Alma Mater Studiorum · Università di Bologna

Scuola di Scienze
Corso di Laurea in Matematica

IL METODO DELLE POTENZE PER IL PAGERANKING

Relatore:
Chiar.ma Prof.ssa
VALERIA SIMONCINI

Presentata da:
MARIANNA PALESTINI

Sessione II

Anno Accademico 2015/2016

Ai miei genitori e alle mie sorelle,
che hanno creduto in me e continuano a farlo.

Indice

Introduzione

Quando si fa una ricerca su Internet con i motori di ricerca, tipo Google, lo scopo è quello di trovare tutte le pagine Web contenenti le parole richieste. Ancor di più, si vorrebbe avere la miglior risposta alla nostra richiesta tra le prime pagine lette, senza dover passare in rassegna tutte le migliaia di pagine che il motore di ricerca propone.
Quello che riceviamo come risposta dal motore di ricerca, è infatti una lista lunghissima di pagine Web, contenenti le parole chiave e ordinate in base all'importanza. Nelle prime posizioni della lista troveremo le pagine più significative, mentre in fondo ci saranno quelle di meno rilevanza.
Ma come viene stabilito se una pagina è più importante di un' altra? Con quale criterio vengono ordinate le pagine?
In questa tesi affronteremo, appunto, il problema di classificazione, il quale non è basato su un giudizio umano, ma sulla struttura dei link del Web. Studieremo, per l'esattezza, il concetto di *pagerank* usato da Google e di conseguenza l'algoritmo, chiamato proprio PageRank, per classificare le pagine Web. Vedremo che il tutto si basa su un'equazione matematica, ossia sul calcolo dell'autovettore, relativo all' autovalore uguale a 1, di una certa matrice che rappresenta la rete Internet.

Nel Capitolo 1 introdurremo alcune definizioni e proprietà che ci saranno utili all'interno dei capitoli successivi. Tratteremo, quindi, alcuni elementi sulle matrici e spiegheremo cos'è un grafo, il quale ci servirà per schematizzare la rete Internet.

Nel Capitolo 2 verrà presentato il *metodo delle potenze*, usato per l'approssimazione dell'autovalore massimo e autovettore ad esso associato, fondamentale in quanto usato nell'algoritmo PageRank.

Nel Capitolo 3 si studierà il pagerank in tutti i suoi aspetti, iniziando a capire cos'è e successivamente a calcolarlo. Si farà un'analisi dettagliata e si applicherà il metodo descritto nel capitolo precedente, modificato per questo

caso specifico. Alla fine del capitolo verrà presentato un esempio reale, applicato alla rete dei links delle Università di Stanford e Berkeley.

Nelle Appendici A e B sono presenti i codici MATLAB, dal caricamento della matrice che rappresenta i collegamenti ipertestuali dei siti, all'algoritmo PageRank con lo script di chiamata.

Capitolo 1

Nozioni utili

Introduciamo alcuni elementi, sulle matrici e sui grafi, che ci saranno utili per questa tesi.

1.1 Matrici e norme vettoriali

Iniziamo con dare alcune definizioni, di matrici e norme, che useremo nei prossimi capitoli.

Definizione 1.1. Sia $\mathbf{A} \in \mathbb{R}^{n\times n}$. $\mathbf{A}$ si dice *simmetrica* se $\mathbf{A} = \mathbf{A}^T$.
$\mathbf{A}$ si dice *ortogonale* se $\mathbf{A}^T\mathbf{A} = \mathbf{A}\mathbf{A}^T = \mathbf{I}$, dove $\mathbf{I}$ è la matrice identità.

Definizione 1.2. Sia $\mathbf{A} \in \mathbb{C}^{n\times m}$. La matrice $\mathbf{A}^H$, detta *matrice aggiunta*, è la trasposta della matrice complessa coniugata $\bar{\mathbf{A}}$.

Definizione 1.3. Sia $\mathbf{A} \in \mathbb{C}^{n\times n}$. $\mathbf{A}$ si dice *hermitiana* se $\mathbf{A} = \mathbf{A}^H$.
$\mathbf{A}$ si dice *unitaria* se $\mathbf{A}^H\mathbf{A} = \mathbf{A}\mathbf{A}^H = \mathbf{I}$, dove $\mathbf{I}$ è la matrice identità.

Definizione 1.4. Sia $\mathbf{A} \in \mathbb{R}^{n\times m}$. Si scrive $\mathbf{A} > 0$, se tutti gli elementi di $\mathbf{A}$ sono strettamente positivi, ossia se

$$A_{i,j} > 0 \quad \forall\, i = 1, \ldots, n, \quad \forall\, j = 1, \ldots, m.$$

In questo caso, la matrice si dice *positiva*.

Definizione 1.5. Una matrice $\mathbf{A}$ quadrata di ordine n, si dice *diagonale* se gli unici elementi non nulli sono quelli sulla diagonale principale, ossia se

$$A_{i,j} = 0, \quad i \neq j, \quad \forall\, i, j = 1, \ldots, n.$$

Definizione 1.6. Una matrice $\mathbf{A}$ si dice *sparsa* se il numero di elementi non nulli è solo del $3 - 5\%$, ossia tale da rendere conveniente il tenerne conto.

Definizione 1.7. Una matrice quadrata $\mathbf{A}$ è detta *riducibile* se esiste una matrice di permutazione $\mathbf{Q}$, tale che

$$\mathbf{QAQ}^{-1} = \begin{pmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{0} & \mathbf{Z} \end{pmatrix},$$

dove $\mathbf{X}$ e $\mathbf{Z}$ sono entrambe quadrate.
Altrimenti la matrice è detta *irriducibile*.

Definizione 1.8. Una *norma vettoriale* è un'applicazione $\|\cdot\| : \mathbb{C}^n \to \mathbb{R}$ tale che

1. $\|\mathbf{x}\| \geq 0 \quad \forall\, \mathbf{x} \in \mathbb{C}^n \quad$ e $\quad \|\mathbf{x}\| = 0$ se e solo se $\mathbf{x} = 0$;
2. $\|\lambda \mathbf{x}\| = |\lambda| \|\mathbf{x}\| \quad \forall\, \lambda \in \mathbb{C}, \quad \forall\, \mathbf{x} \in \mathbb{C}^n$;
3. $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\| \quad \forall\, \mathbf{x}, \mathbf{y} \in \mathbb{C}^n$.

In questo elaborato utilizzeremo la norma-1:

Definizione 1.9. Sia $\mathbf{x} \in \mathbb{C}^n,\ \mathbf{x} = (x_1, \ldots, x_n)^T$, si definisce *norma*-1 di $\mathbf{x}$,

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|.$$

1.2 Autovalori e autovettori

È necessario introdurre alcune definizioni e osservazioni su autovalori e autovettori di una matrice.
Successivamente, infatti, descriveremo il *metodo delle potenze*, il quale approssima autovalori e autovettori estremi, fondamentale per ciò che andremo a studiare.

Definizione 1.10. Sia $\mathbf{A}$ una matrice quadrata di ordine n, il numero $\lambda \in \mathbb{C}$ è detto *autovalore* di $\mathbf{A}$ se $\exists\, \mathbf{x} \in \mathbb{C}^n, \mathbf{x} \neq 0$, tale che

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

Il vettore $\mathbf{x}$ è detto *autovettore* associato all'autovalore λ, e $(\lambda, \mathbf{x})$ è detta *autocoppia*.

Definizione 1.11. Sia $\mathbf{A}$ una matrice quadrata di ordine n.
I vettori $\mathbf{x}, \mathbf{y} \in \mathbb{C}^n$ si dicono rispettivamente *autovettore destro* e *sinistro* di $\mathbf{A}$, associati all'autovalore $\lambda \in \mathbb{C}$, se

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}, \qquad \mathbf{y}^H\mathbf{A} = \lambda\mathbf{y}^H.$$

L'autovalore λ, di una matrice quadrata $\mathbf{A}$, è soluzione dell'equazione caratteristica

$$p_{\mathbf{A}}(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I}) = 0,$$

dove $\mathbf{I}$ è la matrice identità e $p_{\mathbf{A}}(\lambda)$ è detto *polinomio caratteristico* di $\mathbf{A}$.

