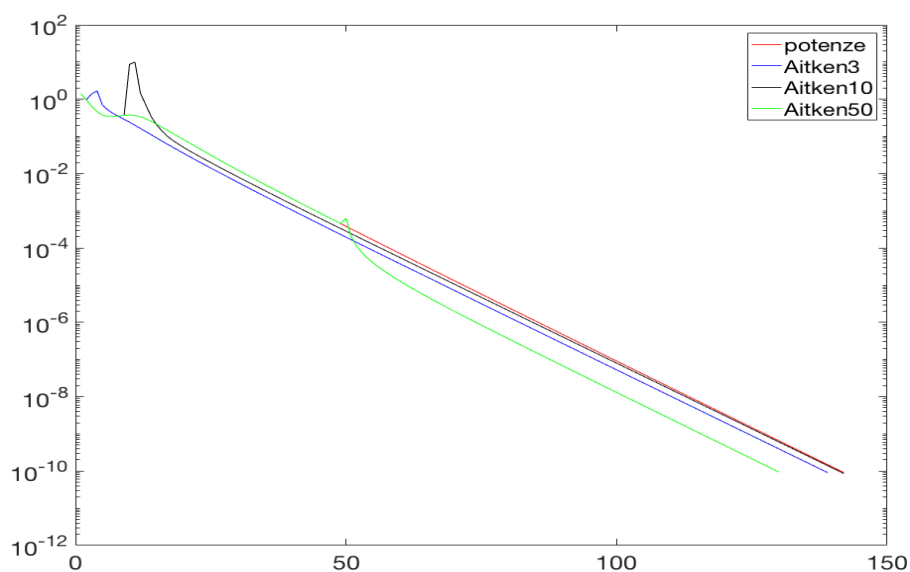


Figura 3.1: Confronto del metodo delle potenze standard e accelerato con Aitken per $c = 0.85$.

La figura 3.1 confronta le differenze nel residuo che si riscontrano nel metodo delle potenze standard e accelerato con Aitken. L'estrapolazione avviene a tre diverse iterate: la terza, la decima e la cinquantesima. Notiamo come estrapolare più tardi provochi il risultato migliore. Questo è dovuto all'approssimazione che si sta compiendo. Supporre, infatti, che l'iterata k -esima possa essere espressa come combinazione lineare dei primi due auto-vettori è molto restrittivo per le iterate iniziali; lo è meno se il metodo delle potenze ha già indirizzato l'iterata verso la soluzione. Il gradino nel residuo che si ottiene estrapolando alla cinquantesima iterata è molto inferiore agli altri due, e ciò provoca l'accelerazione più alta.

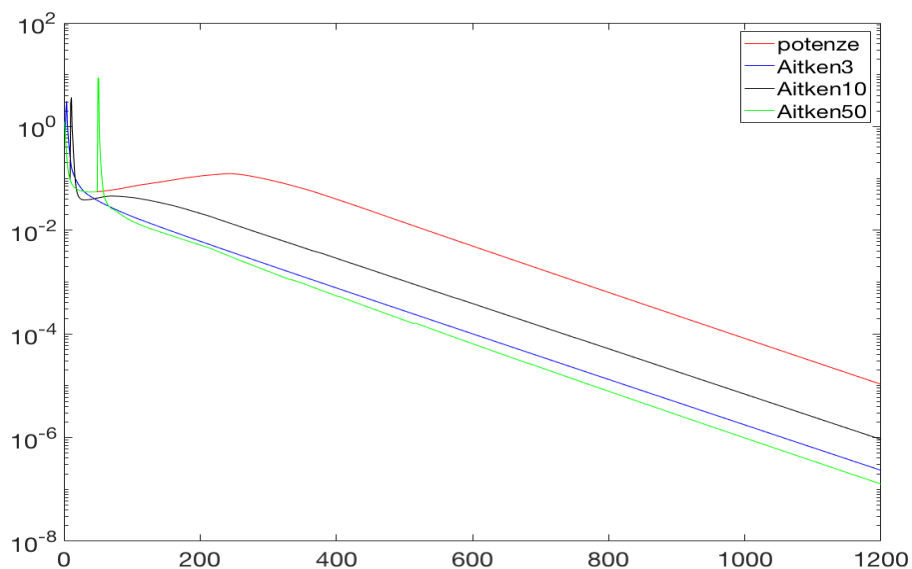


Figura 3.2: Confronto del metodo potenze standard e accelerato con Aitken per $c = 0.99$.

La figura 3.2 mostra gli stessi risultati per un valore di c più alto. In questo modo i benefici ottenuti sono maggiori che per $c = 0.85$.

3.4.1 Estrapolazione periodica di Aitken

Potremmo pensare che eseguire periodicamente l'estrapolazione di Aitken migliori ulteriormente la velocità di convergenza.

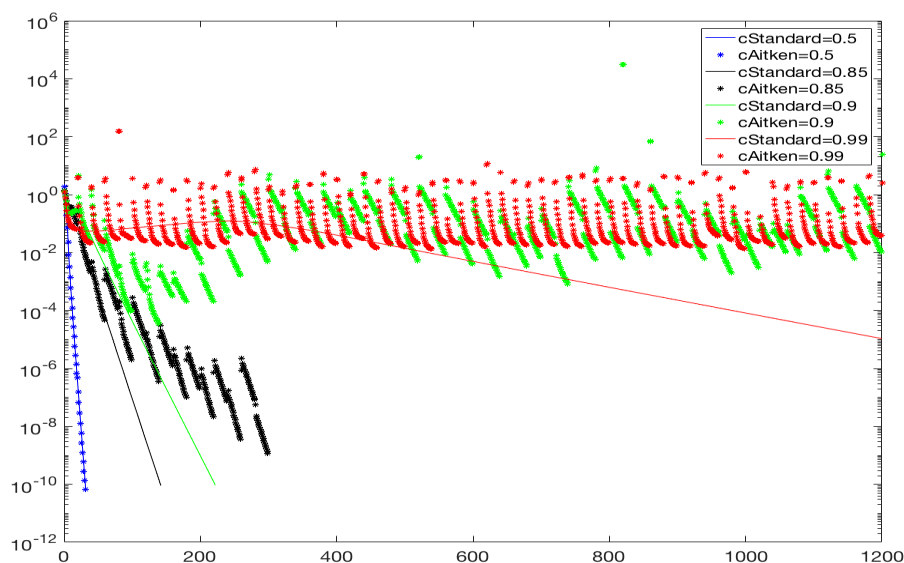


Figura 3.3: Applicazione periodica dell'estrapolazione di Aitken per diversi valori di c , estrapolando ogni 10 iterate.

La figura 3.3 palesa il fatto che l'estrapolazione di Aitken non debba essere applicata periodicamente. Se eseguita ogni 10 iterate i risultati ottenuti sono fallimentari: il metodo spesso non converge e, qualora lo faccia, richiede un numero maggiore di iterate. Il motivo di questo comportamento è da ricercarsi ancora una volta nell'approssimazione che si è compiuta. Essendo l'approssimazione grossolana, il metodo ha bisogno di molte iterate per superare al gradino nel residuo che si forma. Se però questo gradino è presente troppo spesso, l'approssimazione finale è troppo imprecisa per far convergere il metodo.

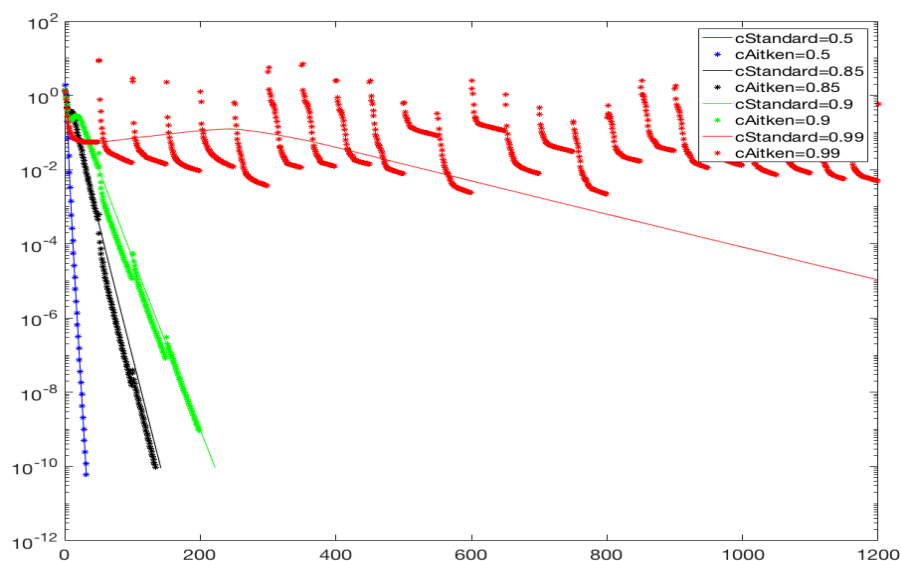


Figura 3.4: Applicazione periodica dell'estrapolazione di Aitken per diversi valori di c , estrapolando ogni 50 iterate.

