

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

---

---

Scuola di scienze  
Corso di Laurea in Matematica

## Non backtracking PageRank

Relatrice:  
Chiar.ma Prof.ssa  
Valeria Simoncini

Presentata da:  
Giovanni Acampa

II Sessione  
Anno Accademico 2021-2022

*“This is the way the world ends  
Not with a bang but a whimper.”*

# Indice

<b>1</b>	<b>PageRank</b>	<b>5</b>
1.1	Matrici dei dati . . . . .	5
1.2	Il vettore del PageRank . . . . .	7
1.3	PseudoPageRank . . . . .	8
<b>2</b>	<b>Non backtracking PageRank</b>	<b>10</b>
2.1	Edge PageRank . . . . .	10
2.2	Non backtracking PageRank . . . . .	14
2.3	Analogie tra i due algoritmi . . . . .	16
<b>3</b>	<b>GMRES</b>	<b>20</b>
3.1	I metodi di proiezione . . . . .	20
3.1.1	Il metodo di Arnoldi . . . . .	20
3.2	GMRES . . . . .	22
3.2.1	Criterio d'arresto . . . . .	23
3.2.2	GMRES con restart . . . . .	26
3.3	Precondizionamento di GMRES . . . . .	26
3.4	Fattorizzazione ILU . . . . .	28
<b>4</b>	<b>Esperimenti numerici</b>	<b>29</b>
4.1	Reti stradali . . . . .	29
4.2	Precondizionamento . . . . .	32

# Introduzione

In questo lavoro analizzeremo il non backtracking PageRank, una variante del PageRank che differisce dall'algoritmo originale poiché considera solo i cammini non backtracking, cioè quelli che non tornano nel nodo da cui sono partiti al passo subito successivo. Il PageRank è un algoritmo di classificazione nato per mettere in ordine di rilevanza i siti web e oggi applicato ai più svariati ambiti. Lo scopo di questa variante è ottenere una classificazione migliore in tutti quei problemi in cui il backtracking viene evitato. Vedremo infatti alcuni esperimenti effettuati sulle reti stradali di alcune città, nelle quali possiamo supporre che il non backtracking PageRank abbia senso: è infatti molto raro che un mezzo di trasporto arrivi a un incrocio e torni subito indietro nella strada dalla quale stava arrivando. Prima di effettuare gli esperimenti, daremo anche grande importanza a come risolvere computazionalmente il problema: l'algoritmo richiede di risolvere un grande sistema lineare con una matrice sparsa. Studieremo quindi il risolutore di sistemi lineari GMRES e faremo delle considerazioni su come si raggiunga la convergenza e su come si possano ridurre il numero di iterazioni e il tempo di calcolo.

# Prerequisiti

Ricordiamo qualche utile prerequisito che daremo per noto in questo lavoro:

**Definizione 0.1** (Grafo). Un grafo è una coppia ordinata  $G = (V, E)$  di insiemi, tali che gli elementi di  $E$  siano coppie di elementi di  $V$ . Gli elementi di  $V$  sono detti vertici del grafo, gli elementi di  $E$  sono i suoi archi.

**Definizione 0.2.** Definiamo alcuni tipi elementari di grafi:

- Un grafo si dice diretto se i suoi archi sono orientati, cioè se gli elementi di  $E$  sono coppie ordinate.
- Un grafo si dice non pesato se a ciascuno dei suoi archi si può assegnare 1 come valore.
- Un percorso in  $G$  è dato da una sequenza di vertici  $v_0, \dots, v_n$  e da una sequenza di archi  $(v_0, v_1), \dots, (v_{n-1}, v_n)$  che li collegano.
- Un grafo si dice debolmente connesso se ogni coppia ordinata di vertici può essere collegata tramite un percorso anche non orientato.
- Un grafo si dice  $k$ -regolare se ogni suo nodo ha esattamente  $k$  archi in entrata e  $k$  archi in uscita.

**Definizione 0.3** (Matrice stocastica). Una matrice  $A = (a_{ij}) \in \mathbb{R}^{n \times n}$  si dice stocastica per righe (o per colonne) se è ad elementi non negativi e la somma delle sue righe (o colonne) è uguale a 1:

$$\forall j = 1, \dots, n \quad a_{ij} \geq 0 \quad \text{e} \quad \sum_{i=1}^n a_{ij} = 1.$$

**Definizione 0.4** (Pseudoinversa). Sia  $A = (a_{ij}) \in \mathbb{R}^{n \times m}$ .  $A^+ \in \mathbb{R}^{m \times n}$  è detta la pseudoinversa di  $A$  se verifica le seguenti proprietà:

- $AA^+A = A$ ,
- $A^+AA^+ = A^+$ ,

- $(AA^+)^T = AA^+$ ,
- $(A^+A)^T = A^+A$ .

**Teorema 0.5.** *La pseudoinversa di una matrice esiste ed è unica.*

**Definizione 0.6** (Matrice definita positiva). La matrice  $A \in \mathbb{R}^{n \times n}$  si dice definita positiva se  $\mathbf{x}^T A \mathbf{x} > 0$  per ogni  $\mathbf{x} \neq 0$ ,  $\mathbf{x} \in \mathbb{R}^n$ .

**Definizione 0.7** (Matrice unitaria). La matrice  $U \in \mathbb{R}^{n \times n}$  si dice unitaria se vale  $U^H U = U U^H = I$  dove  $U^H$  è la matrice trasposta coniugata di  $U$  e  $I$  è la matrice identità.

**Definizione 0.8** (Matrice irriducibile). La matrice  $A \in \mathbb{R}^{n \times n}$  si dice irriducibile se esiste una matrice di permutazione  $T$  tale che:

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

con  $X, Z$  quadrate.  $A$  si dice irriducibile altrimenti.

**Proposizione 0.9** (Formula di Sherman-Morrison). Sia  $A \in \mathbb{R}^{n \times n}$  e siano  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ . Se  $A$  è non singolare e  $1 + \mathbf{v}^T A^{-1} \mathbf{u} \neq 0$ , allora:

$$(A + \mathbf{u} \mathbf{v}^T)^{-1} = A^{-1} - A^{-1} \mathbf{u} (1 + \mathbf{v}^T A^{-1} \mathbf{u})^{-1} \mathbf{v}^T A^{-1}.$$

**Definizione 0.10** (Matrice di Hessenberg superiore). La matrice  $A \in \mathbb{R}^{n \times n}$  si dice matrice di Hessenberg superiore se ha valori pari a zero sotto la prima sottodiagonale.