Di conseguenza, gli autovalori di una matrice triangolare inferiore
$\mathbf{A} = \begin{pmatrix} A_{1,1} & 0 & \dots & 0 \\ A_{2,1} & A_{2,2} & \dots & 0 \\ \vdots & & \ddots & \vdots \\ A_{n,1} & A_{n,2} & \dots & A_{n,n} \end{pmatrix}$, sono rappresentati dagli elementi sulla diagonale.

Proposizione 1.1. *Se* $\mathbf{A}$ *è una matrice quadrata a coefficienti reali allora gli autovalori complessi di* $\mathbf{A}$ *sono necessariamente complessi coniugati.*

Proposizione 1.2. *Sia* $\mathbf{A} \in \mathbb{R}^{n\times n}$ *e* λ *un autovalore reale di* $\mathbf{A}$*, allora esiste un autovettore di* $\mathbf{A}$ *relativo a* λ *a coefficienti reali.*

L'autovalore λ di una matrice quadrata $\mathbf{A}$, corrispondente all'autovettore $\mathbf{x}$ si ottiene calcolando

$$\frac{\mathbf{x}^H\mathbf{A}\mathbf{x}}{\mathbf{x}^H\mathbf{x}},$$

detto quoziente di Rayleigh.

Definizione 1.12. Sia $\mathbf{A}$ una matrice quadrata di ordine n e $\lambda \in \mathbb{C}$ autovalore di $\mathbf{A}$. Si chiama *molteplicità algebrica* di λ, la molteplicità che λ ha come radice del polinomio caratteristico.
L'autovalore λ si dice *semplice* se ha molteplicità algebrica 1, altrimenti si dice *multiplo*.

1.3 Trasformazioni per similitudine

Definizione 1.13. Siano $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{n\times n}$, si dicono *simili* se $\exists\ \mathbf{S} \in \mathbb{C}^{n\times n}$ invertibile, tale che

$$\mathbf{A} = \mathbf{S}^{-1}\mathbf{B}\mathbf{S}.$$

La trasformazione da $\mathbf{B}$ a $\mathbf{S}^{-1}\mathbf{B}\mathbf{S}$ è detta *trasformazione per similitudine.*

Definizione 1.14. Una matrice $\mathbf{A} \in \mathbb{C}^{n\times n}$ è *diagonalizzabile* se è simile a una matrice diagonale. Ossia se $\exists\ \mathbf{S} \in \mathbb{C}^{n\times n}$ invertibile, tale che

$$\mathbf{A} = \mathbf{S}^{-1}\mathbf{D}\mathbf{S},$$

dove $\mathbf{D} \in \mathbb{C}^{n\times n}$ è matrice diagonale.

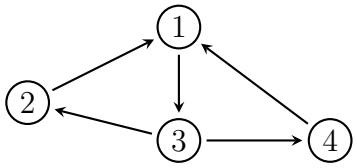
Due matrici simili hanno gli stessi autovalori, con le stesse molteplicità.

1.4 Grafi

Definizione 1.15. Un *grafo* $G = (N, E)$ è costituito da una coppia di insiemi , dove N è insieme di nodi (o vertici), ed E è insieme di archi, tali che gli elementi di E siano coppie di elementi di N.

Definizione 1.16. Un grafo $G = (N, E)$ è detto *diretto* (o *orientato*) se E è insieme di archi orientati.

Esempio 1.1. Grafo diretto:



dove $1, 2, 3, 4$ sono i nodi, e le frecce sono gli archi orientati.

Diremo che N_i *punta a* N_j se c'è un arco che parte dal nodo N_i, orientato verso il nodo N_j.

Definizione 1.17. Dato un grafo $G = (N, E)$, un *cammino* da a a b è una sequenza di nodi $\{N_0, N_1, \ldots, N_k\}$, con $k \geq 1$, tale che

$$a = N_0, \quad b = N_k \quad \text{e} \quad (N_{i-1}, N_i) \in E \quad \text{per } i = 1, \ldots, k.$$

Se il grafo è diretto, il cammino si dice orientato.

Definizione 1.18. Un grafo diretto $G = (N, E)$ si dice *fortemente connesso* se per ogni coppia di nodi (N_i, N_j) esiste un cammino orientato da N_i a N_j e viceversa.

Capitolo 2

Metodo delle Potenze

Il *metodo delle potenze* è un metodo iterativo particolarmente adatto per il calcolo degli autovalori estremi della matrice e i relativi autovettori associati.
La soluzione di tale problema è di grande interesse in numerosi problemi suggeriti da applicazioni reali, dove il calcolo di tale autovalore, e del corrispondente autovettore associato, è collegato alla determinazione di certe grandezze fisiche.
Quello che interessa a noi è l'approssimazione dell'autovalore massimo λ_1 di una matrice $\mathbf{A}$, e dell'autovettore ad esso corrispondente.

Sia $\mathbf{A}$ una matrice quadrata di ordine n diagonalizzabile. $\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}$, dove $\mathbf{X} \in \mathbb{C}^{n\times n}$ è matrice dei suoi autovettori destri $\mathbf{x}_i$, $i = 1, \ldots, n$, e $\mathbf{\Lambda} = diag(\lambda_1, \ldots, \lambda_n)$ è la matrice diagonale dei suoi autovalori.
Supponiamo, inoltre, che gli autovalori di $\mathbf{A}$ siano ordinati in modo tale che

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|,$$

dove λ_1 ha molteplicità algebrica 1. Sotto tali ipotesi, λ_1 è detto autovalore *dominante* per la matrice $\mathbf{A}$.
Fissato un vettore arbitrario iniziale $\mathbf{x}^{(0)} \in \mathbb{C}^n$ di norma euclidea unitaria, l'iterazione del metodo delle potenze consiste nella seguente:

Per $k = 0, 1, 2, \ldots$, fino a convergenza

$$\begin{aligned}
\mathbf{y}^{(k+1)} &= \mathbf{A}\mathbf{x}^{(k)} \\
\mathbf{x}^{(k+1)} &= \mathbf{y}^{(k+1)}/\|\mathbf{y}^{(k+1)}\| \\
\lambda^{(k+1)} &= (\mathbf{x}^{(k+1)})^H \mathbf{A}\mathbf{x}^{(k+1)}
\end{aligned}$$

Indicata con $(\lambda_1, \mathbf{x}_1)$ l'autocoppia dominante di $\mathbf{A}$, avremo due successioni $\{\mathbf{x}^{(k)}\}_{k\in\mathbb{N}}$ e $\{\lambda^{(k)}\}_{k\in\mathbb{N}}$ tali che

$$\mathbf{x}^{(k)} \longrightarrow \mathbf{x}_1 \quad \text{e} \quad \lambda^{(k)} \longrightarrow \lambda_1 \quad \text{per } k \to \infty.$$

Essendo $\mathbf{A}$ diagonalizzabile, è possibile scrivere l'iterato $\mathbf{x}^{(k)}$ del metodo delle potenze nel seguente modo:

$$\begin{aligned}\mathbf{x}^{(k)} = \mathbf{A}^k\mathbf{x}^{(0)} &= \mathbf{X}\mathbf{\Lambda}^k \underbrace{\mathbf{X}^{-1}\mathbf{x}^{(0)}}_{=:\mathbf{y}} = \begin{pmatrix}\mathbf{x}_1 & \dots & \mathbf{x}_n\end{pmatrix}\begin{pmatrix}\lambda_1^k & & \\ & \ddots & \\ & & \lambda_n^k\end{pmatrix}\begin{pmatrix}y_1 \\ \vdots \\ y_n\end{pmatrix} \\ &= \mathbf{x}_1\lambda_1^k y_1 + \sum_{i=2}^{n}\mathbf{x}_i\lambda_i^k y_i \\ &= \lambda_1^k(\mathbf{x}_1 y_1 + \sum_{i=2}^{n}\mathbf{x}_i\left(\frac{\lambda_i}{\lambda_1}\right)^k y_i) \to \alpha\mathbf{x}_1 \qquad \text{per } k \to \infty,\end{aligned}$$

dato che $|\lambda_1| > |\lambda_i| \quad \text{con} \quad i \geq 2 \;\Rightarrow\; \left(\frac{\lambda_i}{\lambda_1}\right)^k \to 0 \qquad \text{per } k \to \infty.$

Quindi $\mathbf{x}^{(k)}$ tende a essere parallelo all'autovettore associato all'autovalore dominante, o meglio a coincidere con esso.
Nel caso in cui λ_1 fosse multiplo, non comporterebbe problemi per la convergenza. Fondamentale è, invece, l'ipotesi che non ci siano autovalori distinti da λ_1 ma con lo stesso modulo. A tal proposito, se la matrice $\mathbf{A}$ è reale, l'autovalore dominante deve essere necessariamente reale, altrimenti, per la Proposizione 1.1 avrei due autovalori λ e $\bar{\lambda}$, diversi ma con stesso modulo.