Se l'estrapolazione è applicata più raramente, i risultati sono migliori, ma ancora non soddisfacenti per giustificare l'utilizzo. Possiamo quindi affermare che l'estrapolazione di Aitken, per migliorare il metodo delle potenze, va applicata solo una volta. Siccome abbiamo già osservato come l'iterata in cui si estrapola influenzi l'accelerazione ottenuta, possiamo investigare quale sia l'iterata ottimale.

3.4.2 Iterata ottimale di estrapolazione

Studiamo ora quanto l'iterata in cui si estrapola influenzi l'accelerazione ottenuta, cercando di determinare quale sia l'iterata ottimale di estrapolazione.

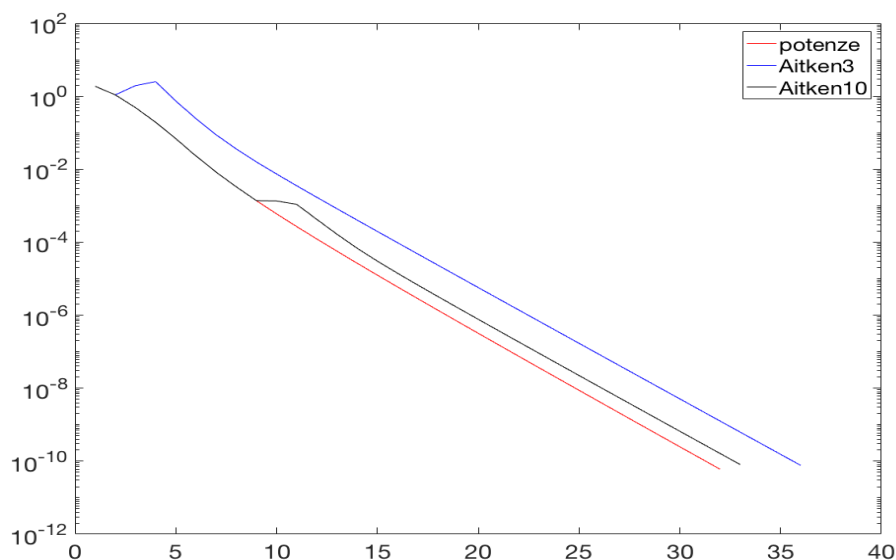


Figura 3.5: Confronto del metodo delle potenze standard e accelerato con Aitken per $c = 0.5$.

La figura 3.5 mostra come, per valori piccoli di c , l'extrapolazione di Aitken provochi un aumento delle iterate richieste per la convergenza. Ciò è dovuto al gradino iniziale. Inoltre può verificarsi che, se il metodo delle potenze converge molto velocemente, esso giunga a convergenza prima dell'extrapolazione, vanificando così l'utilità del metodo stesso.

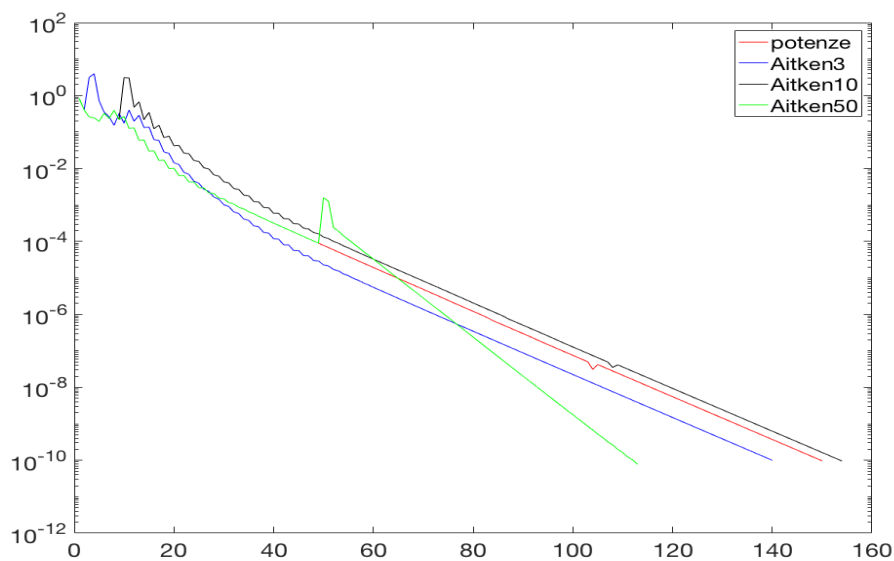


Figura 3.6: Confronto del metodo delle potenze standard e accelerato con Aitken per $c = 0.99$.

Per valori più grandi di c , invece l'estrapolazione di Aitken ottiene ottimi risultati, soprattutto se applicata dopo un numero elevato di iterate. Si tratta, quindi, di applicare l'estrapolazione dopo un numero di iterate sufficientemente grande da permettere al metodo delle potenze di indirizzare l'iterata verso la soluzione, ma non tale da permettere che l'algoritmo originario converga, senza aver ancora applicato l'estrapolazione.

3.4.3 La sparsità della matrice e il residuo

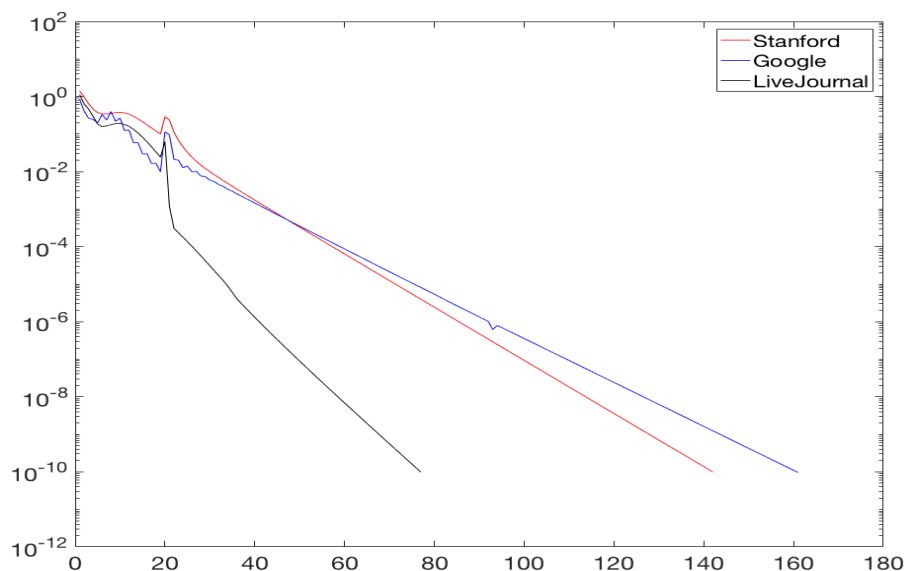


Figura 3.7: Confronto del residuo per tre diverse matrici.

La figura 3.7 confronta il residuo ottenuto con l'estrapolazione di Aitken e $c = 0.85$ per le tre matrici campione, estrapolando alla 30-iterata.

Matrice:	$c = 0.5$	$c = 0.85$	$c = 0.99$
Stanford	0.7117	2.8697	24.8612
Google	9.3898	9.4302	9.5933
LiveJournal	45.8483	46.1404	47.0130

Tabella 3.1: Secondi impiegati per arrivare a convergenza.

La tabella 3.1 riporta il tempo, espresso in secondi, impiegato per ottenere la convergenza col metodo di estrapolazione di Aitken.

Come è normale aspettarsi, una matrice di dimensioni ridotte richiede un tempo basso per arrivare a convergenza. Ciò che invece non ha impatto sul tempo finale è la sparsità della matrice stessa, poiché tutte e tre le matrici hanno circa 7 milioni di elementi non zero; ma il tempo finale è molto differente.

La sparsità della matrice, tuttavia, in relazione al valore di c , modifica il tempo per il calcolo del vettore PageRank. Per matrici molto sparse, infatti, il tempo finale cresce moderatamente al crescere di c ; per la matrice meno sparsa invece c'è, invece, un'enorme incremento.

3.4.4 Conclusioni

I risultati sperimentali riportati hanno mostrato come l'estrapolazione di Aitken sia utile per accelerare il calcolo del vettore PageRank.

Questo metodo, però, non comporta benefici qualora venga applicato periodicamente, per via della ripetuta approssimazione che si compie.

Per ottenere i benefici migliori bisogna estrapolare a una iterata ottimale, che deve essere sufficientemente alta da permettere al metodo delle potenze di aver già indirizzato l'iterata verso il vettore PageRank, ma non troppo elevata, in modo che l'estrapolazione possa essere comunque applicata anche qualora la convergenza sia molto veloce, cioè quando c è piccolo.

In particolare, applicare l'estrapolazione intorno alla trentesima iterata sembra essere un buon compromesso tra le due richieste.

Infine, la sparsità della matrice non influenza la velocità di convergenza del metodo, ma influenza invece il tempo richiesto per la convergenza al variare di c .