**Definizione 0.11** (Cerchi di Gerschgorin). Sia  $A \in \mathbb{C}^{n \times n}$ . I cerchi del piano complesso

$$\mathcal{G}_i^{(r)} := \left\{ z \in \mathbb{C} \mid |z - A_{ii}| \leq \sum_{j=1, j \neq i}^n |A_{ij}| \right\}, \quad i = 1, \dots, n,$$

di centro  $A_{ii}$  e raggio  $\sum_{j=1, j \neq i}^n |A_{ij}|$  sono detti cerchi di Gerschgorin per righe.

**Teorema 0.12** (Primo teorema di Gerschgorin). Sia  $A \in \mathbb{C}^{n \times n}$ . Allora:

$$\text{spec}(A) \subset \bigcup_{i=1}^n \mathcal{G}_i^{(r)},$$

*cioè gli autovalori di  $A$  sono contenuti nell'unione dei cerchi di Gerschgorin per righe.*

# Capitolo 1

## PageRank

Negli anni novanta, la mole di dati disponibile su internet aumentava a dismisura. Era dunque necessario un modo per cercare le informazioni sul web in maniera rapida e precisa. Proprio a questo lavorarono Larry Page e Sergey Brin nel loro celebre articolo: “The PageRank Citation Ranking: Bringing Order to the Web” [1]. I due autori crearono così il PageRank, un algoritmo per classificare le pagine web in base alla loro rilevanza rispetto alla ricerca effettuata.

La strategia scelta è sintetizzata in una frase degli stessi autori: “The importance of a page is judged by the number of pages linking to it as well as their importance”. Parafrasato significa che il primo criterio che influisce sull'importanza di una pagina è il numero di altre pagine che puntano ad essa. Questo criterio però, potrebbe essere facilmente manipolabile, ad esempio creando molte pagine che puntino a quella di cui si vuole aumentare l'importanza. Ed è per questo che esiste il secondo criterio, il quale afferma che l'importanza di una pagina è data anche dall'importanza delle pagine che puntano ad essa.

### 1.1 Matrici dei dati

Possiamo vedere il web come un grafo diretto e non pesato, nel quale i nodi sono le pagine e gli archi sono i link fra di esse. Consideriamo un modello senza archi multipli, quindi ogni pagina punta verso ciascuna altra pagina al più una volta. Possiamo definire così la relativa matrice delle adiacenze:

**Definizione 1.1** (Matrice delle adiacenze). Sia  $\tilde{G} = (V, \tilde{E})$  un grafo diretto, non pesato e senza archi multipli, con  $n$  nodi e  $\tilde{m}$  archi. La sua matrice delle adiacenze  $A \in \mathbb{R}^{n \times n}$  è la matrice dove  $a_{ij} = 1$  se esiste un arco che va dal nodo  $i$  al nodo  $j$ ,  $a_{ij} = 0$  se tale arco non esiste.

Tale matrice ha come elementi solo 0 e 1. Notiamo che se la  $i$ -esima riga è tutta fatta da 0 allora la  $i$ -esima pagina non ha nessun link in uscita; analogamente se

la  $j$ -esima colonna è tutta fatta da 0 allora la  $j$ -esima pagina non ha nessun link in entrata. Diamo la seguente utile definizione:

**Definizione 1.2** (Dangling nodes e source nodes). Un nodo di un grafo è detto dangling node se non ha archi in uscita, è detto source node se non ha archi in entrata.

Prima di procedere, è necessaria una correzione sulla matrice delle adiacenze: rimuoviamo i dangling nodes sostituendo le righe di  $A$  fatte da soli 0 con righe fatte da soli 1. Notiamo che per ogni correzione effettuata abbiamo aggiunto al grafo  $n$  archi. La nuova matrice delle adiacenze è della forma  $W = A + \boldsymbol{\chi}\mathbf{1}^T$ , dove  $\boldsymbol{\chi} \in \mathbb{R}^n$  ha come componenti  $\chi_i = 1$  se il nodo  $i$  è un dangling node,  $\chi_i = 0$  altrimenti;  $\mathbf{1} \in \mathbb{R}^n$  è il vettore con 1 in ogni componente. Chiamiamo  $G = (V, E)$  il grafo relativo a  $W$ : esso ha ancora  $n$  nodi, ma ha adesso  $m = \tilde{m} + kn$  archi, con  $k$  numero di dangling nodes di  $\tilde{G}$ .

In questa trattazione rimuoviamo i dangling nodes dal grafo per avere l'unicità del vettore del PageRank, che segue dal teorema di Perron-Frobenius [2]. Affinché valga tale teorema, però, è indispensabile che  $A$  sia irriducibile, condizione equivalente alla connessione forte del grafo. La connessione forte segue direttamente dalla rimozione dei dangling nodes.

**Teorema 1.3.** *Una matrice  $A$  è irriducibile se e solo se il suo grafo diretto è fortemente connesso.*

**Teorema 1.4** (Variante del teorema di Perron-Frobenius). *Sia  $A$  una matrice irriducibile e stocastica per colonne. Allora il suo autovalore dominante è 1 e tale autovalore è semplice. L'autovettore corrispondente ha tutte componenti positive ed è l'unico autovettore con tutte componenti non-negative.*

Osserviamo che aggiungere una riga formata da tutti 1 corrisponde ad avere uguale probabilità di passare alle altre pagine, quindi non altera la loro importanza.

Definiamo adesso la matrice sorgente e la matrice bersaglio che segnano rispettivamente da quali nodi escono gli archi e in quali nodi essi entrano:

**Definizione 1.5** (Matrice sorgente e matrice bersaglio). Ordiniamo gli  $m$  archi di  $G$  in un qualunque ordine  $\{e\}_{e=1}^m$ . Definiamo la matrice sorgente  $L \in \mathbb{R}^{m \times n}$  come:

$$L_{ei} = \begin{cases} 1 & \text{se } e \text{ è uscente dal nodo } i, \\ 0 & \text{altrimenti.} \end{cases}$$

e la matrice bersaglio  $R \in \mathbb{R}^{m \times n}$  come:

$$R_{ej} = \begin{cases} 1 & \text{se } e \text{ è entrante nel nodo } j, \\ 0 & \text{altrimenti.} \end{cases}$$

**Definizione 1.6** (Out-degree e in-degree). Definiamo il vettore degli out-degrees di una matrice come il vettore che ha per componenti gli out-degrees dei singoli nodi, cioè il numero di archi che escono da ciascun nodo. Analogamente definiamo il vettore degli in-degrees quello che ha come componenti gli in-degrees dei singoli nodi, cioè il numero di archi che entrano in ciascun nodo.