Al passo k si ha la seguente stima dell'errore:

Teorema 2.1. *Sia* $\mathbf{A}$ *una matrice quadrata di ordine* n*, diagonalizzabile, i cui autovalori soddisfino* $|\lambda_1| > |\lambda_i|$*,* $i = 2, \dots, n$*, con* $|\lambda_1|$ *semplice.*
Assumendo $\mathbf{x}^{(0)} = \mathbf{X}\mathbf{y}$ *tale che* $y_1 \neq 0$*, esiste una costante* $C > 0$ *tale che*

$$\|\mathbf{x}_1 - \mathbf{x}^{(k)}\| \leq C\left|\frac{\lambda_2}{\lambda_1}\right|^k, \qquad k \geq 1. \tag{2.1}$$

La (2.1) esprime la convergenza della successione $\mathbf{x}^{(k)}$ all'autovettore $\mathbf{x}_1$. Quindi la convergenza sarà tanto più rapida quanto più piccolo è il rapporto $|\lambda_2/\lambda_1|$.

Nel caso in cui la matrice $\mathbf{A}$ è Hermitiana, quindi $\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^H$, con $\mathbf{X}$ unitaria, si può mostrare, sempre assumendo $y_1 \neq 0$, che vale la seguente stima

$$|\lambda_1 - \lambda^{(k)}| = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right).$$

La convergenza della successione $\lambda^{(k)}$ a λ_1 per $\mathbf{A}$ Hermitiana, è quindi quadratica rispetto al rapporto $|\lambda_2/\lambda_1|$. Infatti, sia $\mathbf{x}^{(k)} = \mathbf{A}^k\mathbf{x}^{(0)}$, si ha

$$\begin{aligned}
\lambda^{(k)} &= \frac{\left(\mathbf{x}^{(k)}\right)^H \mathbf{A}\mathbf{x}^{(k)}}{\left(\mathbf{x}^{(k)}\right)^H \mathbf{x}^{(k)}} = \frac{\left(\mathbf{A}^k\mathbf{x}^{(0)}\right)^H \mathbf{A}\left(\mathbf{A}^k\mathbf{x}^{(0)}\right)}{\left(\mathbf{A}^k\mathbf{x}^{(0)}\right)^H \left(\mathbf{A}^k\mathbf{x}^{(0)}\right)} = \frac{\left(\mathbf{x}^{(0)}\right)^H \mathbf{A}^{2k+1}\mathbf{x}^{(0)}}{\left(\mathbf{x}^{(0)}\right)^H \mathbf{A}^{2k}x^{(0)}} \\
&= \frac{\left(\mathbf{X}^H\mathbf{x}^{(0)}\right)^H \mathbf{\Lambda}^{2k+1}\left(\mathbf{X}^H\mathbf{x}^{(0)}\right)}{\left(\mathbf{X}^H\mathbf{x}^{(0)}\right)^H \mathbf{\Lambda}^{2k}\left(\mathbf{X}^H\mathbf{x}^{(0)}\right)} = \frac{\mathbf{y}^H\mathbf{\Lambda}^{2k+1}\mathbf{y}}{\mathbf{y}^H\mathbf{\Lambda}^{2k}\mathbf{y}} = \frac{\displaystyle\sum_{i=1}^{n}\lambda_i^{2k+1}|y_i|^2}{\displaystyle\sum_{i=1}^{n}\lambda_i^{2k}|y_i|^2} \\
&= \frac{\lambda_1^{2k+1}|y_1|^2 + \displaystyle\sum_{i=2}^{n}\lambda_i^{2k+1}|y_i|^2}{\lambda_1^{2k}|y_1|^2 + \displaystyle\sum_{i=2}^{n}\lambda_i^{2k}|y_i|^2} = \lambda_1\frac{|y_1|^2 + \displaystyle\sum_{i=2}^{n}\left(\frac{\lambda_i}{\lambda_1}\right)^{2k+1}|y_i|^2}{|y_1|^2 + \displaystyle\sum_{i=2}^{n}\left(\frac{\lambda_i}{\lambda_1}\right)^{2k}|y_i|^2} \\
&\leq \lambda_1\left(1 + \frac{1}{|y_i|^2}\sum_{i=2}^{n}\left(\frac{\lambda_i}{\lambda_1}\right)^{2k+1}|y_i|^2\right) = \lambda_1\left(1 + \mathcal{O}\left(\frac{\lambda_2}{\lambda_1}\right)^{2k+1}\right).
\end{aligned}$$

Concludiamo dicendo che un criterio di arresto per il metodo delle potenze si ha usando il residuo scalare. Introducendo, quindi, al passo k

$$\mathbf{r}^{(k+1)} = \mathbf{A}\mathbf{x}^{(k+1)} - \lambda^{(k+1)}\mathbf{x}^{(k+1)},$$

un test di arresto relativo può essere

$$\frac{\|\mathbf{r}^{(k+1)}\|}{|\lambda_1|} \leq tol,$$

dove *tol* è la tolleranza fissata.

Capitolo 3

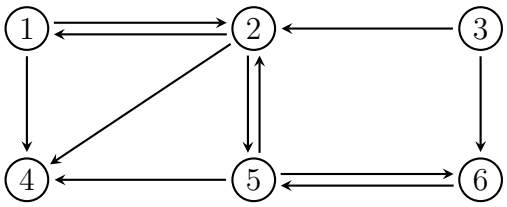
Page Ranking per Google

Il *PageRank* di Google nasce come algoritmo di analisi. Tale algoritmo assegna un peso numerico, detto pagerank, ad ogni pagina Web di un collegamento ipertestuale (hyperlink) di un insieme di documenti, con lo scopo di quantificare l'importanza di ogni pagina.
L'algoritmo PageRank non considera l'effettivo contenuto delle pagine, né si basa su un giudizio umano, tiene conto unicamente della struttura del Web, ossia dei collegamenti ipertestuali.
Quando un utente naviga in Internet fa una teorica passeggiata aleatoria, dove sceglie casualmente dei links per passare da una pagina all'altra. La probabilità che questo ipotetico navigatore, dopo un gran numero di passi, visiti una specifica pagina è il pagerank della pagina.
La strategia scelta da Google per calcolare il pagerank si basa sul presupposto che il numero dei links da e verso una pagina Web dia informazioni sull'importanza di essa.

3.1 Pagerank

Iniziamo col definire il Web come un grafo. La struttura di hyperlink del Web forma infatti un grafo diretto, dove i nodi rappresentano le pagine Web e gli archi diretti o links rappresentano gli hyperlinks.
Supponiamo che tutte le pagine Web siano ordinate da 1 a n e supponiamo che i sia una pagina specifica. Chiamiamo *outlinks* di i le pagine che vengono puntate da i, mentre *inlinks* di i le pagine che hanno come outlinks i.

Esempio 3.1. Il seguente grafo illustra un insieme di 6 pagine Web con i loro oulinks e inlinks:



La pagina 1 ha come outlinks le pagine 2, 4 e come inlinks solo la 2, mentre la pagina 5 ha come outlinks le pagine 2, 6 e come inlinks la 2, la 4 e la 6, e così via ...

L'idea alla base della strategia PageRank è

"una pagina i è più importante più sono gli inlinks".

Preso così questo criterio, tuttavia, è facilmente manipolabile, in quanto basterebbe creare tante pagine senza importanza che hanno come outlinks i. Per rendere questo criterio più stabile, si modifica in

"una pagina i è più importante più sono importanti gli inlinks".

Questo significa che più sono i collegamenti importanti ad una specifica pagina, più aumenta la probabilità che essa venga aperta.