Una matrice poco sparsa aumenterà vertiginosamente il tempo impiegato per il calcolo del vettore PageRank al crescere di c , mentre una più sparsa ne sarà poco influenzata.

Il tempo richiesto per la convergenza è inoltre influenzato dalla grandezza della matrice di iterazione.

Questa estrapolazione, tuttavia, perde gran parte della propria efficacia quando $|\lambda_2|$ e $|\lambda_3|$ sono vicini. In questo caso, infatti, l'approssimazione che si compie è troppo grossolana: nel caso limite in cui $\lambda_2 = \overline{\lambda_3}$, l'estrapolazione di Aitken non comporta alcun beneficio.

In ultima analisi, possiamo comunque affermare che il metodo di estrapolazione di Aitken è un buono strumento per accelerare il calcolo del vettore PageRank.

Capitolo 4

Estrapolazione epsilon

4.1 Il metodo di estrapolazione epsilon

L'idea alla base dell'estrapolazione epsilon è la stessa dell'estrapolazione di Aitken, ovvero che la generica iterata $x^{(k-2)}$ possa essere espressa come combinazione lineare dei primi due autovettori di A^1 . Anche in questo caso si otterrà una soluzione in forma chiusa per u_1 .

Procediamo quindi come nell'estrapolazione di Aitken, fino alla definizione di g_i per $i = 1, \dots, n$.

Definiamo, in alternativa ad Aitken:

$$g_i^2 = (x_i^{(k-1)} - x_i^{(k-2)})(x_i^{(k)} - x_i^{(k-1)}) = \alpha_2^2 \lambda_2 (\lambda_2 - 1)^2 (u_2)_i^2 \quad (4.1)$$

Mentre h_i è definita come in (3.5). Allora:

$$f_i = \frac{g_i}{h_i} = \frac{\alpha_2^2 \lambda_2 (\lambda_2 - 1)^2 (u_2)_i^2}{\alpha_2 (\lambda_2 - 1)^2 (u_2)_i} = \alpha_2 \lambda_2 (u_2)_i, \quad (4.2)$$

¹S. D. Kamvar, T.H. Haveliwala, C. D. Manning, G. H. Golub, *Extrapolation methods for accelerating PageRank computations*, Proceedings of the twelfth international World Wide Web conference, May 2003.

²C. Brezinski, M. Redivo-Zaglia, *The PageRank vector: properties, computation, approximation and acceleration*, SIMA 2006

da cui per (3.2):

$$u_1 = x^{(k-1)} - \alpha_2 \lambda_2 u_2 = x^{(k-1)} - f. \quad (4.3)$$

Anche in questo caso, possiamo inserire l'estrapolazione epsilon all'interno del metodo delle potenze per accelerare la convergenza.

4.2 Algoritmi

Algoritmo 5

```
function x = Epsilon(x(k-2), x(k-1), x(k))
g = (x(k-1) - x(k-2)) .* (x(k) - x(k-1));
h = (x(k) - 2x(k-1) + x(k-2));
x = x(k-1) - g./h;
end
```

L'Algoritmo 5 calcola l'iterata successiva mediante estrapolazione epsilon. La procedura va introdotta all'interno del metodo delle potenze standard per ottenere un'accelerazione della convergenza.

Algoritmo 6

```
function x(n) = EpsilonPotenze(kmax, ε, v)
x(0) = v;
k = 1, δ = 1;
while δ > ε and k < kmax
    x(k) = Ax(k-1);
    δ = ||x(k) - x(k-1)||1;
    All'iterata l: x(l) = Epsilon(x(l-2), x(l-1), x(l))
    k = k + 1;
end
end
```


L'Algoritmo 6 mostra come viene modificato il metodo delle potenze standard per ottenere i benefici dell'extrapolazione epsilon. L'extrapolazione avviene all'iterata l -esima.

4.3 Costo computazionale

Il costo computazionale aggiuntivo per l'extrapolazione epsilon è molto simile a quello introdotto dall'extrapolazione di Aitken. L'unica variazione riguarda, appunto, la definizione di g , che in questo caso ha un costo di $3n$, invece che di $2n$. Stiamo quindi aggiungendo un costo computazionale minimo, dello stesso ordine di quello di una singola iterazione del metodo delle potenze, ovvero $\mathcal{O}(n)$.

Vedremo se questa differenza nel costo computazionale, comunque limitata, avrà effetti sulla velocità di convergenza.

4.4 Risultati sperimentali

Esaminiamo ora i risultati che si ottengono introducendo l'extrapolazione epsilon all'interno del metodo delle potenze.

La figura 4.1 mostra come l'extrapolazione epsilon influenzi il residuo in base alla iterata cui è applicato. Per $c = 0.85$ non si ha, però, un aumento della velocità di convergenza, qualsiasi sia l'iterata in cui viene applicata l'extrapolazione. Questo rallentamento della convergenza è dovuto sia all'approssimazione che si compie, ma soprattutto al fatto che non si ottiene quasi accelerazione in seguito all'extrapolazione.

L'extrapolazione avviene rispettivamente alla terza, alla decima o alla cinquantesima iterata. I risultati più apprezzabili si ottengono utilizzando la matrice utilizzata è "Google".

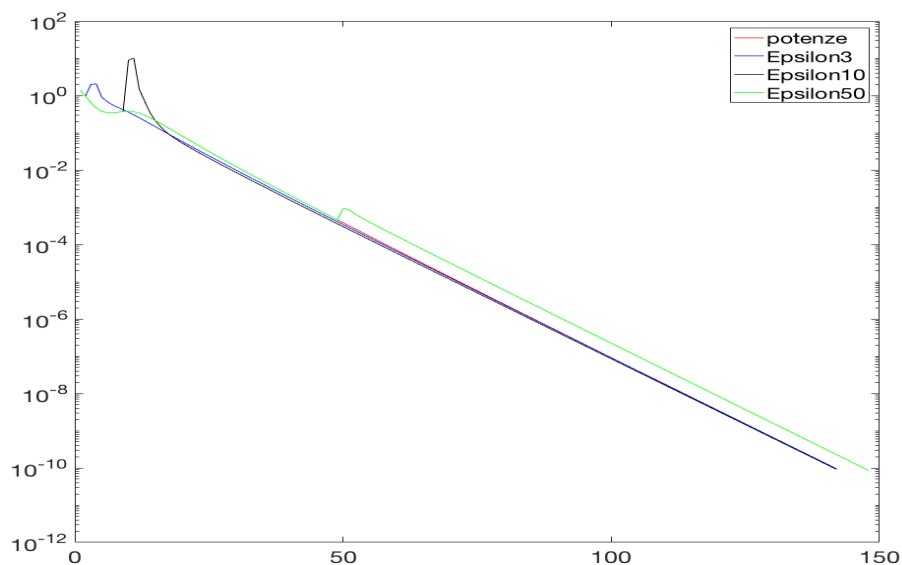


Figura 4.1: Confronto del metodo delle potenze standard e accelerato con epsilon per $c = 0.85$.

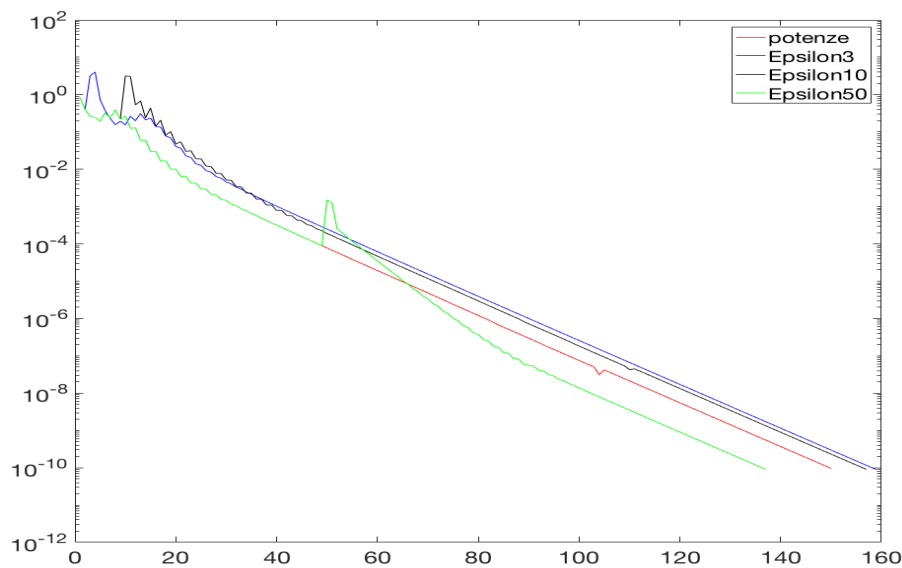


Figura 4.2: Confronto del metodo delle potenze standard e accelerato con epsilon per $c = 0.99$.

La figura 4.2 mette a confronto il metodo delle potenze standard con l'estrapolazione epsilon per un valore maggiore di c . In questo caso abbiamo un'accelerazione soltanto se l'estrapolazione è applicata tardivamente. Anche

se i risultati sono migliori che per $c = 0.85$, ancora non abbiamo un risultato apprezzabile.