Facciamo adesso qualche osservazione. Indichiamo  $\mathbf{d} := W\mathbf{1}$  il vettore degli out-degrees di  $W$  e chiamiamo  $D$  la matrice diagonale con il vettore  $\mathbf{d}$  sulla diagonale. Grazie alla correzione effettuata per ottenere  $W$ ,  $\mathbf{d}$  ha tutte componenti positive e quindi  $D$  è invertibile. Si può inoltre verificare che  $W = L^T R$  e che  $D = L^T L$ . Infine, anche la matrice  $R^T R$  è diagonale e ha sulla diagonale il vettore con gli in-degrees di  $W$ . Osserviamo però che condizione sufficiente perché  $R^T R$  sia invertibile è che  $W \neq A$  e quindi con la correzione dei dangling nodes sono corretti in maniera collaterale anche i source nodes. Se invece  $W = A$ , non sappiamo se  $R^T R$  sia invertibile.

## 1.2 Il vettore del PageRank

Possiamo formulare ora l'algoritmo con cui trovare il vettore del PageRank. Intuitivamente esso può essere visto da due prospettive differenti: la prima è quella di un random surfer, cioè di un navigatore che si muove a caso fra le pagine del web usando i link fra di esse e che ad ogni passo ha una certa probabilità di teletrasportarsi in una pagina casuale. La seconda prospettiva è il risultato di una votazione fra pagine, nella quale le pagine più importanti sono più influenti e si vota una pagina avendo un link che punta verso di essa.

Possiamo definire il vettore del PageRank  $\mathbf{x}$  come l'unico autovettore con autovalore 1, che sappiamo esistere ed essere unico per il Teorema 1.4, variante di Perron-Frobenius. Dal punto di vista del random surfer questo vettore è la distribuzione stazionaria di una catena di Markov:

$$P^T \mathbf{x} = \mathbf{x}, \quad (1.1)$$

dove la matrice  $P = \alpha D^{-1} W + (1 - \alpha) F$  è stocastica per righe,  $\alpha \in (0, 1)$ ,  $D$  e  $W$  sono definite come sopra,  $F = \mathbf{1}\mathbf{v}^T$  con  $\mathbf{v} \geq 0$ ,  $\|\mathbf{v}\|_1 = 1$  e  $\|\mathbf{x}\|_1 = 1$ . Lo scalare  $\alpha$  è detto parametro di teletrasporto o fattore di amplificazione, la matrice  $F$  è detta matrice di teletrasporto e il vettore  $\mathbf{v}$  è detto vettore di distribuzione del teletrasporto.

Il secondo metodo invece, definisce il vettore del PageRank  $\mathbf{x} := \tilde{\mathbf{x}} / \|\tilde{\mathbf{x}}\|_1$ , con  $\tilde{\mathbf{x}}$  soluzione del seguente sistema lineare:

$$(I - \alpha W^T D^{-1}) \tilde{\mathbf{x}} = (1 - \alpha) \mathbf{v}. \quad (1.2)$$

Osserviamo che le due formulazioni sono equivalenti, infatti partendo dalla prima:

$$\mathbf{x} = P^T \mathbf{x} = \alpha W^T D^{-1} \mathbf{x} + (1 - \alpha) \mathbf{v} (\mathbf{1}^T \mathbf{x}),$$

e  $\mathbf{1}^T \mathbf{x} = 1$  per definizione di  $\mathbf{x}$ . Ma allora:

$$(I - \alpha W^T D^{-1}) \mathbf{x} = (1 - \alpha) \mathbf{v}, \quad (1.3)$$

che è equivalente alla seconda formulazione.

Scegliamo infine  $\mathbf{v} = \frac{1}{n} \mathbf{1}$ , in modo che la probabilità di teletrasportarsi in ciascuna pagina sia sempre la stessa.

### 1.3 PseudoPageRank

Notiamo che se non eliminiamo i dangling nodes all'inizio, il problema che otteniamo è il seguente:

$$(I - \alpha A^T D_A^+) \mathbf{x} = \frac{(1 - \alpha)}{n} \mathbf{1}, \quad (1.4)$$

dove  $(D_A)_{ii} = (A\mathbf{1})_i$ , il simbolo “+” indica la pseudoinversa di  $D_A$  e la matrice  $A^T D_A^+$  è substocastica per colonne, cioè la somma degli elementi di ciascuna colonna è sempre  $\leq 1$ . Il problema ottenuto è detto PseudoPageRank. Il seguente teorema collega strettamente PageRank e PseudoPageRank [3].

**Teorema 1.7.** *Il vettore del PageRank  $\mathbf{x}$  si può ottenere normalizzando il vettore  $\mathbf{x}$  dello PseudoPageRank:*

$$\mathbf{x} = \frac{\mathbf{x}}{\|\mathbf{x}\|_1}. \quad (1.5)$$

*Dimostrazione.* Per semplicità di notazione poniamo  $R = I - \alpha A^T D_A^+$ , così che il vettore  $\mathbf{x}$  sia soluzione di  $R\mathbf{x} = (1 - \alpha)\mathbf{v}$  per la (1.4).

È necessario fare un'osservazione su  $D^{-1}W$  prima di procedere: poiché  $W = A + \chi \mathbf{1}^T$  per definizione,  $D^{-1}W = D_A^+ A + \frac{\chi}{n} \mathbf{1}$ . In questo modo abbiamo normalizzato sia a destra che a sinistra le colonne rispetto alle rispettive 1-norme.