Sia i una pagina Web. Il pagerank di i è uguale a

$$r_i = \sum_{j \in I_i} \frac{r_j}{N_j}, \tag{3.1}$$

dove I_i è l'insieme degli inlinks di i e N_j è il numero di outlinks di j.
Si definisce, inoltre, $\mathbf{r}$ il vettore contenente i pagerank di tutte le pagine.

Con questo nuovo criterio, quindi, diviene più rilevante essere puntati da poche pagine con un pagerank molto alto, piuttosto che essere puntati da molte pagine di poca importanza.
Nel seguito studieremo il pagerank andando a riformulare (3.1) come un problema di autovalori e autovettori per una matrice che rappresenta la rete associata ad Internet.

Definizione 3.1. La matrice $\mathbf{P}$ che rappresenta la rete associata ad Internet si definisce nel seguente modo

$$P_{i,j} = \begin{cases} \frac{1}{N_j} & \text{se } j \text{ punta ad } i, \\ 0 & \text{altrimenti.} \end{cases} \tag{3.2}$$

In questa matrice le righe e le colonne indicano rispettivamente gli inlinks e gli outlinks. Precisamente, se la pagina i ha come inlinks le pagine j e k e come outlinks le pagine k e h, in questa matrice, la riga i-esima avrà elementi non zero nelle posizioni j e k e la colonna i-esima avrà elementi non zero nelle posizioni k e h.
La matrice $\mathbf{P}^T$ è chiamata *matrice di transizione di probabilità* e l'elemento $P^T_{i,j}$ indica la probabilità di passare dalla pagina j alla pagina i.

Esempio 3.2. Riprendiamo il grafo dell'esempio 3.1 e andiamo a vedere la matrice corrispondente

$$\mathbf{P} = \begin{pmatrix} 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{3} & 0 \end{pmatrix}.$$

Notiamo che la pagina 4 non ha outlinks, infatti la colonna corrispondente ha tutti 0.

La definizione (3.1) corrisponde al prodotto scalare tra la riga i e il vettore $\mathbf{r}$. Questo si traduce in forma matriciale,

$$\lambda \mathbf{r} = \mathbf{P}\mathbf{r}, \qquad \text{con } \lambda = 1, \tag{3.3}$$

dove $\mathbf{r}$ è autovettore di $\mathbf{P}$ associato all'autovalore $\lambda = 1$.
Calcolare il vettore di pagerank, quindi, equivale a determinare l'autovettore della matrice $\mathbf{P}$ relativo all'autovalore 1.

Esistono vari metodi efficienti per calcolare l'autovettore di una matrice, ma data la grande dimensione di Internet, il metodo più semplice e meno dispendioso per calcolare l'autovettore di pagerank $\mathbf{r}$ è il *metodo delle potenze*.
Prima di applicare questo metodo, però, facciamo alcune considerazioni.

3.2 Analisi del pagerank

Perchè il pagerank sia ben definito, come specificato nell'equazione (3.3), occorre che l'autovalore λ sia necessariamente uguale a 1, ma, per adesso, non abbiamo la certezza che la matrice $\mathbf{P}$ abbia suddetto autovalore. Inoltre, sempre per poter affermare che la definizione di pagerank è ben posta, bisogna dimostrare che l'autovettore $\mathbf{r}$ è unico.

Prima di affrontare questi due aspetti, c'è bisogno di fare un'ulteriore osservazione.
Un ipotetico utente, durante la navigazione in Internet, può trovarsi bloccato in una pagina che non ha nessun outlink. Come teorica passeggiata aleatoria questa cosa non deve succedere, ossia, nel grafo di hyperlinks non devono esserci nodi che non puntano a nulla. Questi nodi vengono chiamati *dangling nodes*, e corrispondono a una colonna di zeri nella matrice $\mathbf{P}$ (si veda l'Esempio 3.2).
Per risolvere questo problema, il modello viene modificato aggiungendo un valore costante al posto degli zeri.
Definiamo a tal proposito il vettore $\mathbf{v}$, dove

$$v_j = \begin{cases} 1 & \text{se } N_j = 0, \\ 0 & \text{altrimenti.} \end{cases}$$

La matrice verrà modificata come

$$\widetilde{\mathbf{P}} = \mathbf{P} + \mathbf{w}\mathbf{v}^T, \tag{3.4}$$

dove il vettore $\mathbf{w}$ è detto *vettore di personalizzazione* ed è tale che $\mathbf{w} > 0$ e $\|\mathbf{w}\|_1 = 1$. Questo vettore, a seconda di come viene scelto, influenza l'importanza, ossia rende la ricerca orientata verso determinati tipi di pagine Web. Di solito si sceglie di distribuire in maniera uniforme i collegamenti ipertestuali dai dangling nodes a tutte le pagine Web. Per questo viene preso

$$\mathbf{w} = \frac{1}{n}\mathbf{e}, \tag{3.5}$$

con n =numero totale delle pagine Web, ed $\mathbf{e} = (1, 1, \ldots, 1)^T \in \mathbb{R}^n$.
La matrice (3.4) sarà

$$\widetilde{\mathbf{P}} = \mathbf{P} + \frac{1}{n}\mathbf{e}\mathbf{v}^T. \tag{3.6}$$

Con questa modifica, oltre a risolvere il problema dei dangling nodes, ci siamo messi nella condizione di avere autovalore uguale a 1. Infatti, la matrice modificata $\widetilde{\mathbf{P}}$ è *stocastica per colonna*, ossia ha elementi non negativi e la somma degli elementi di ogni colonna è uguale a 1. Quanto appena detto può essere riformulato nel modo seguente.

Proposizione 3.1. *Sia* $\mathbf{A}$ *una matrice stocastica per colonna, allora*

$$\mathbf{e}^T = \mathbf{A}\mathbf{e}^T,$$

dove $\mathbf{e} = (1, 1, \ldots, 1)^T$.

Grazie a questa proposizione, quindi, abbiamo la certezza che la matrice (3.6) ha autovalore uguale a 1.

Esempio 3.3. Con la modifica (3.6) la matrice dell'Esempio 3.2 viene modificata come

$$\widetilde{\mathbf{P}} = \begin{pmatrix} 0 & \frac{1}{3} & 0 & \frac{1}{6} & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{6} & \frac{1}{3} & 0 \\ 0 & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{3} & 0 & \frac{1}{6} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & 0 & \frac{1}{6} & 0 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{6} & \frac{1}{3} & 0 \end{pmatrix}.$$

Si noti che l'unica modifica apportata è nella colonna 4, dove tutti gli zeri sono stati sostituiti dal valore costante $\frac{1}{6}$.

Vedremo nel paragrafo successivo che, per $\mathbf{P}$ di grandi dimensioni, che é il caso della matrice di Internet, $\widetilde{\mathbf{P}}$ non viene mai formata esplicitamente, in quanto memorizzarla risulterebbe molto consistente.
Ci manca da verificare l'unicità dell'autovettore associato all'autovalore 1.
Per garantire ciò ci viene in aiuto una variante del Teorema di *Perron-Frobenius*.
Prima di introdurre questo teorema, riprendendo la Definizione 1.7 di matrice irriducibile e la Definizione 1.18 di grafo fortemente connesso, riportiamo un altro utile teorema:

Teorema 3.2. *Una matrice* $\mathbf{A}$ *è irriducibile se e solo se il suo grafo diretto è fortemente connesso.*

Presentiamo ora il Teorema di *Perron-Frobenius*:

Teorema 3.3. *Sia* $\mathbf{A}$ *una matrice irriducibile, stocastica per colonna. Allora l'autovalore dominante* λ_1 *è uguale a 1 ed esiste un unico autovettore associato* $\mathbf{r}$ *tale che* $\mathbf{r} > 0$ *e* $\|\mathbf{r}\|_1 = 1$.
Inoltre, se $\mathbf{A} > 0$ *allora* $|\lambda_i| < 1,\ i = 2, 3, \ldots, n$.