4.4.1 Estrapolazione periodica epsilon

Così come per l'estrapolazione di Aitken, potremmo pensare di estrapolare periodicamente per incrementare la velocità di convergenza.

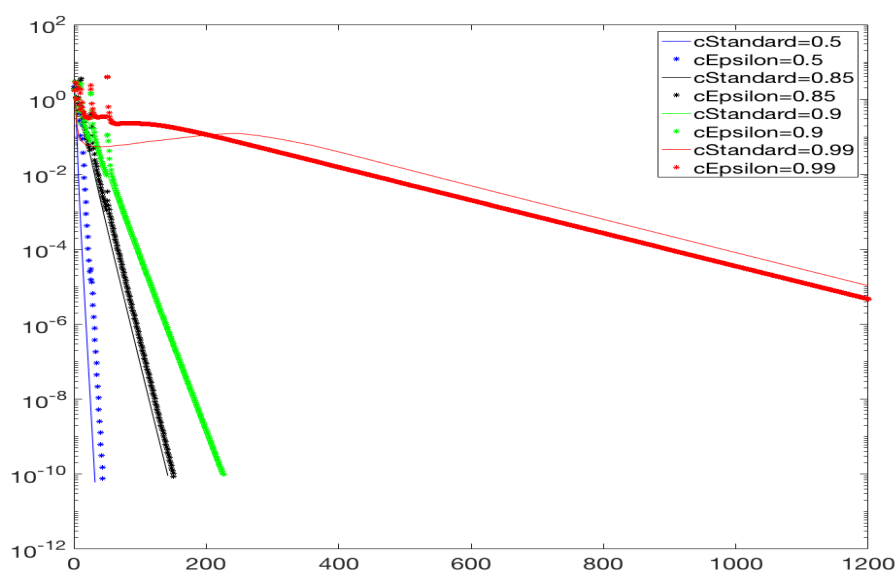


Figura 4.3: Applicazione periodica dell'estrapolazione epsilon per diversi valori di c .

La figura 4.3 mostra il residuo ottenuto estrapolando ogni 50 iterate, per diversi valori di c , sulla matrice “Stanford”. Sebbene i risultati siano decisamente più apprezzabili rispetto all'estrapolazione periodica di Aitken, ancora una volta non lo sono abbastanza da accelerare il metodo delle potenze.

4.4.2 Confronto epsilon e Aitken

Il confronto più interessante è senza dubbio quello tra il residuo ottenuto estrapolando con Aitken e quello con epsilon. Partendo infatti dalla stessa idea, la differenza tra i due metodi risiede nella bontà dell'approssimazione

che si compie e dal costo computazionale aggiuntivo. Il metodo di Aitken richiede un costo computazionale minore, a fronte però di un minor grado di precisione dell'approssimazione. Il metodo di epsilon, al contrario, per un costo computazionale più alto offre un'approssimazione più precisa.

La figura 4.4 confronta il residuo dei tre metodi visti finora, sulla matrice “Stanford”, per $c = 0.5$. Sappiamo che per c basso il metodo delle potenze ottiene già dei buoni risultati.

In questo caso, infatti, entrambi i metodi estrapolativi rallentano l'algoritmo originario, ma il metodo di Aitken è comunque il migliore.

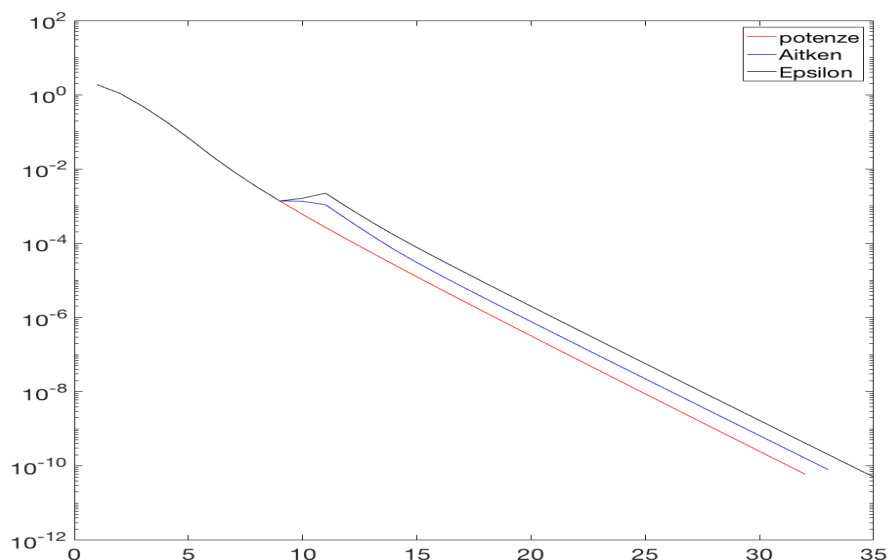


Figura 4.4: Confronto dell'estrapolazione epsilon e di Aitken per $c = 0.5$.

In figura 4.5 sono invece riportati i tre diversi residui per un valore di c più alto. In questo caso sappiamo che il metodo delle potenze non è valido, e qui i due metodi estrapolativi si rivelano molto utili ad accelerare l'algoritmo originario. Anche in questo caso, però, è il metodo di estrapolazione di Aitken a ottenere i risultati migliori.

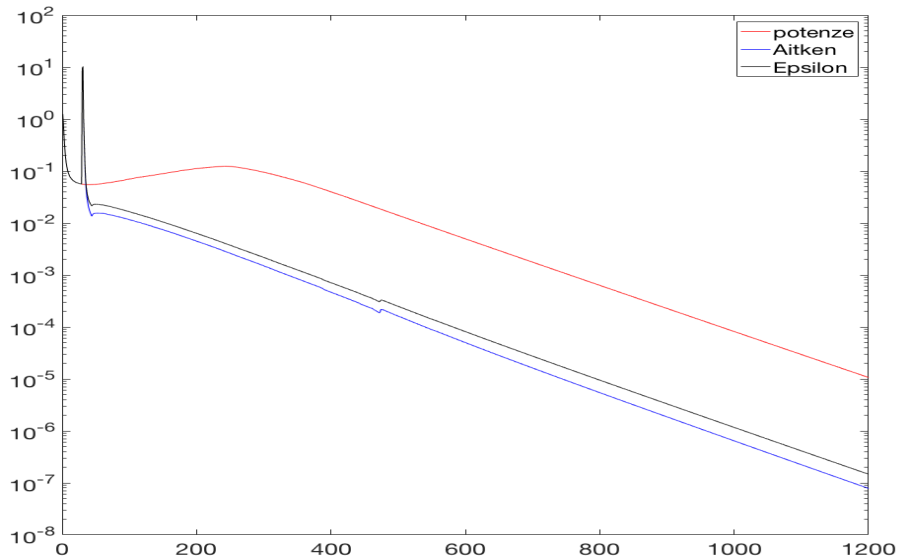


Figura 4.5: Confronto dell'extrapolazione epsilon e di Aitken per $c = 0.99$.

4.4.3 Tempi di convergenza

La bontà di un metodo per calcolare il vettore PageRank si misura con il tempo richiesto per arrivare a convergenza. Alcuni metodi infatti possono risolvere il problema in poche iterate, dal costo computazionale elevato, impiegando più tempo di altri che richiedono molte operazioni poco costose.

Matrice:	$c = 0.5$	$c = 0.85$	$c = 0.99$
Stanford	0.9046	3.1879	24.9016
Google	9.3056	11.5034	11.5766
LiveJournal	48.7432	46.9922	62.4199

Tabella 4.1: Secondi impiegati per la convergenza.

La tabella 4.1 riporta i tempi di convergenza per l'extrapolazione epsilon, per tutte e tre le matrici campione, con i valori di c più significativi. Confrontandola con la tabella 3.1, notiamo come il tempo finale per arrivare a convergenza sia superiore nel caso dell'extrapolazione epsilon; pertanto questo metodo risulta meno indicato per accelerare il calcolo del vettore PageRank.

4.4.4 Conclusioni

Il metodo di estrapolazione epsilon vuole essere un metodo per accelerare il calcolo del vettore PageRank. I suoi risultati sono molto influenzati dalla matrice su cui viene applicato, ma in generale possiamo affermare che per valori alti di c esso comporti discreti benefici, purché non sia applicato troppo presto. Là dove il metodo delle potenze funziona già bene, questo algoritmo rallenta soltanto la convergenza.

L'applicazione periodica dell'extrapolazione epsilon non è indicata a risolvere il problema in questione, poiché provoca la divergenza dell'iterata, o comunque il rallentamento della convergenza. Questo però è l'unico caso in cui i risultati sono più apprezzabili rispetto al metodo di estrapolazione di Aitken. L'iterata ottimale per l'extrapolazione è analoga a quella del metodo di Aitken: intorno alla trentesima iterata risulta essere un buon compromesso per ottenere i migliori benefici.