Per questa osservazione e per la (1.1) il vettore del PageRank  $\mathbf{x}$  risolve:

$$(I - \alpha A^T D_A^+ - \alpha \mathbf{v} \chi^T) \mathbf{x} = (1 - \alpha) \mathbf{v},$$

ovvero:

$$(R - \alpha \mathbf{v} \chi^T) \mathbf{x} = (1 - \alpha) \mathbf{v}.$$

Se  $R$  è invertibile possiamo usare il teorema di Sherman-Morrison per risolvere il sistema lineare.  $R = I - \alpha A^T D_A^+$  è invertibile, infatti  $A^T D_A^+$  è una matrice stocastica e quindi ha tutti autovalori  $\leq 1$ . Poiché  $\alpha \in (0, 1)$  e si sottrae  $\alpha A^T D_A^+$  dalla matrice identità, gli autovalori di  $R$  sono tutti diversi da 0. Allora:

$$(R - \alpha \mathbf{v} \boldsymbol{\chi}^T)^{-1} = R^{-1} + \frac{R^{-1} \mathbf{v} \boldsymbol{\chi}^T R^{-1}}{1/\alpha + \boldsymbol{\chi}^T R^{-1} \mathbf{v}}.$$

e quindi

$$\mathbf{x} = \mathbf{x} + \frac{\frac{\alpha}{1-\alpha} \boldsymbol{\chi}^T \mathbf{x}}{1/\alpha + \boldsymbol{\chi}^T \mathbf{x}} = \gamma \mathbf{x},$$

con  $\gamma$  uno scalare. Poiché  $\|\mathbf{x}\|_1 = 1$  si ha che  $\|\gamma \mathbf{x}\|_1 = \gamma \|\mathbf{x}\|_1 = 1$ , cioè:

$$\mathbf{x} = \frac{\mathbf{x}}{\|\mathbf{x}\|_1}.$$

□

In conclusione, abbiamo introdotto alcune nozioni di base e spiegato che cos'è e come impostare il problema del PageRank standard in modo ben posto. Vediamo nel prossimo capitolo la variante di questo algoritmo che ci interessa.

## Capitolo 2

# Non backtracking PageRank

Nel tempo l'algoritmo del PageRank si è diffuso a molti altri ambiti oltre che alla classificazione delle pagine web; di conseguenza ne sono state studiate delle varianti che si comportassero meglio con il nuovo problema da risolvere.

In questo capitolo studieremo una variante del PageRank chiamata non backtracking PageRank [4]: se consideriamo un cammino su un grafo, in generale, dal nodo in cui siamo si può seguire qualunque arco disponibile. Nella versione senza backtracking togliamo la possibilità di riseguire subito l'arco che abbiamo usato per arrivare nel nodo dove ci troviamo.

L'applicazione di tale variante che analizzeremo in questo lavoro è relativa ai modelli di trasporto: quando un veicolo che si muove in una certa area arriva ad un incrocio, è ragionevole supporre che non tornerà subito indietro dalla strada da cui è arrivato.

### 2.1 Edge PageRank

In questa sezione effettuiamo il primo passo per la costruzione del non backtracking PageRank: studiamo come passare dallo spazio dei nodi al suo duale. Definiamo il line graph, cioè il duale di un grafo:

**Definizione 2.1** (Line graph). Il line graph  $L(G)$  di un grafo  $G$  è un altro grafo, i cui nodi sono gli archi di  $G$  e nel quale due nodi sono adiacenti se e solo se il punto di arrivo di un arco è il punto di partenza dell'altro arco in  $G$ .

La matrice delle adiacenze  $\mathcal{W}$  associata a  $L(G)$  è definita così:

$$\mathcal{W}_{i \rightarrow j, k \rightarrow l} = \delta_{jk},$$

dove  $\delta_{jk}$  è la delta di Kronecker,  $i \rightarrow j$  indica l'arco che va da  $i$  a  $j$  in  $G$  e quindi il nodo associato in  $L(G)$ .

Osserviamo che  $\mathcal{W} \in \mathbb{R}^{m \times m}$ . Inoltre si può verificare che  $\mathcal{W} = RL^T$ .

Costruiamo una matrice diagonale  $\mathcal{D}$  con  $(\mathcal{D})_{ii} = (\mathcal{W}\mathbf{1})_i$ , la cui diagonale è composta dagli out-degrees di ciascun nodo di  $L(G)$ . Osserviamo che l'out-degree di ciascun nodo  $i \rightarrow j$  di  $L(G)$  coincide con il numero di archi che lasciano il nodo  $j$  di  $G$ ; poiché avevamo eliminato tutti i dangling nodes in partenza, abbiamo che  $\mathcal{W}\mathbf{1} > 0$  e quindi  $\mathcal{D}$  è invertibile.

Il vettore dell'Edge PageRank è definito in modo analogo a quello del PageRank, ovvero come la soluzione di:

$$\mathcal{P}^T \hat{\mathbf{x}} = \hat{\mathbf{x}}, \quad (2.1)$$

dove  $\mathcal{P} = \alpha \mathcal{D}^{-1} \mathcal{W} + (1 - \alpha) \mathcal{F}$ , con  $\alpha$ ,  $\mathcal{D}$  e  $\mathcal{W}$  come definite prima,  $\mathcal{F} = \frac{1}{N} \mathbf{1} \boldsymbol{\pi}^T$  nuova matrice di teletrasporto, dove  $N$  è il numero di nodi con in-degree positivo e  $\boldsymbol{\pi} = R(R^T R)^+ \mathbf{1}$  è il vettore che associa a ciascun arco l'inverso dell'in-degree del suo nodo di arrivo. La pseudoinversa di  $R^T R$  è necessaria, infatti non è detto che  $R^T R$  sia invertibile.