La matrice di collegamento $\widetilde{\mathbf{P}}$, data la grande dimensione di Internet, siamo sicuri essere riducibile, quindi l'autovettore pagerank continua a non essere ben definito. Dobbiamo, allora, fare un'ulteriore modifica alla matrice. Per garantire l'irriducibilità, la matrice $\widetilde{\mathbf{P}}$ viene così trasformata:

$$\mathbf{Q} = \alpha\widetilde{\mathbf{P}} + (1-\alpha)\,\frac{1}{n}\mathbf{e}\mathbf{e}^T, \tag{3.7}$$

con $\alpha,\ 0 \leq \alpha \leq 1$.
$\mathbf{Q}$ è chiamata *matrice di Google*. In questo modello, α corrisponde alla probabilità che un un navigatore Web passi da una pagina all'altra seguendo il collegamento ipertestuale della struttura del Web. Il valore $(1-\alpha)$, invece, è la probabilità che egli apra una pagina arbitraria. In particolare, scegliendo il vettore di personalizzazione $\mathbf{w} = (1/n)\,\mathbf{e}$, $(1-\alpha)/n$ indica la probabilità che il navigatore scelga di aprire una pagina casuale.
Verifichiamo ora che la matrice $\mathbf{Q}$ è ancora stocastica per colonna:

$$\begin{aligned}\mathbf{e}^T\mathbf{Q} &= \mathbf{e}^T\left(\alpha\widetilde{\mathbf{P}} + (1-\alpha)\,\frac{1}{n}\mathbf{e}\mathbf{e}^T\right) = \alpha\mathbf{e}^T\widetilde{\mathbf{P}} + (1-\alpha)\,\frac{1}{n}\mathbf{e}^T\mathbf{e}\mathbf{e}^T = \\ &= \alpha\mathbf{e}^T + (1-\alpha)\,\mathbf{e}^T = \mathbf{e}^T.\end{aligned}$$

Con la modifica (3.7) abbiamo, quindi, che la matrice è sia irriducibile che stocastica per colonna. Grazie al Teorema 3.3, abbiamo ora la certezza che il vettore pagerank è ben definito.

Studiamo gli autovalori, andando a vedere come sono cambiati con la modifica (3.7).

Teorema 3.4. *Siano* $\{1, \lambda_2, \lambda_3, \ldots, \lambda_n\}$ *gli autovalori della matrice* $\widetilde{\mathbf{P}}$*. Allora gli autovalori di* $\mathbf{Q}$ *sono* $\{1, \alpha\lambda_2, \alpha\lambda_3, \ldots, \alpha\lambda_n\}$*.*

Dimostrazione. Poniamo $\hat{\mathbf{e}} = \frac{1}{\sqrt{n}}\mathbf{e}$ e prendiamo $\mathbf{U}_1 \in \mathbb{R}^{n\times(n-1)}$ tale che $\mathbf{U} = (\hat{e}, \mathbf{U}_1)$ è ortogonale. Ricordiamo che $\widetilde{\mathbf{P}}$ è stocastica per colonna, allora vale $\hat{\mathbf{e}}^T\widetilde{\mathbf{P}} = \hat{\mathbf{e}}^T$. Per cui

$$\mathbf{U}^T\widetilde{\mathbf{P}}\mathbf{U} = \begin{pmatrix} \hat{\mathbf{e}}^T\widetilde{\mathbf{P}} \\ \mathbf{U}_1^T\widetilde{\mathbf{P}} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{e}} & \mathbf{U}_1 \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{e}}^T\hat{\mathbf{e}} & \hat{\mathbf{e}}^T\mathbf{U}_1 \\ \mathbf{U}_1^T\widetilde{\mathbf{P}}\hat{\mathbf{e}} & \mathbf{U}_1^T\widetilde{\mathbf{P}}\mathbf{U}_1 \end{pmatrix} = \begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{s} & \mathbf{T} \end{pmatrix}, \tag{3.8}$$

dove $\mathbf{s} \in \mathbb{R}^{(n-1)\times 1}$, $\mathbf{s} = \mathbf{U}_1^T\widetilde{\mathbf{P}}\hat{\mathbf{e}}$ e $\mathbf{T} \in \mathbb{R}^{(n-1)\times(n-1)}$, $\mathbf{T} = \mathbf{U}_1^T\widetilde{\mathbf{P}}\mathbf{U}_1$. Poichè abbiamo una trasformazione di similarità, la matrice $\mathbf{T}$ ha come autovalori $\{\lambda_2, \lambda_3, \ldots, \lambda_n\}$.
Inoltre, definendo $\mathbf{w} := \frac{1}{n}\mathbf{e}$ abbiamo

$$\mathbf{U}^T\mathbf{w} = \begin{pmatrix} \frac{1}{\sqrt{n}}\mathbf{e}^T\mathbf{w} \\ \mathbf{U}_1^T\mathbf{w} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{n}} \\ \mathbf{U}_1^T\mathbf{w} \end{pmatrix}. \tag{3.9}$$

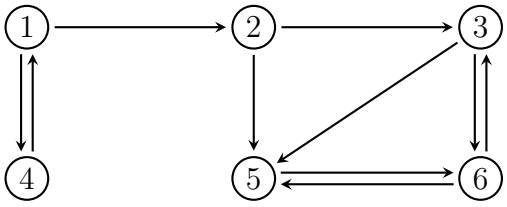
Per cui, considerando (3.8) e (3.9) si ha

$$\begin{aligned} \mathbf{U}^T\mathbf{Q}\mathbf{U} &= \mathbf{U}^T\left(\alpha\widetilde{\mathbf{P}} + (1-\alpha)\,\mathbf{w}\mathbf{e}^T\right)\mathbf{U} = \alpha\mathbf{U}^T\widetilde{\mathbf{P}}\mathbf{U} + (1-\alpha)\,\mathbf{U}^T\mathbf{w}\mathbf{e}^T\mathbf{U} \\ &= \alpha\begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{s} & \mathbf{T} \end{pmatrix} + (1-\alpha)\begin{pmatrix} \frac{1}{\sqrt{n}} \\ \mathbf{U}_1^T\mathbf{w} \end{pmatrix}\begin{pmatrix} \sqrt{n} & \mathbf{0} \end{pmatrix} \\ &= \alpha\begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{s} & \mathbf{T} \end{pmatrix} + (1-\alpha)\begin{pmatrix} 1 & \mathbf{0} \\ \sqrt{n}\mathbf{U}_1^T\mathbf{w} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{s}_1 & \alpha\mathbf{T} \end{pmatrix}. \end{aligned}$$

□

Il Teorema 3.4 implica che se $\widetilde{\mathbf{P}}$ ha un autovalore multiplo uguale a 1, che è effettivamente il caso per la matrice di Internet, il secondo autovalore più grande di $\mathbf{Q}$ è sempre uguale ad α.
Vediamo quanto appena detto con un esempio.

Esempio 3.4. Dato il seguente grafo non fortemente connesso



la matrice corrispondente è:

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & 0 & 1 & 0 \end{pmatrix}.$$

È una matrice stocastica per colonna, ma riducibile.
Per renderla irriducibile andiamo ad usare la (3.7) scegliendo $\alpha = 0.85$.
Calcoliamo gli autovalori e autovettori con il seguente codice MATLAB:

```
Aut_A=eig(A)';
e=ones(6,1);
alpha=0.85;
Q=alpha*A+(1-alpha)/6*e*e';
[l,L]=eig(Q);
```

Il risultato è il seguente:

```
Autovalori di A:
-0.5000    1.0000   -0.5000    1.0000   -1.0000         0
```

```
Autovalori di Q:
 1.0000    0.8500   -0.0000   -0.8500   -0.4250   -0.4250

Autovettori di Q:
 0.4468   -0.3651    0.3536   -0.0000   -0.8165   -0.0339
 0.4297   -0.3651   -0.3536    0.0000    0.4082    0.7234
 0.4297   -0.3651   -0.3536   -0.0000    0.4082   -0.6896
 0.0572   -0.0000    0.7071    0.0000   -0.0000    0.0000
 0.4690    0.5477    0.0000    0.7071   -0.0000    0.0000
 0.4559    0.5477   -0.3536   -0.7071    0.0000    0.0000
```

Come indicato dal Teorema 3.3, l'autovalore dominante di **Q** è uguale a 1 e l'autovettore associato è l'unico con tutte componenti non negative.
Inoltre, anche se **A** ha 1 come autovalore multiplo, **Q** ha autovalore 1 semplice, quindi come secondo autovalore più grande α.