In ultima analisi, possiamo affermare che il metodo di estrapolazione epsilon è un discreto strumento per migliorare l'algoritmo originario quando esso ottiene i risultati peggiori. Però, essendo meno efficiente rispetto all'extrapolazione di Aitken, esso viene messo in secondo piano.

Capitolo 5

Estrapolazione quadratica

5.1 Il metodo di estrapolazione quadratica

Il metodo di estrapolazione quadratica è l'ultimo metodo per accelerare il metodo delle potenze che andremo ad analizzare.

Questo metodo parte dall'assunzione che la matrice A abbia soltanto tre autovettori e che quindi l'iterata $x^{(k-3)}$ possa essere espressa come combinazione lineare di questi tre autovettori. Anche con questo metodo riusciremo a trovare una soluzione in forma chiusa per u_1 .

Naturalmente, la matrice A ha più di tre autovettori e quella che stiamo compiendo è soltanto un'approssimazione. Come negli altri due metodi però la bontà dell'approssimazione crescerà al crescere di n .

Supponiamo quindi di poter scrivere:

$$x^{(k-3)} = u_1 + \alpha_2 u_2 + \alpha_3 u_3 \quad (5.1)$$

e denotiamo le tre iterate successive con:

$$x^{(k-2)} = Ax^{(k-3)}, \quad x^{(k-1)} = Ax^{(k-2)}, \quad x^{(k)} = Ax^{(k-1)} \quad (5.2)$$

Siccome A ha soltanto tre autovettori, il suo polinomio caratteristico $p_A(\lambda)$ avrà grado tre. Possiamo quindi scrivere:

$$p_A(\lambda) = \gamma_0 + \gamma_1 \lambda + \gamma_2 \lambda^2 + \gamma_3 \lambda^3 \quad (5.3)$$

che vogliamo risolvere per $\gamma_0, \gamma_1, \gamma_2$ e γ_3 .

Sappiamo che $\lambda = 1$ è autovalore, e che gli autovalori della matrice sono anche radici del suo polinomio caratteristico. Quindi ricaviamo che

$$p_A(1) = 0 = \gamma_0 + \gamma_1 + \gamma_2 + \gamma_3 \quad (5.4)$$

che possiamo scrivere come:

$$\gamma_0 = -\gamma_1 - \gamma_2 - \gamma_3 \quad (5.5)$$

Teorema 5.1.1. Teorema di Cayley-Hamilton.

Sia A una matrice quadrata e $p_A(\lambda)$ il suo polinomio caratteristico. Allora:

$$p_A(A) = 0 \quad (5.6)$$

Applicando (5.6) al nostro caso $\forall z \in \mathbb{R}^n$, ricaviamo:

$$p_A(A) = 0 \Rightarrow [\gamma_0 \mathbb{I} + \gamma_1 A + \gamma_2 A^2 + \gamma_3 A^3]z = 0 \quad (5.7)$$

In particolare, prendendo $z = x^{(k-3)}$

$$[\gamma_0 \mathbb{I} + \gamma_1 A + \gamma_2 A^2 + \gamma_3 A^3]x^{(k-3)} = 0, \quad (5.8)$$

e dall'equazione (5.2) si ottiene:

$$\gamma_0 x^{(k-3)} + \gamma_1 x^{(k-2)} + \gamma_2 x^{(k-1)} + \gamma_3 x^{(k)} = 0, \quad (5.9)$$

da cui, sostituendo γ_0 dall'equazione (5.5)

$$(-\gamma_1 - \gamma_2 - \gamma_3)x^{(k-3)} + \gamma_1 x^{(k-2)} + \gamma_2 x^{(k-1)} + \gamma_3 x^{(k)} = 0 \quad (5.10)$$

Raccogliamo ora rispetto a γ_1, γ_2 e γ_3 :

$$(x^{(k-2)} - x^{(k-3)})\gamma_1 + (x^{(k-1)} - x^{(k-3)})\gamma_2 + (x^{(k)} - x^{(k-3)})\gamma_3 = 0 \quad (5.11)$$

Ponendo:

$$y^{(k-2)} = x^{(k-2)} - x^{(k-3)} \quad (5.12)$$

$$y^{(k-1)} = x^{(k-1)} - x^{(k-3)} \quad (5.13)$$

$$y^{(k)} = x^{(k)} - x^{(k-3)} \quad (5.14)$$

possiamo riscrivere (5.11) in forma matriciale:

$$(y^{(k-2)} \quad y^{(k-1)} \quad y^{(k)})\gamma = 0 \quad (5.15)$$

che vogliamo risolvere per $\gamma = \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{pmatrix} \neq \mathbf{0}$.

Possiamo porre:

$$\gamma_3 = 1 \quad (5.16)$$

poiché vincolare un singolo termine del polinomio caratteristico non influenza gli zeri. Rimane dunque da risolvere:

$$(y^{(k-2)} \quad y^{(k-1)}) \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = -y^{(k)} \quad (5.17)$$

Questo sistema è sovradeterminato, perciò per risolvere in γ_1 e γ_2 dobbiamo risolvere il corrispondente problema ai minimi quadrati:

$$\begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = -Y^+ y^{(k)} \quad (5.18)$$

dove Y^+ è la pseudoinversa di $Y = (y^{(k-2)} \quad y^{(k-1)})$, ovvero l'inversa per matrici rettangolari.

Le equazioni (5.5), (5.16) e (5.18) permettono infine di determinare tutti i coefficienti del polinomio caratteristico $p_A(\lambda)$.

Dividiamo ora il polinomio caratteristico appena determinato per $\lambda - 1$, che sappiamo essere possibile poiché $\lambda = 1$ è radice.

Definiamo quindi:

$$q_A(\lambda) := \frac{\gamma_0 + \gamma_1\lambda + \gamma_2\lambda^2 + \gamma_3\lambda^3}{\lambda - 1} = \beta_0 + \beta_1\lambda + \beta_2\lambda^2 \quad (5.19)$$

Da semplici calcoli si ricava che:

$$\beta_0 = \gamma_1 + \gamma_2 + \gamma_3 \quad (5.20)$$

$$\beta_1 = \gamma_2 + \gamma_3 \quad (5.21)$$

$$\beta_2 = \gamma_3 \quad (5.22)$$

Ancora per il *Teorema di Cayley-Hamilton*

$$q_A(A)z = u_1 \quad \forall z \in \mathbb{R}^n \quad (5.23)$$

Per cui, prendendo $z = x^{(k-2)}$,

$$u_1 = q_A(A)x^{(k-2)} = [\beta_0\mathbb{I} + \beta_1A + \beta_2A^2]x^{(k-2)}, \quad (5.24)$$

Dall'equazione (5.2) otteniamo la soluzione in forma chiusa per u_1 :

$$u_1 = \beta_0x^{(k-2)} + \beta_1x^{(k-1)} + \beta_2x^{(k)} \quad (5.25)$$

Ricordiamo che questa iterazione proviene dall'assunzione che A abbia solo tre autovettori. Nel caso in cui A ne abbia di più, quest'iterata sarà soltanto un'approssimazione, che comunque aumenterà di precisione al crescere di n . Anche questo metodo può essere introdotto periodicamente all'interno del metodo delle potenze per accelerare il calcolo del vettore PageRank.

5.2 Algoritmi

Algoritmo 7

```
function x = QuadraticExtrap(x(k-3), x(k-2), x(k-1), x(k))
for i = k - 2 : k
    y(i) = x(i) - x(k-3);
end
Y = (y(k-2) y(k-1));
γ3 = 1;
 $\begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = -Y + y^{(k)}$ ;
γ0 = -(γ1 + γ2 + γ3);
```

```

 $\beta_0 = \gamma_1 + \gamma_2 + \gamma_3;$ 
 $\beta_1 = \gamma_2 + \gamma_3;$ 
 $\beta_2 = \gamma_3;$ 
 $x = \beta_0 x^{(k-2)} + \beta_1 x^{(k-1)} + \beta_2 x^{(k)};$ 
end

```

L'Algoritmo 7 calcola l'iterata successiva mediante estrapolazione quadratica. Esso va introdotto all'interno del metodo delle potenze standard per ottenere un'accelerazione della convergenza.

Algoritmo 8

```

function  $x^{(n)} = \text{QuadraticPotenze}(kmax, \epsilon, v)$ 
 $x^{(0)} = v;$ 
 $k = 1, \delta = 1;$ 
while  $\delta > \epsilon$  and  $k < kmax$ 
     $x^{(k)} = Ax^{(k-1)};$ 
     $\delta = \|x^{(k)} - x^{(k-1)}\|_1;$ 
    Periodicamente:  $x^{(k)} = \text{QuadraticExtrap}(x^{(k-3)}, x^{(k-2)}, x^{(k-1)}, x^{(k)})$ 
     $k = k + 1;$ 
end
end

```

L'Algoritmo 8 mostra come viene modificato il metodo delle potenze standard per ottenere i benefici dell'estrapolazione quadratica. L'estrapolazione avviene periodicamente ogni l iterate.