Poiché  $\boldsymbol{\pi} > 0$  e  $\|\boldsymbol{\pi}\|_1 = N$ , si ha che  $\mathcal{F}\mathbf{1} = \mathbf{1}$ , inoltre anche  $\mathcal{D}^{-1} \mathcal{W}\mathbf{1} = \mathbf{1}$  (la moltiplicazione da sinistra per  $\mathcal{D}^{-1}$  normalizza rispetto alla 1-norma le righe di  $\mathcal{W}$ ). Ma allora  $\mathcal{P}$  è stocastica per righe e  $\hat{\mathbf{x}} \geq 0$  con  $\|\hat{\mathbf{x}}\|_1 = 1$  soddisfa  $\mathcal{P}^T \hat{\mathbf{x}} = \hat{\mathbf{x}}$ . Dalla (2.1), usando il fatto che  $\|\hat{\mathbf{x}}\|_1 = 1$  e che  $\|\boldsymbol{\pi}\|_1 = N$  si ha che:

$$(I - \alpha \mathcal{W}^T \mathcal{D}^{-1}) \hat{\mathbf{x}} = (1 - \alpha) \frac{\boldsymbol{\pi}}{\|\boldsymbol{\pi}\|_1}. \quad (2.2)$$

Ci rimane da provare che questa nuova formulazione è equivalente al PageRank, cioè che proiettando indietro  $\hat{\mathbf{x}}$  nello spazio dei nodi tramite  $R^T$  troviamo il vettore  $\mathbf{x}$  del PageRank, ovvero dimostriamo che:

$$\mathbf{z} := R^T \hat{\mathbf{x}} = \mathbf{x} \quad (2.3)$$

Dimostriamolo nel caso particolare in cui non ci sono source nodes. Per farlo enunciamo i seguenti risultati:

**Lemma 2.2.** *Con la notazione precedente,  $\mathcal{D}^{-1} R \mathcal{D} = R$ .*

*Dimostrazione.* Poiché  $\mathcal{D}$  è invertibile, è equivalente provare che  $R \mathcal{D} = \mathcal{D} R$ . Scrivendo esplicitamente chi è ogni termine si ha che:

$$(R \mathcal{D})_{ej} = \begin{cases} d_j & \text{se } e \text{ è entrante nel nodo } j, \\ 0 & \text{altrimenti.} \end{cases},$$

e per come è fatta la diagonale di  $\mathcal{D}$ :

$$(\mathcal{D} R)_{ej} = \begin{cases} d_j & \text{se } e \text{ è entrante nel nodo } j, \\ 0 & \text{altrimenti.} \end{cases}.$$

□

Usiamo il risultato appena dimostrato per provare un secondo lemma:

**Lemma 2.3.** *Con la notazione precedente,  $L^T \mathcal{D}^{-1} R = W D^{-1}$ .*

*Dimostrazione.* Partendo da  $\mathcal{D}^{-1} R D = R$ , moltiplichiamo a sinistra per  $L^T$ :

$$L^T \mathcal{D}^{-1} R D = L^T R$$

e ricordando che  $W = L^T R$ , si ha che  $L^T \mathcal{D}^{-1} R D = W$ . Infine, poiché  $D$  è invertibile, si ha la tesi moltiplicando per  $D^{-1}$  da destra:

$$L^T \mathcal{D}^{-1} R = W D^{-1}.$$

□

**Teorema 2.4.** *Se un grafo non contiene alcun source node, allora  $\forall \alpha \in (0, 1)$  si ha che:*

$$(I - \alpha W^T D^{-1})^{-1} = R^T (I - \alpha W^T \mathcal{D}^{-1})^{-1} R (R^T R)^{-1} \quad (2.4)$$

*Dimostrazione.* Si ha che:

$$\begin{aligned} (I - \alpha W^T \mathcal{D}^{-1})^{-1} &= \sum_{k=0}^{\infty} \alpha^k (W^T \mathcal{D}^{-1})^k = \\ &= I + \alpha L R^T \mathcal{D}^{-1} + \alpha^2 L (R^T \mathcal{D}^{-1} L) R^T \mathcal{D}^{-1} + \alpha^3 L (R^T \mathcal{D}^{-1} L)^2 R^T \mathcal{D}^{-1} + \dots = \\ &= I + \alpha L R^T \mathcal{D}^{-1} + \alpha^2 L (D^{-1} W^T) R^T \mathcal{D}^{-1} + \alpha^3 L (D^{-1} W^T)^2 R^T \mathcal{D}^{-1} + \dots = \\ &= I + \alpha L \left[ I + \alpha D^{-1} W^T + \alpha^2 (D^{-1} W^T)^2 + \dots \right] R^T \mathcal{D}^{-1}, \end{aligned}$$

dove la prima uguaglianza vale perché  $\rho(W^T \mathcal{D}^{-1}) = 1 > \alpha$ ; nella seconda uguaglianza usiamo  $W^T = L R^T$ ; la terza uguaglianza segue dal Lemma 2; la quarta uguaglianza si ottiene raccogliendo  $\alpha L$  a sinistra e  $R^T \mathcal{D}^{-1}$  a destra.

Infine:

$$\begin{aligned} R^T (I - \alpha W^T \mathcal{D}^{-1})^{-1} R (R^T R)^{-1} &= \\ &= I + \alpha R^T L \left[ I + \alpha D^{-1} W^T + \alpha^2 (D^{-1} W^T)^2 + \dots \right] R^T \mathcal{D}^{-1} R (R^T R)^{-1} = \\ &= I + \left[ \alpha W^T D^{-1} + \alpha^2 (W^T D^{-1})^2 + \dots \right] R^T R (R^T R)^{-1} = \sum_{k=0}^{\infty} \alpha^k (W^T D^{-1})^k \end{aligned}$$

□

Dal teorema appena provato segue subito tale corollario:

**Corollario 2.5.** *Se un grafo non contiene source nodes il vettore dell'Edge PageRank si può proiettare tramite  $R^T$  al vettore del PageRank originale:  $R^T \hat{\mathbf{x}} = \mathbf{x}$ .*

*Dimostrazione.* Usando il Teorema 2.4 e la definizione di  $\boldsymbol{\pi} = R(R^T R)^{-1} \mathbf{1}$  abbiamo:

$$\begin{aligned} \mathbf{x} &= \frac{(1-\alpha)}{n} (I - \alpha W^T D^{-1})^{-1} \mathbf{1} = \\ &= \frac{(1-\alpha)}{\|\boldsymbol{\pi}\|_1} R^T (I - \alpha \mathcal{W}^T \mathcal{D}^{-1})^{-1} \boldsymbol{\pi} = R^T \hat{\mathbf{x}}. \end{aligned}$$

Si noti che utilizziamo l'ipotesi dell'assenza di source nodes per porre  $N = n$  e quindi  $\|\boldsymbol{\pi}\|_1 = n$ .  $\square$