3.3 Applicazione del Metodo delle Potenze

Come già introdotto alla fine del paragrafo 3.1, la dimensione della matrice del Web è enorme, quindi per calcolare l'autovettore pagerank non conviene assolutamente usare uno dei metodi standard, quali ad esempio l'iterazione QR.
Un semplice metodo ritenuto efficace è il *metodo delle potenze*, il quale memorizza un numero molto basso di vettori rispetto ad altri metodi più potenti, e questo risulta essere molto importante, a livello di costi, per problemi di così grandi dimensioni. Vedremo che il metodo descritto nel Capitolo 2, dato il contesto, viene un po' modificato.

Supponiamo che **Q** sia diagonalizzabile. Nel nostro caso non c'è bisogno di calcolare l'approssimazione dell'autovalore massimo, poichè sappiamo già con certezza essere uguale a 1.
Proprio per questo, la velocità di convergenza è determinata da $|\lambda_2|$, il quale se è vicino a 1 comporta un'iterazione molto lenta, ma fortunatamente non è il caso della matrice di Google.
Nell'usuale metodo delle potenze, ad ogni iterazione il vettore viene normalizzato per evitare underflow o overflow. Dato che qui ci troviamo a che fare con una matrice stocastica per colonna, questo non è necessario. Vale infatti il seguente risultato:

Proposizione 3.5. *Sia* $\mathbf{x}$ *un vettore tale che* $\|\mathbf{x}\|_1 = \mathbf{e}^T\mathbf{x} = 1$*, e sia* $\mathbf{A}$ *una matrice stocastica per colonna. Allora*

$$\|\mathbf{Ax}\|_1 = 1.$$

Dimostrazione. Poniamo $\mathbf{y} = \mathbf{Ax}$. $\mathbf{A}$ per ipotesi è stocastica per colonna, allora vale $\mathbf{e}^T\mathbf{A} = \mathbf{e}^T$. Per cui abbiamo,

$$\|\mathbf{y}\|_1 = \mathbf{e}^T\mathbf{y} = \mathbf{e}^T\mathbf{Ax} = \mathbf{e}^T\mathbf{x} = 1. \qquad \square$$

Questa proprietà permette, quindi, di non fare la normalizzazione in modo esplicito, così da tenere basso il costo computazionale.
Un altro aspetto da tenere in considerazione, causa ancora l'enorme dimensione della matrice con cui ci troviamo a lavorare, è il prodotto matrice per vettore che diventa non banale.
Ricordiamo che la matrice che viene moltiplicata per il vettore è
$\mathbf{Q} = \alpha\widetilde{\mathbf{P}} + (1-\alpha)\,\frac{1}{n}\mathbf{e}\mathbf{e}^T$, dove $\widetilde{\mathbf{P}} = \mathbf{P} + \frac{1}{n}\mathbf{e}\mathbf{v}^T$.
Questo vuol dire che per formare $\widetilde{\mathbf{P}}$ vengono inseriti un gran numero di vettori pieni in $\mathbf{P}$, di conseguenza, memorizzare $\widetilde{\mathbf{P}}$ in modo esplicito diventa molto pesante. Studiamo nel dettaglio il prodotto metrice-vettore:

$$\begin{aligned}\mathbf{y} = \mathbf{Qx} &= \left(\alpha\widetilde{\mathbf{P}} + (1-\alpha)\,\frac{1}{n}\mathbf{e}\mathbf{e}^T\right)\mathbf{x} \\ &= \alpha\left(\mathbf{P} + \frac{1}{n}\mathbf{e}\mathbf{v}^T\right)\mathbf{x} + (1-\alpha)\,\frac{1}{n}\mathbf{e}\left(\mathbf{e}^T\mathbf{x}\right) = \alpha\mathbf{Px} + \delta\frac{1}{n}\mathbf{e},\end{aligned} \tag{3.10}$$

dove $\delta = \alpha\mathbf{v}^T\mathbf{x} + (1-\alpha)\,\mathbf{e}^T\mathbf{x}$.

È possibile migliorare ulteriormente il prodotto matrice-vettore, senza dover controllare quali pagine sono senza outlinks, senza quindi memorizzare il vettore $\mathbf{v}$.
Dato che, per la Proposizione 3.5, vale $\|\mathbf{y}\|_1 = \mathbf{e}^T\mathbf{y} = 1$, se calcoliamo la norma-1 dell'equazione (3.10), ossia moltiplichiamo a destra e a sinistra dell'uguale per $\mathbf{e}^T$, si ha:

$$1 = \mathbf{e}^T\left(\alpha\mathbf{Px}\right) + \delta\mathbf{e}^T\left(\frac{1}{n}\mathbf{e}\right) = \mathbf{e}^T\left(\alpha\mathbf{Px}\right) + \delta,$$

allora

$$\delta = 1 - \|\alpha \mathbf{Q}\mathbf{x}\|_1. \tag{3.11}$$

Ora abbiamo tutto il materiale per poter scrivere l'algoritmo PageRank.
Essendo $\mathbf{P}$ una matrice a coefficienti reali e l'autovalore dominante reale, precisamente $\lambda = 1$, per la Proposizione 1.2 possiamo scegliere il vettore arbitrario iniziale $\mathbf{x}^{(0)}$ reale, pertanto:

Fissato $\mathbf{x}^{(0)} \in \mathbb{R}^n$ di norma-1 unitaria,
fissato α (per es. $\alpha = 0.85$),
e fissato il vettore di personalizzazione $\mathbf{w}$

Per $k = 0, 1, 2, \ldots$, fino a convergenza

$$\begin{aligned}
\mathbf{y}^{(k+1)} &= \alpha \mathbf{P}\mathbf{x}^{(k)} \\
\delta^{(k+1)} &= 1 - \|\mathbf{y}^{(k+1)}\|_1 \\
\mathbf{y}^{(k+1)} &= \mathbf{y}^{(k+1)} + \delta^{(k+1)}\mathbf{w} \\
\mathbf{x}^{(k+1)} &= \mathbf{y}^{(k+1)}/\|\mathbf{y}^{(k+1)}\|_1
\end{aligned}$$

Da notare che nell'assegnazione $\mathbf{x}^{(k+1)} = \mathbf{y}^{(k+1)}/\|\mathbf{y}^{(k+1)}\|_1$ abbiamo lasciato la normalizzazione, nonostante nella Proposizione 3.5 avessimo dimostrato che $\|\mathbf{y}\|_1 = 1$. Questo perchè nell'ambito dell'aritmetica floating point, una normalizzazione potrebbe essere utile, in quanto gli errori di arrotondamento potrebbero togliere accuratezza.
Come criterio di arresto usiamo il residuo scalare al passo k,

$$res^{(k+1)} = \|\mathbf{y}^{(k+1)} - \mathbf{x}^{(k+1)}\|_1.$$

3.4 Implementazione

In questa sezione proponiamo un'implementazione di quanto descritto in precedenza.
Usiamo una funzione "loadMatrix.m", qui riportata per completezza, che carica e crea la matrice di hyperlinks delle università di Stanford e di Berkeley.
Andiamo a descrivere nel dettaglio cosa fa questa funzione.
Usando il comando

```
load stanford-berkeley-bool-sorted.dat;
```

viene caricata la matrice dei links delle università. Questa matrice è a coefficienti reali di dimensione $h \times 3$, dove h è il numero totale dei link. In essa le prime due colonne identificano le pagine Web, la colonna 3, invece, è composta da tutti 1 che stanno a indicare che c'è un link tra le pagine della colonna 1 e quelle della colonna 2.