5.3 Costo computazionale

Il costo computazionale aggiuntivo del metodo di estrapolazione quadratica è quasi interamente contenuto nel calcolo del problema ai minimi

quadrati:

$$\begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = -Y^+ y^{(k)}$$

poiché tutte le altre operazioni aggiuntive hanno un costo computazionale di $\mathcal{O}(n)$ oppure $\mathcal{O}(1)$, che è paragonabile a quello di una singola iterazione del metodo delle potenze.

Il calcolo di un problema ai minimi quadrati solitamente avrebbe un costo computazionale di $\mathcal{O}(n^2)$, ma poiché la matrice Y ha dimensioni $n \times 2$, possiamo risolverlo opportunamente limitandone il costo.

È infatti possibile risolvere il problema con due sole iterazioni del metodo di ortogonalizzazione di Gram-Schmidt, per ottenere la fattorizzazione QR della matrice con un basso costo computazionale.

Algoritmo 9

- 1) Calcolare $Y = QR$ con Gram-Schmidt.
- 2) Risolvere il sistema triangolare superiore:

$$R \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = -Q^T y^{(k)} \text{ usando la sostituzione all'indietro.}$$

L'algoritmo 9 mostra come risolvere il problema ai minimi quadrati che compare nell'estrapolazione quadratica con un costo computazionale di $\mathcal{O}(n)$.

Siccome tutte le operazioni aggiuntive sono dello stesso ordine di quelle di una singola iterazione del metodo delle potenze, possiamo dire che il costo computazionale aggiuntivo è minimo. Bisogna però ricordare che il metodo di estrapolazione quadratica va applicato periodicamente, e che questo comporta un costo maggiore rispetto ai due metodi estrapolativi precedentemente analizzati.

Vedremo se la migliore approssimazione che si sta compiendo giustificherà questo costo più alto.

5.4 Risultati sperimentali

Analizziamo innanzitutto quanto questo metodo migliori l'algoritmo originario del metodo delle potenze.

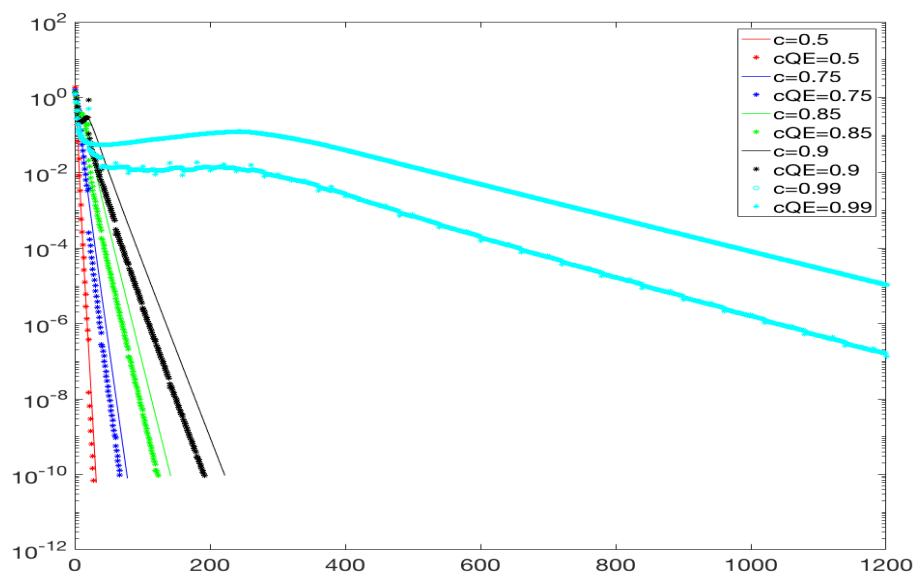


Figura 5.1: Confronto del metodo delle potenze standard e accelerato con QE per diversi valori di c .

La figura 5.1 mostra come, per qualsiasi valore di c , il metodo di estrapolazione quadratica provochi un'accelerazione del metodo. Questa accelerazione cresce al crescere di c , ma è sempre apprezzabile.

La matrice su cui è stato applicato è "Stanford", la frequenza di estrapolazione è 20 iterate.

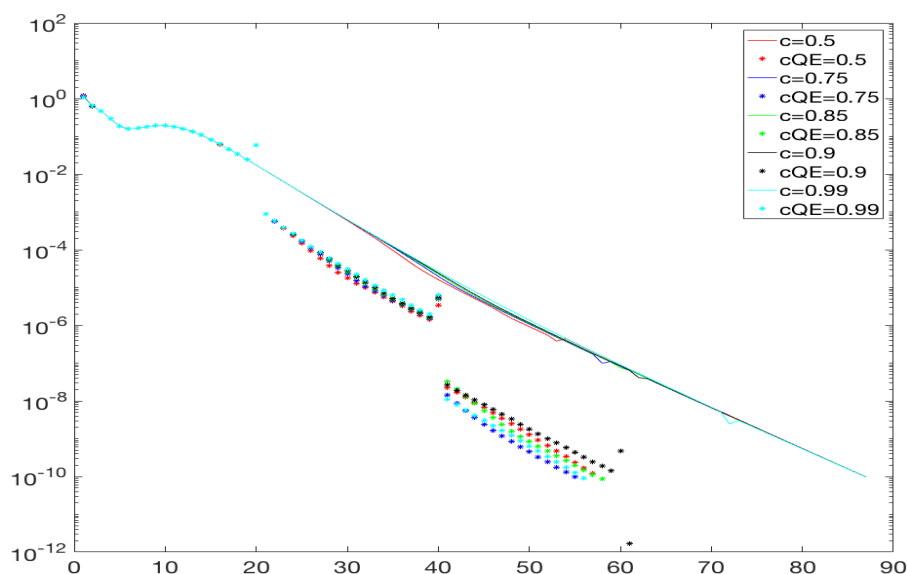


Figura 5.2: Confronto del metodo delle potenze standard e accelerato con QE per vari valori di c .

La figura 5.2 invece esegue lo stesso test sulla matrice “LiveJournal”. In questo caso la grandezza di c non influenza l’accelerazione ottenuta. Anche per questa matrice, però, si ottengono sempre degli ottimi risultati.

5.4.1 Confronto QE, epsilon e Aitken

Per apprezzare meglio la bontà dell’estrpolazione quadratica, confrontiamo il residuo ottenuto estrapolando con Aitken, con epsilon e con QE. La figura 5.3 mostra nel dettaglio come, per $c = 0.5$, il metodo QE sia l’unico che accelera la convergenza.

Ciò è dovuto alla bontà dell’approssimazione che compie: il gradino nel residuo in corrispondenza dell’estrpolazione è infatti assente.

La matrice su cui è stato effettuato il test è “Stanford”.

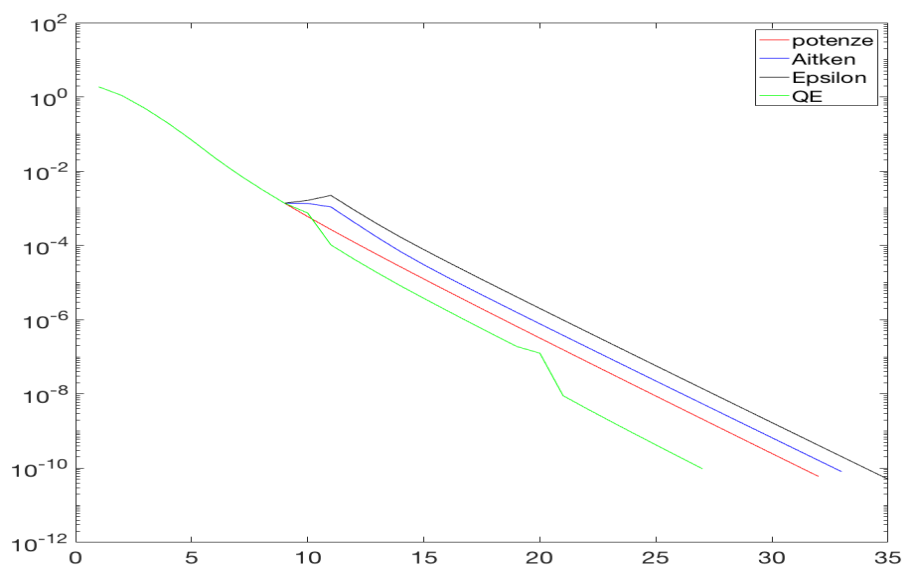


Figura 5.3: Confronto del metodo di estrapolazione quadratica, epsilon e Aitken per $c = 0.5$.