Passiamo ora al caso in cui sono presenti  $s = n - N \geq 1$  source nodes. In questo caso dall'osservazione sulla correzione dei dangling nodes segue che non ci sono dangling nodes (se ci fossero stati, la loro correzione avrebbe indotto anche la correzione dei source nodes). La matrice delle adiacenze è di questa forma:

$$W = A = \begin{bmatrix} 0_{s,s} & S \\ 0_{N,s} & \widetilde{W} \end{bmatrix},$$

dove  $S \in \mathbb{R}^{s \times N}$  descrive gli archi che vanno da un source node a un non source node e  $\widetilde{W}$  è la matrice delle adiacenze del grafo escludendo i source nodes. Inoltre si ha che:

$$D = \begin{bmatrix} D_S & \\ & \widetilde{D} \end{bmatrix}$$

con  $(D_S)_{ii} = (S\mathbf{1})_i$  e  $(\widetilde{D})_{ii} = (\widetilde{W}\mathbf{1})_i$ . Osserviamo inoltre che  $R^T R$  non è invertibile. Con un semplice calcolo otteniamo che:

$$(I - \alpha W^T D^{-1})^{-1} = \begin{bmatrix} I_s & 0_{s,N} \\ Y & (I - \alpha \widetilde{W}^T \widetilde{D}^{-1})^{-1} \end{bmatrix},$$

con  $Y = \alpha (I - \alpha \widetilde{W}^T \widetilde{D}^{-1})^{-1} S^T D_S^{-1}$ .

Dunque, il nostro obiettivo è ancora mostrare che  $\mathbf{z} = \mathbf{x}$ . Purtroppo in presenza di dangling nodes, ciò non è vero. Infatti, invertendo l'equazione (1.3):

$$\mathbf{x} = \frac{(1-\alpha)}{n} (I - \alpha W^T D^{-1})^{-1} \mathbf{1} = \frac{(1-\alpha)}{n} \begin{bmatrix} \mathbf{1}_s \\ * \end{bmatrix},$$

mentre, invertendo l'equazione (2.2):

$$\mathbf{z} = R^T \hat{\mathbf{x}} = \frac{(1-\alpha)}{N} R^T (I - \alpha \mathcal{W}^T \mathcal{D}^{-1})^{-1} R (R^T R)^+ \mathbf{1} = \frac{(1-\alpha)}{N} \begin{bmatrix} \mathbf{0}_s \\ * \end{bmatrix}.$$

I due vettori hanno diverse almeno le prime  $s$  componenti, quindi non possono essere uguali.

Ci chiediamo quindi se ci sia un altro modo per dimostrarne l'equivalenza. Dobbiamo riformulare l'Edge PageRank in modo meno intuitivo, associando ad ogni arco il proprio nodo di partenza invece che quello di arrivo, poiché la matrice  $D = L^T L$  è sempre invertibile grazie alla correzione dei dangling nodes. Al posto di  $\boldsymbol{\pi}$  usiamo il vettore  $\boldsymbol{v} = L(L^T L)^{-1} \mathbf{1}$  che ha  $\|\boldsymbol{v}\|_1 = n$  e proiettiamo  $\hat{\boldsymbol{x}}$  nello spazio dei nodi tramite  $L^T$ . Il Teorema 2.4 si riscrive così:

**Teorema 2.6.**  $\forall \alpha \in (0, 1)$  si ha che:

$$(I - \alpha W^T D^{-1})^{-1} = L^T (I - \alpha W^T \mathcal{D}^{-1})^{-1} L D^{-1} \quad (2.5)$$

*Dimostrazione.* Ricordando che  $W^T = L R^T$ , che vale il Lemma 2.3 e che  $\rho(W^T \mathcal{D}^{-1}) = 1 > \alpha$  si procede con calcoli analoghi a quelli svolti nella dimostrazione del Teorema 2.4:

$$\begin{aligned} L^T (I - \alpha W^T \mathcal{D}^{-1})^{-1} L D^{-1} &= L^T [I + \alpha W^T \mathcal{D}^{-1} + \alpha^2 W^T \mathcal{D}^{-1} W^T \mathcal{D}^{-1} + \dots] L D^{-1} = \\ &= I + L^T [\alpha L (R^T \mathcal{D}^{-1} L) + \alpha^2 L (R^T \mathcal{D}^{-1} L)^2 + \dots] D^{-1} = \\ &= I + L^T [\alpha L D^{-1} W^T + \alpha^2 L D^{-1} W^T D^{-1} W^T + \dots] D^{-1} = \\ &= I + \alpha W^T D^{-1} + \alpha^2 W^T D^{-1} W^T D^{-1} + \dots = \\ &= (I - \alpha W^T D^{-1})^{-1}. \end{aligned}$$

□

Esiste anche un analogo del Corollario 2.5, che conclude la formulazione dell'Edge PageRank.

**Corollario 2.7.** *Il vettore dell'Edge PageRank  $\hat{\boldsymbol{x}}$  proiettato nello spazio dei nodi tramite  $L^T$  è uguale al vettore del PageRank  $\boldsymbol{x}$ :*

$$L^T \hat{\boldsymbol{x}} = \boldsymbol{x}.$$

*Dimostrazione.* Usando il teorema appena provato e ricordando come è definito  $\boldsymbol{v}$  abbiamo:

$$\begin{aligned} \boldsymbol{x} &= \frac{(1 - \alpha)}{n} (I - \alpha W^T D^{-1})^{-1} \mathbf{1} = \\ &= \frac{(1 - \alpha)}{\|\boldsymbol{v}\|_1} L^T (I - \alpha W^T \mathcal{D}^{-1})^{-1} \boldsymbol{v} = L^T \hat{\boldsymbol{x}}. \end{aligned}$$

□

## 2.2 Non backtracking PageRank

Siamo passati al duale, ma non abbiamo ancora introdotto la componente non backtracking nell'algoritmo. Per farlo, definiamo la matrice di Hashimoto  $\mathcal{B}$  come:

$$\mathcal{B}_{i \rightarrow j, k \rightarrow l} = \delta_{jk} (1 - \delta_{il}).$$

Questa matrice ha un 1 in posizione  $i \rightarrow j, k \rightarrow l$  se i due archi sono consecutivi ma non formano un cammino backtracking. Indicando con  $\circ$  il prodotto elemento per elemento si ha che:

$$\mathcal{B} = \mathcal{W} - \mathcal{W} \circ \mathcal{W}^T.$$

Anche se questa matrice è stata costruita dopo aver rimosso i dangling nodes, essa può ancora avere delle righe tutte fatte da zeri: questo accade quando un nodo ha come unico arco in uscita quello che torna nel nodo da cui siamo arrivati. Tali archi sono detti dangling edges; conducono a nodi da cui si può uscire solo con backtrack. Non dobbiamo correggere di nuovo la matrice grazie al Teorema 1.7, che ci dice che basta normalizzare la soluzione trovata per ottenere il vettore del PageRank.