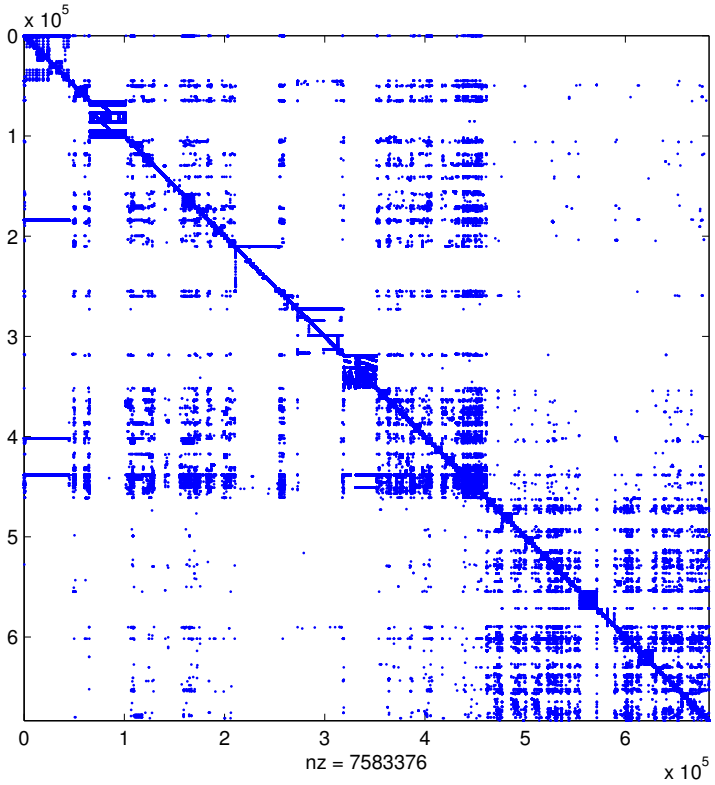


Figura 3.1: Matrice di hyperlinks Berkeley-Stanford.

Successivamente, con il comando

```
A=spconvert(stanford_berkeley_bool_sorted);
```

viene creata la seguente matrice quadrata sparsa:

$$A_{i,j} = \begin{cases} m & \text{se } i \text{ punta a } j, \\ 0 & \text{altrimenti,} \end{cases}$$

dove m è il numero totale di link da i a j. La funzione "loadMatrix.m", per essere precisi, prende in considerazione meno pagine rispetto a quelle esistenti, esattamente 683446.
All'interno della suddetta funzione, viene poi chiamata una funzione "normalize.m", la quale utilizzando la norma-1, normalizza ogni riga della matrice trasposta, ottenendo come risultato la matrice di hyperlinks (3.2).
Il risultato è una matrice 683446×683446 con 7583376 elementi non zero, rappresentata in Figura 3.1.
Per i codici MATLAB delle due funzioni, che chiamano la matrice e la normalizzano, si guardi Appendice A.

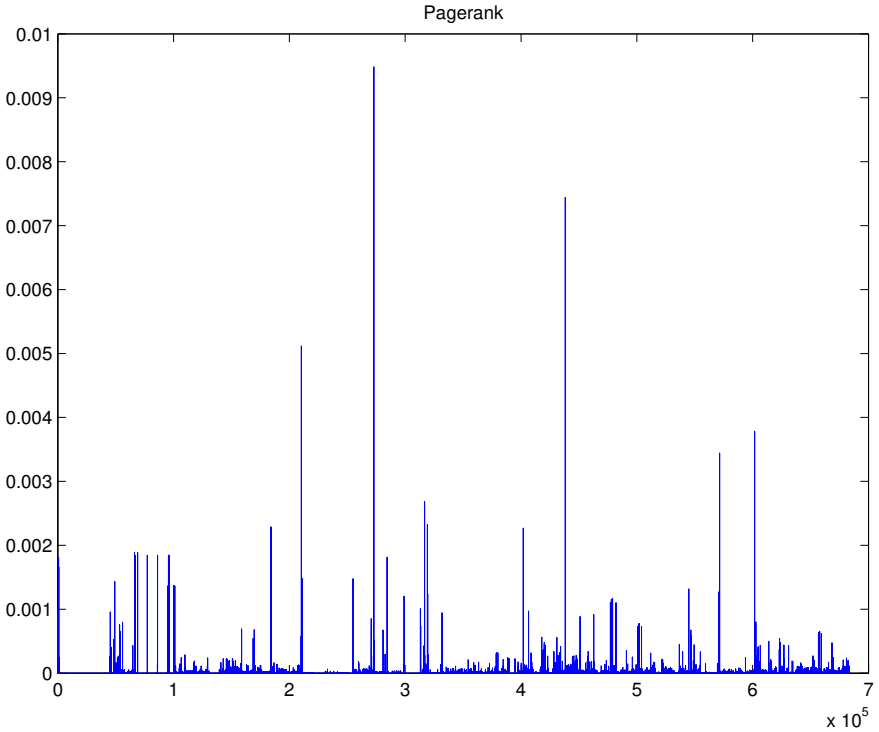


Figura 3.2: Il vettore di pagerank delle pagine di Berkeley-Stanford.

Dalla matrice di hyperlinks, calcoliamo l'autovettore $\mathbf{r}$ utilizzando l'algoritmo PageRank descritto nell'Appendice B.
Per il calcolo del vettore di pagerank, rappresentato in Figura 3.2, si è scelto $\alpha = 0.85$ e $\mathbf{w} = (1/n)\mathbf{e}$.

Per il test d'arresto, basato sul residuo scalare, si è fissata una tolleranza di 10^{-8}.

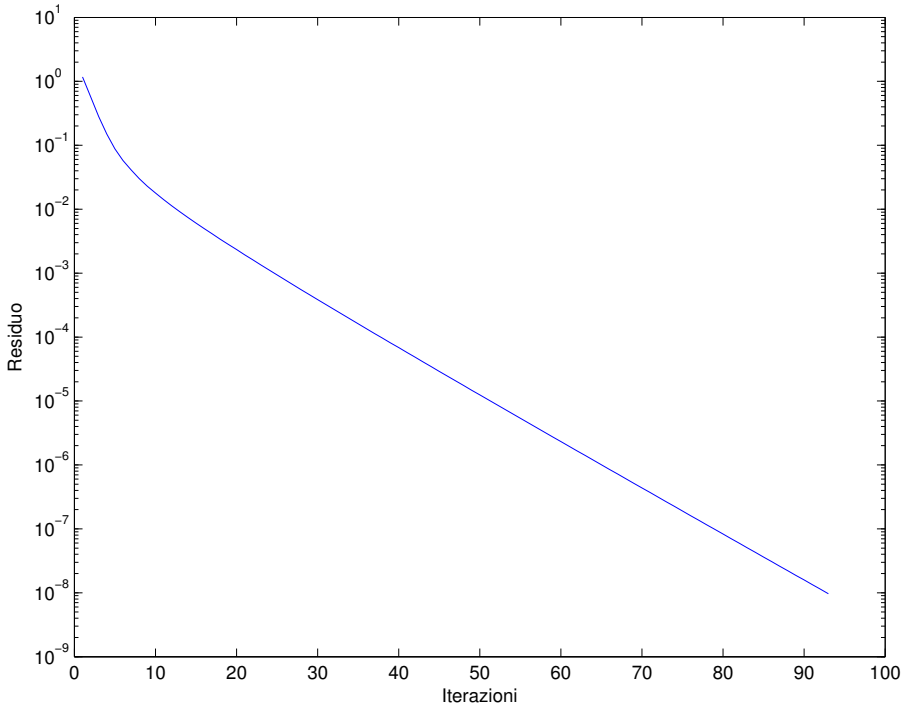


Figura 3.3: Andamento del residuo scalare del metodo delle potenze, per il calcolo di pagerank della matrice Berkeley-Stanford, fino a convergenza.

Come si può vedere in Figura 3.3, l'algoritmo converge dopo 93 iterazioni, quando la norma-1 del residuo risulta essere inferiore alla tolleranza fissata. Come detto, inoltre, nel Capitolo 2, la convergenza è tanto più rapida quanto più piccolo è il rapporto degli autovalori, $|\lambda_2/\lambda_1|$. Nel caso della matrice di Internet, si ha $|\lambda_2| \leq \alpha$ e $\lambda_1 = 1$, allora, tutto dipende dal valore di α.
La Tabella 3.1 e la Figura 3.4 riportano uno studio della convergenza al variare di α. Precisamente, la tabella riporta il numero di iterazioni del metodo delle potenze, più α è piccolo più diminuisce il numero di iterazioni per trovare l'autovettore di pagerank. La figura, invece, rappresenta l'errore del metodo delle potenze all'aumentare dell'iterazione k, che, grazie al Teorema 2.1 sappiamo si comporta come $\mathcal{O}\left(|\alpha|^k\right)$.
Quindi la convergenza sembrerebbe migliore al diminuire di alfa. D'altra par-

te, α piccolo comporta maggiore probabilità di aprire pagine nuove, piuttosto che seguire i collegamenti ipertestuali della struttura Web. Questo implica valori di pagerank completamente diversi. Il valore $\alpha = 0.85$ sembra essere il giusto compromesso.