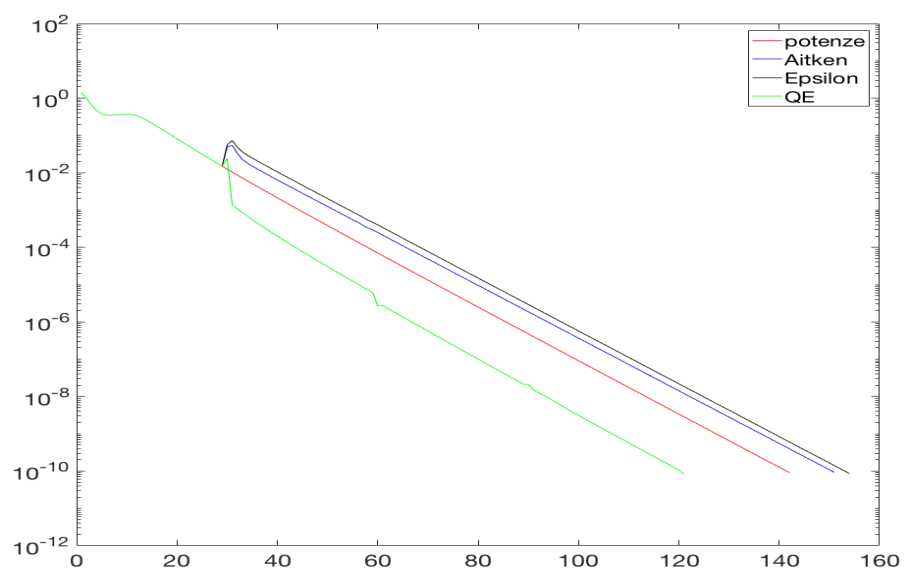


Figura 5.4: Confronto del metodo di estrapolazione quadratica, epsilon e Aitken per $c = 0.85$.

In figura 5.4 è stato eseguito lo stesso test per un diverso valore di c . Anche in questo caso, l'unico metodo utile è QE.

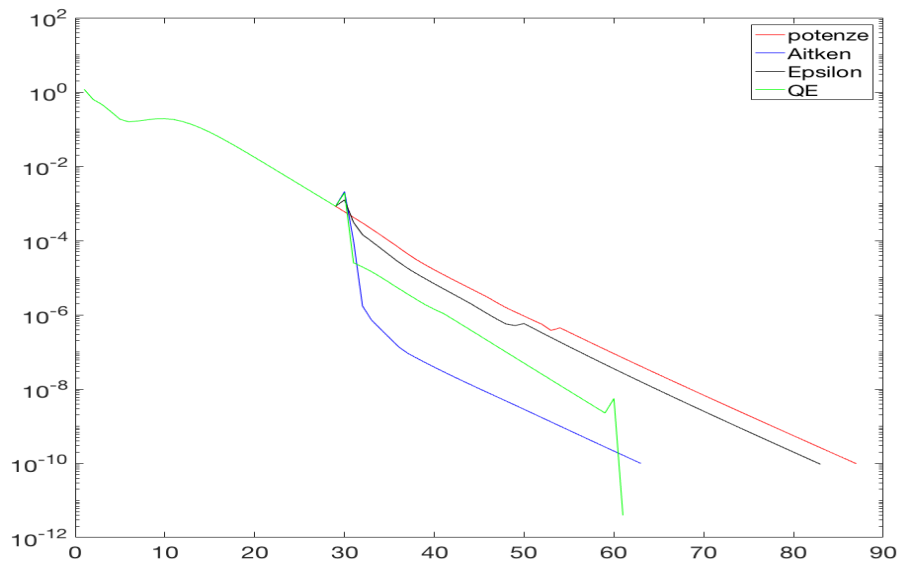


Figura 5.5: Confronto del metodo di estrapolazione quadratica, epsilon e Aitken per $c = 0.5$

Ancora, la figura 5.5 riporta lo stesso test per $c = 0.5$, eseguito però sulla matrice “LiveJournal”. In questo caso, tutti i metodi riescono ad accelerare la convergenza, ma l’estrpolazione quadratica si rivela ancora il migliore. Spicca la brusca accelerazione guadagnata subito dopo le due estrapolazioni quadratiche.

La frequenza di applicazione di estrapolazione quadratica è di 30 iterate.

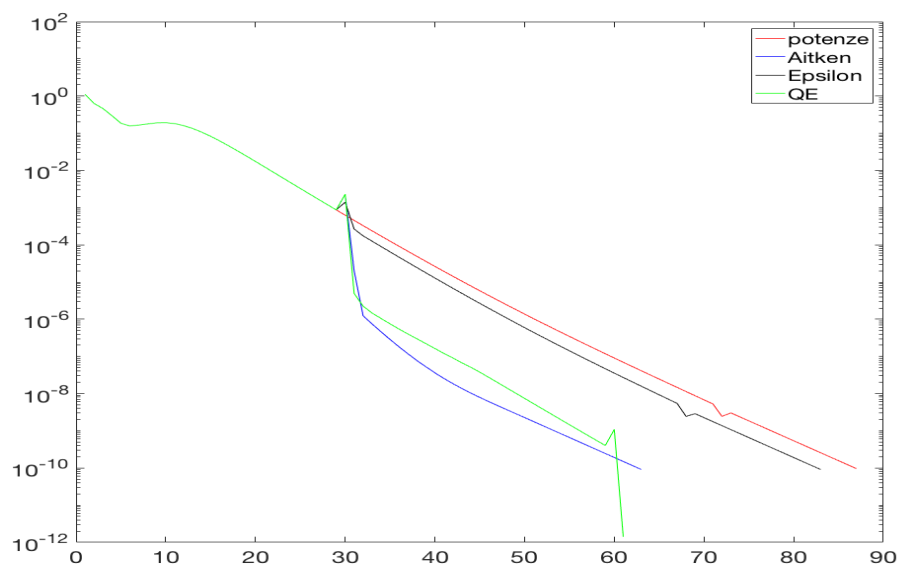


Figura 5.6: Confronto del metodo di estrapolazione quadratica, epsilon e Aitken per $c = 0.99$.

Infine, la figura 5.6 esegue sempre lo stesso test per $c = 0.99$ sulla matrice “LiveJournal”. La similarità del residuo ottenuto per $c = 0.5$ è evidente.

5.4.2 Frequenza di applicazione e residuo

Una delle proprietà più interessanti dell'estrapolazione quadratica è che non è necessario applicarla troppo frequentemente per ottenere comunque una grande accelerazione. In questa sezione studiamo la relazione tra la frequenza di estrapolazione e il residuo che ne risulta.

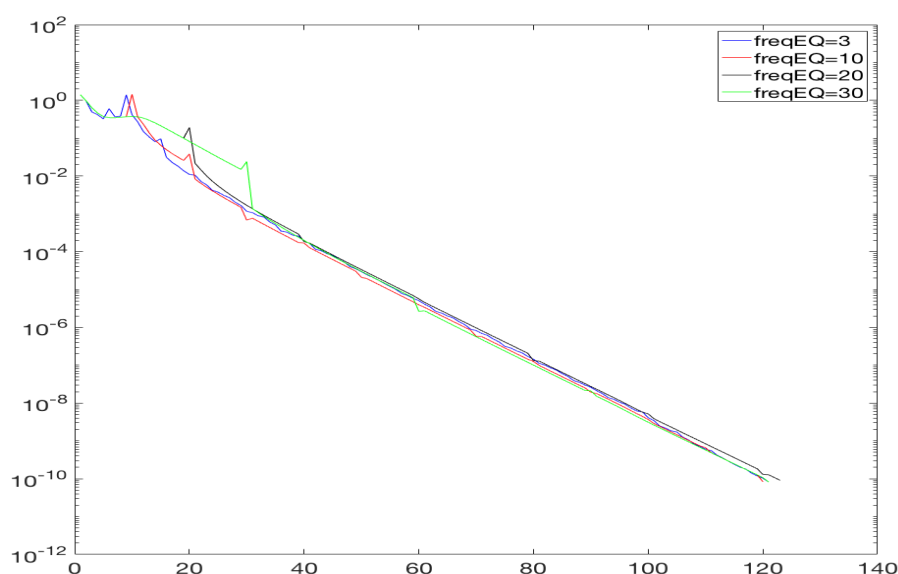


Figura 5.7: Confronto del residuo ottenuto estrapolando con diverse frequenze per $c = 0.85$.

Il test riportato in figura 5.7 è stato eseguito sulla matrice “Stanford”, che è quella meno sparsa. In questo caso, estrapolare di frequente o di rado quasi non ha effetto sul residuo finale. Però, estrapolando spesso, il costo computazionale aggiuntivo diventa più elevato.

Lo stesso test applicato sulla matrice “Google”, come si vede nella figura 5.8, favorisce leggermente l’applicazione più frequente dell’estrapolazione. Poiché questa differenza è comunque limitata, essa non giustifica il grosso costo computazionale che si aggiunge all’algoritmo originario.