Dobbiamo adesso definire la matrice di teletrasporto. Qui sorge una domanda: dovremmo consentire il backtracking durante il teletrasporto? La risposta è sì: il teletrasporto dev'essere indipendente dalla natura del problema e le scelte riguardanti l'algoritmo non dovrebbero influenzarlo. Si noti inoltre che in un grafo molto grande, la probabilità che si verifichi il backtracking durante il teletrasporto è molto bassa. Quindi:

$$\mathcal{F} = \frac{1}{n} \mathbf{1} \mathbf{v}^T,$$

con  $\mathbf{v} = LD^{-1} \mathbf{1}$  e  $\|\mathbf{v}\|_1 = n$ .

Assumiamo che non ci siano dangling edges nel grafo, altrimenti usiamo il Teorema 1.7. Definiamo la matrice  $\mathcal{P}$  per il non backtracking PageRank come:

$$\mathcal{P} = \alpha \mathcal{D}^{-1} \mathcal{B} + (1 - \alpha) \mathcal{F}, \quad (2.6)$$

con  $\alpha \in (0, 1)$ ,  $\mathcal{D}$  la matrice diagonale con il vettore  $\mathcal{B} \mathbf{1}$  sulla diagonale e  $\mathcal{B}$ ,  $\mathcal{F}$  come sopra. Poiché assumiamo che non ci siano dangling edges, si ha che il vettore  $\mathcal{B} \mathbf{1} > 0$  e  $\mathcal{P} \mathbf{1} = \mathbf{1}$ . Allora, analogamente a quanto visto per l'Edge PageRank, il vettore del PageRank nel duale è  $\hat{\mathbf{y}}$  non negativo con  $\|\hat{\mathbf{y}}\|_1 = 1$  tale che:

$$(I - \alpha \mathcal{B}^T \mathcal{D}^{-1}) \hat{\mathbf{y}} = \frac{(1 - \alpha)}{n} \mathbf{v}. \quad (2.7)$$

Il vettore  $\hat{\mathbf{y}}$  andrà infine proiettato indietro nello spazio dei nodi tramite  $L^T$ , ottenendo  $\mathbf{y} := L^T \hat{\mathbf{y}}$ . Per le ipotesi su  $\hat{\mathbf{y}}$  e poiché  $L \mathbf{1} = \mathbf{1}$  il vettore  $\mathbf{y}$  sarà non negativo e di 1-norma unitaria.

Tutto quello di cui abbiamo parlato in questa sezione si sintetizza in questa importante definizione:

**Definizione 2.8.** Sia  $\mathcal{B}$  la matrice di Hashimoto di un grafo diretto senza dangling nodes né dangling edges. Sia  $\mathcal{D}$  la matrice diagonale con il vettore  $\mathcal{B} \mathbf{1}$  sulla diagonale e sia  $L$  la matrice sorgente del grafo. Definiamo  $\forall \alpha \in (0, 1)$  il vettore del non backtracking PageRank come il vettore non negativo e di 1-norma unitaria

$$\mathbf{y} := L^T \hat{\mathbf{y}} \quad (2.8)$$

con  $\hat{\mathbf{y}} \geq 0$ ,  $\|\hat{\mathbf{y}}\|_1 = 1$  soluzione di (2.7).

*Osservazione 2.9.* Come visto nella formulazione del PageRank, il sistema lineare (2.7) che risolviamo è equivalente alla risoluzione di:

$$\mathcal{P}^T \hat{\mathbf{y}} = \hat{\mathbf{y}}. \quad (2.9)$$

### 2.3 Analogie tra i due algoritmi

Quello che ci aspettiamo, in generale, è che la classificazione ottenuta con il non backtracking PageRank sia diversa da quella ottenuta con il PageRank. Come vedremo con gli esperimenti sulle reti stradali nel Capitolo 4 è effettivamente così. Prima però vogliamo esporre un risultato che mostra l'equivalenza fra i due algoritmi sotto alcune particolari condizioni:

**Teorema 2.10.** *Sia  $W$  la matrice delle adiacenze di un grafo  $G$   $k$ -regolare non orientato, con  $k \geq 2$ ;  $\mathcal{B}$  la sua matrice di Hashimoto. Siano  $D$  e  $\mathcal{D}$  le matrici diagonali rispettivamente con i vettori  $W\mathbf{1}$  e  $\mathcal{B}\mathbf{1}$  sulla diagonale. Definiamo,  $\forall \alpha \in (0, 1)$  le matrici  $P = \alpha D^{-1}W + \frac{(1-\alpha)}{n}\mathbf{1}\mathbf{1}^T$  e  $\mathcal{P} = \alpha \mathcal{D}^{-1}\mathcal{B} + \frac{(1-\alpha)}{n}\mathbf{1}\mathbf{v}^T$ , con  $\mathbf{v} = LD^{-1}\mathbf{1}$ . Se  $P^T \mathbf{x} = \mathbf{x}$  e  $\mathcal{P}^T \hat{\mathbf{y}} = \hat{\mathbf{y}}$  con  $\|\mathbf{x}\|_1 = \|\hat{\mathbf{y}}\|_1 = 1$ , allora:*

$$L^T \hat{\mathbf{y}} = \mathbf{x}, \quad (2.10)$$

dove  $L$  è la matrice sorgente associata a  $W$ .

*Dimostrazione.* Si ricordi che un grafo non orientato può essere considerato come un grafo orientato in cui ogni arco si considera come due archi con orientazioni opposte.