Tabella 3.1: Numero di iterazioni del metodo delle potenze al variare di α.

α	n. iterazioni
0.75	54
0.85	93
0.95	289

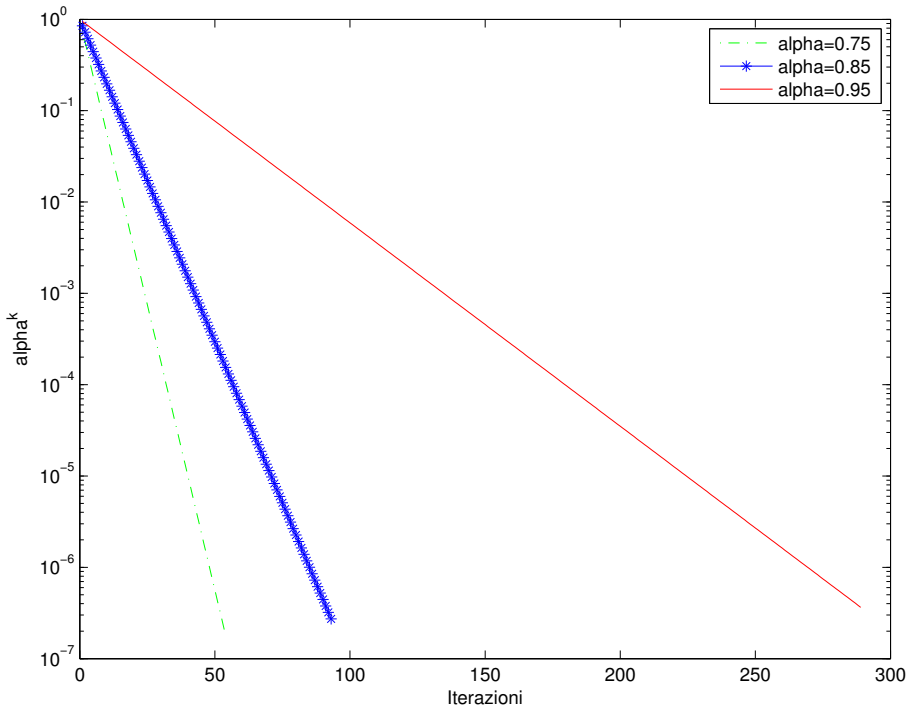


Figura 3.4: Errore del metodo delle potenze al variare di α

Conclusioni

In questo elaborato si è analizzato come l'algoritmo PageRank, usato da Google, assegna un valore di importanza (pagerank) alle pagine Web.
La filosofia adottata da Larry Page e Sergey Brin, fondatori di Google e creatori di PageRank, è la seguente:

"L'importanza di una pagina Web è giudicata dal numero di pagine a essa collegate così come dalla loro importanza".

Abbiamo presentato il metodo delle potenze, metodo usato dall'algoritmo per determinare il vettore di pagerank. Sebbene questo metodo sia semplice, è di grande utilità per questa applicazione, in quanto le dimensioni della rete Internet, dell'ordine di miliardi, non permettono l'uso di algoritmi più efficienti.
Si è riportato che Google considera il Web come un enorme grafo al quale è associato una matrice, detta matrice di hyperlinks, e abbiamo constatato che il vettore di pagerank altro non è che l'autovettore della matrice associato all'autovalore 1. Abbiamo visto come adattare questa matrice in modo da renderla irriducibile e stocastica per colonna, così da garantire l'esistenza e l'unicità rispettivamente dei suddetti autovalore e autovettore.
Considerando questo e alcune proprietà che ne derivano, si è leggermente variato il metodo delle potenze che avevamo descritto in maniera generale e abbiamo assemblato l'algoritmo PageRank.
Alla fine di questa tesi abbiamo implementato l'algoritmo su un sottoinsieme di Internet, per la precisione sulla rete dei links delle Università di Stanford e Barkeley. Questo naturalmente è solo un piccolo esempio rispetto ai collegamenti ipertestuali e al numero di pagine Web realmente presi in considerazione. Nella pratica, infatti, un calcolo pagerank può richiedere diversi giorni.
I dettagli sull'effettiva implementazione del modello usato da Google per ordinare le pagine Web è coperto dal segreto industriale e non ci è dato di conoscere. Quello che sappiamo è che, sicuramente, nella classificazione delle pagine, oltre al pagerank, intervengono altri fattori, come area geografica,

attualità dei contenuti.
Attualmente la ricerca sui modelli di PageRank è molto attiva, si studiano diversi miglioramenti per la procedura di iterazione, alcuni dei quali sono arrivati a segnalare un'accellerazione fino al 30%.

Appendice A

Qui di seguito, le funzioni *loadMatrix* e *normalize*, usate per caricare la matrice di hyperlinks.

loadMatrix.m

```
% loadMatrix carica matrice e chiama la funzione che la normalizza
%
% OUTPUT
% A: Matrice di hyperlinks 683446x683446
%
function [A]=loadMatrix;

truesize=683446;
load stanford-berkeley-bool-sorted.dat;
A=spconvert(stanford_berkeley_bool_sorted);

% riduciamo la dimensione
n=max(size(A));
A(n,n)=0;
A=A(1:truesize,1:truesize);

% chiamiamo la funzione che normalizza ogni riga
A=normalize(A,1);
A=A';
```

normalize.m

```
% normalize normalizza ogni riga di una matrice
%
% INPUT
% A: matrice da normalizzare
% lvalue: norma con cui normalizzare.
% Se come input si ha un solo argomento, si pone lvalue=2
%
% OUTPUT
% A: matrice normalizzata
%
function A=normalize(A,lvalue)

if nargin < 2
        lvalue=2;
end
[m,n]=size(A);

% viene trasposta per velocizzare, vengono normalizzate le colonne
A=A';
for i=1:m,
        nrm=norm(A(:,i),lvalue);
        if nrm~=0
                A(:,i)=A(:,i)/nrm;
        else
                A(:,i)=A(:,i);
        end
end
A=A';
```

Appendice B

In questa Appendice è trascritto l'algoritmo *PageRank* e successivamente lo script per la sua esecuzione.

PageRank.m

```
% PageRank calcola l'autovettore di pagerank con il metodo delle potenze
%
% INPUT
% P: matrice di hyperlinks di partenza
% x0: vettore arbitrario iniziale
% maxit: massimo di iterazioni
% tol: tolleranza per criterio di arresto residuo
% alpha: compreso tra 0 e 1
% w: vettore di personalizzazione
%
% OUTPUT
% x: autovettore di pagerank
% it: numero di iterazioni fatte
% res: vettore dei residui
%
function [x,it,res]=PageRank(P,x0,maxit,tol,alpha,w);

y=x0/norm(x0,1);
it=0;
res=norm(y,1);
while (res>tol & it<maxit)
    it=it+1;
    x=y/norm(y,1);

    % calcolo matrice per vettore
    y=alpha*P*x;
```

```
    beta=1-norm(y,1);
    y=y+beta*w;

    res(it)=norm(y-x,1);
end
semilogy(res)
xlabel('Iterazioni')
ylabel('Residuo')
figure
plot(y)
title('Pagerank')
```

script.m

```
% script per mandare la funzione PageRank della matrice di hyperlinks
%
% fissiamo i dati da dare poi in input
%
P=loadMatrix;
n=max(size(P));
e=ones(n,1);
x0=rand(n,1);
maxit=500;
tol=1e-8;
alpha=0.85;
w=(1/n)*e;

% chiamiamo la funzione
[r,it,res]=PageRank(P,x0,maxit,tol,alpha,w)
```

Bibliografia

[1] L. Eldèn, *Matrix Methods in Data Mining and Pattern Recognition*, SIAM, 2007.

[2] Amy N. Langville, Carl D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press, 2006.

[3] A. Quarteroni, R. Sacco, F. Saleri, *Matematica Numerica*, Springer, 2008.