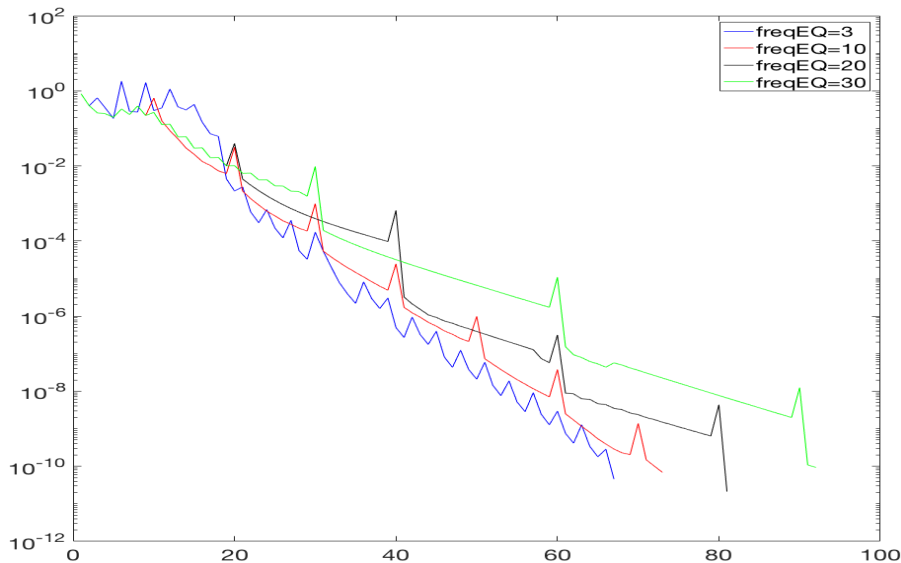


Figura 5.8: Confronto del residuo ottenuto estrapolando con diverse frequenze per $c = 0.99$.

Abbiamo quindi visto come non sia necessario estrapolare troppo frequentemente per ottenere grossi benefici. D'altra parte, estrapolare troppo di rado ha gli stessi effetti, perché si limitano i benefici del metodo. Investigheremo ora quale sia la frequenza ottimale per ridurre maggiormente il tempo richiesto per la convergenza.

5.4.3 Frequenza ottimale di estrapolazione

Cerchiamo adesso di capire quale sia la frequenza di estrapolazione che assicura il minor tempo necessario per la convergenza. Per farlo, applichiamo l'estrapolazione quadratica a tutte e tre le matrici, per diversi valori di c , con diverse frequenze.

In tabella 5.1 troviamo i secondi richiesti per arrivare a convergenza sulla matrice "Stanford". Osserviamo come l'applicazione troppo frequente dell'estrapolazione abbia i risultati peggiori per qualsiasi valore di c , addirittura peggiori rispetto all'algoritmo originario.

$\#QE$:	$c = 0.5$	$c = 0.85$	$c = 0.99$
3	0.8613	3.5155	55.6983
10	0.7279	3.1781	43.9534
30	0.7029	2.6427	40.4640
50	0.6648	2.5094	40.9670
200	0.6880	2.9195	42.6660
Mai	0.7099	3.0798	51.8182

Tabella 5.1: Tempi di convergenza per la matrice “Stanford”.

Ciò è dovuto all’incremento del costo computazionale, senza che ne derivi una soddisfacente riduzione del residuo.

La frequenza ottimale per questa matrice è intorno alle 50 iterate.

$\#QE$:	$c = 0.5$	$c = 0.85$	$c = 0.99$
3	4.2799	4.3262	4.6806
10	4.0268	4.4387	4.5420
30	5.5759	5.6487	5.6692
50	6.1370	6.2959	6.3730
200	8.8247	9.0863	9.1023
Mai	9.0220	9.1475	9.3386

Tabella 5.2: Tempi di convergenza per la matrice “Google”.

Per la matrice “Google”, invece, i risultati sono differenti.

La frequenza ottimale cresce molto: si aggira intorno alle 10 iterate, e soprattutto non presenta molta differenza tra i vari valori di c .

Ricordiamo come questa matrice sia molto più sparsa e grande dimensionalmente rispetto a quella precedente.

#QE:	$c = 0.5$	$c = 0.85$	$c = 0.99$
3	23.2347	22.8735	24.9413
5	22.6219	22.9425	23.0279
10	26.3708	25.1714	25.8282
30	36.7211	36.8796	36.4493
50	39.5390	42.6744	44.8090
200	49.6705	51.0695	52.4919
Mai	50.7074	51.0626	52.9576

Tabella 5.3: Tempi di convergenza per la matrice “LiveJournal”.

Per la matrice “LiveJournal”, infine, la frequenza ottimale cresce ancora, diventando di 3 iterate, ovvero la più alta frequenza per cui può essere applicata l’extrapolazione quadratica. Anche in questo caso, come mostra la figura 5.3, il valore di c non influenza il tempo richiesto per la convergenza, che si mantiene pressoché costante lungo le righe delle matrici.

5.4.4 Conclusioni

Il metodo di extrapolazione quadratica è risultato il miglior metodo di extrapolazione per accelerare il calcolo del vettore PageRank.

In tutti i test effettuati, i suoi risultati sono i più soddisfacenti, indipendentemente dalla matrice utilizzata o dal valore di c . Questo accade per via della bontà dell’approssimazione che si compie al momento dell’extrapolazione - che risulta essere la migliore - e del costo computazionale aggiuntivo che, per iterazione, è paragonabile agli altri due metodi.

Per ottenere i benefici migliori, questa extrapolazione va applicata periodicamente, ma la frequenza ottimale di applicazione è molto influenzata dalla matrice su cui viene applicato. I risultati sperimentali sembrano indicare che, al crescere delle dimensioni della matrice, cresca anche la frequenza ottimale.

Conclusioni

Abbiamo analizzato il comportamento di tre diversi metodi estrapolativi per il calcolo del vettore PageRank.

Tutti e tre hanno avuto dei discreti risultati finali, accelerando l'algoritmo di base, il metodo delle potenze, e limitando così il tempo richiesto per risolvere il problema in questione.

In particolare, l'estrapolazione di Aitken ottiene dei buoni risultati se applicato dopo un certo numero di iterate, quando cioè il metodo delle potenze ha già indirizzato l'iterata verso la convergenza. L'approssimazione che si compie è abbastanza grossolana, per cui, appena applicato questo metodo, si forma un gradino nel residuo. Il rallentamento viene però colmato dall'accelerazione che si guadagna proprio grazie all'approssimazione compiuta. Qualora c fosse abbastanza limitato da garantire già una buona velocità di convergenza dell'algoritmo originario, questo metodo non sarebbe indicato poiché non riuscirebbe a recuperare il rallentamento iniziale. I risultati che si ottengono per c grande, invece, sono decisamente soddisfacenti.

L'estrapolazione di Aitken è pertanto indicata proprio per sopperire ai difetti del metodo delle potenze.

L'estrapolazione epsilon è molto simile a quella di Aitken. L'approssimazione che si compie è più precisa, ma richiede un maggior costo computazionale e perciò il gradino iniziale è meno marcato, ma l'accelerazione che si ottiene è chiaramente minore. Per questa ragione, questo metodo

si comporta meglio per valori di c bassi, mentre è più lento per c grande. Siccome questi due metodi estrapolativi sono indicati per migliorare l'algoritmo originario proprio per valori grandi di c , l'estrapolazione epsilon è meno adatta a calcolare il vettore PageRank rispetto all'estrapolazione di Aitken.

Il metodo di estrapolazione quadratica, infine, ottiene risultati egregi in tutti i casi. L'approssimazione compiuta è la più precisa e anche l'accelerazione guadagnata è la migliore. A fronte di ciò, il costo computazionale aggiuntivo è il maggiore. La sua peculiarità è l'applicazione periodica, che, chiaramente è partecipe del costo computazionale che si somma all'algoritmo originario. Per questa ragione i risultati più soddisfacenti si ottengono a una certa frequenza ottimale di applicazione, che però è largamente influenzata dalla matrice su cui il metodo viene applicato.

In ogni caso, questo metodo è il più indicato per velocizzare la soluzione del problema, poiché i risultati ottenuti sono i migliori per qualsiasi valore di c , anche qualora la frequenza di applicazione non sia quella ottimale.

Bibliografia

- [1] Sepandar D. Kamvar, Taher H. Haveliwala, Christopher D. Manning and Gene H. Golub, *Extrapolation methods for accelerating PageRank computations*, Proceedings of the twelfth international World Wide Web conference, May 2003.
- [2] Claude Brezinski and Michela Redivo-Zaglia, *The PageRank vector: properties, computation, approximation and acceleration*, SIAM J. MATRIX ANAL. APPL. Vol. 28, No. 2, pp. 551–575, 2006
- [3] Davide Palitta e Valeria Simoncini, *Dispense del corso di Calcolo Numerico. Modulo di Algebra Lineare*, 2016.
- [4] Valeria Simoncini, *Dispense del corso di Matematica Computazionale*, 2017.

Ringraziamenti

Un ringraziamento speciale va alla relatrice, la Prof.ssa Valeria Simoncini, per la disponibilità e la celerità dimostrate nella correzione di questo elaborato.

Vorrei ringraziare inoltre Francesca, per essere stata un punto di riferimento in questi tre anni.

Un grazie anche a Marta, Giulia e Francesco, per il sostegno e l'aiuto prestato nelle fasi finali della scrittura.

Infine, vorrei ringraziare tutti i presenti per aver condiviso con me questo giorno.