Riformuliamo  $P^T \mathbf{x} = \mathbf{x}$  e  $\mathcal{P}^T \hat{\mathbf{y}} = \hat{\mathbf{y}}$  come i sistemi lineari associati:

$$(I - \alpha W^T D^{-1})\mathbf{x} = \frac{(1-\alpha)}{n}\mathbf{1}$$

e

$$(I - \alpha \mathcal{B}^T \mathcal{D}^{-1})\hat{\mathbf{y}} = \frac{(1-\alpha)}{n}\mathbf{v}.$$

Osserviamo che  $W$  è simmetrica per ipotesi poiché il grafo è non orientato. Partiamo dal primo sistema lineare: poiché il grafo è  $k$ -regolare si ha che  $W\mathbf{1} = k\mathbf{1}$  e quindi  $D^{-1} = k^{-1}I$ . Ma allora:

$$\mathbf{x} = \frac{(1-\alpha)}{n} \sum_{r=0}^{\infty} (\alpha D^{-1}W)^r \mathbf{1} = \frac{(1-\alpha)}{n} \sum_{r=0}^{\infty} \frac{\alpha^r}{k^r} k^r \mathbf{1} = n^{-1}\mathbf{1}.$$

Per il secondo sistema lineare, invece, sappiamo che  $\mathcal{B}^T \mathbf{1} = (k-1)\mathbf{1}$  per l'ipotesi di  $k$ -regolarità e sapendo che in ogni nodo viene eliminato l'arco che concede il

backtrack; quindi  $\mathcal{D}^{-1} = (k-1)^{-1}I$ . Infine, poiché  $L\mathbf{1} = \mathbf{1}$  si ha che  $\mathbf{v} = k^{-1}\mathbf{1}$ . Ma allora:

$$\hat{\mathbf{y}} = \frac{(1-\alpha)}{n} \sum_{r=0}^{\infty} (\alpha \mathcal{D}^{-1} \mathcal{B}^T)^r \mathbf{v} = \frac{(1-\alpha)}{n} \sum_{r=0}^{\infty} \frac{\alpha^r}{(k-1)^r} (k-1)^r \frac{\mathbf{1}}{k} = (kn)^{-1} \mathbf{1}.$$

Ne segue la tesi:

$$L^T \hat{\mathbf{y}} = n^{-1} \mathbf{1} = \mathbf{x}.$$

□

Dunque, sotto le ipotesi del teorema, la classificazione è la stessa. Portiamo adesso due controesempi, eliminando alcune ipotesi, che mostrano come i due algoritmi possano ottenere risultati diversi.

Nel primo esempio rimuoviamo l'ipotesi di  $k$ -regolarità e consideriamo il grafo in Figura 2.1, con due archi diretti al posto di ogni arco non orientato.

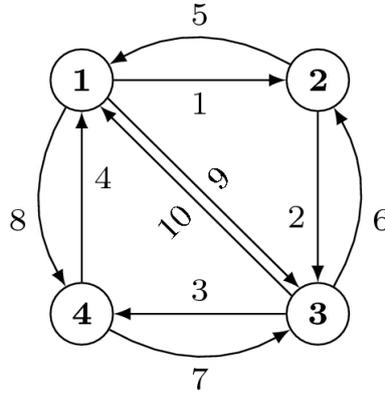


Figura 2.1: Grafo non  $k$ -regolare

Con questo ordine di nodi e archi si può scrivere la matrice sorgente  $L$ :

$$L^T = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ & 1 & 0 & 1 & 0 & 0 \\ & & 1 & 0 & 1 & 0 \\ & & & 1 & 0 & 0 \\ & & & & 1 & 0 \\ & & & & & 1 \end{bmatrix}.$$

Dalla simmetria dei nodi segue che, quando calcoliamo il vettore del PageRank  $\mathbf{x}$  le sue componenti sono tali che:  $x_1 = x_3$  e  $x_2 = x_4$ ; d'altra parte dalla simmetria degli archi, quando calcoliamo il vettore del non backtracking PageRank  $\hat{\mathbf{y}}$  le sue componenti sono tali che:  $\hat{y}_1 = \hat{y}_3 = \hat{y}_6 = \hat{y}_8$ ,  $\hat{y}_2 = \hat{y}_4 = \hat{y}_5 = \hat{y}_7$  e  $\hat{y}_9 = \hat{y}_{10}$ . Con queste relazioni è possibile risolvere i sistemi lineari dei due problemi in funzione

di  $\alpha$ , ottenendo come soluzioni:

$$x_1 = \frac{3(1+\alpha)}{4(3+2\alpha)}, \quad x_2 = \frac{3+\alpha}{4(3+2\alpha)},$$

$$\hat{y}_1 = \frac{\alpha^2 + 2\alpha + 3}{12(\alpha^2 + 2\alpha + 2)}, \quad \hat{y}_2 = \frac{\alpha^2 + 3\alpha + 2}{12(\alpha^2 + 2\alpha + 2)}, \quad \hat{y}_9 = \frac{\alpha^2 + \alpha + 1}{6(\alpha^2 + 2\alpha + 2)},$$

ma quindi le componenti del vettore  $\mathbf{y} = L^T \hat{\mathbf{y}}$  sono:

$$y_1 = y_3 = \frac{2\alpha^2 + 4\alpha + 3}{6(\alpha^2 + 2\alpha + 2)}, \quad y_2 = y_4 = \frac{\alpha^2 + 2\alpha + 3}{6(\alpha^2 + 2\alpha + 2)}.$$

Si verifica facilmente che  $x_1 = y_1$  e  $y_2 = x_2$  se e solo se  $\alpha = 0, 1$ , ma allora i due vettori  $x$  e  $y$  sono diversi  $\forall \alpha \in (0, 1)$ . Nonostante ciò, la classificazione è preservata: i nodi 1 e 3 hanno un valore più alto dei nodi 2 e 4.

Mostriamo ora un esempio nel quale la classificazione data dai due algoritmi è differente. Si consideri il grafo in figura:

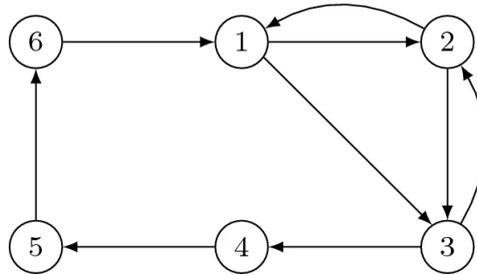


Figura 2.2: Grafo con  $\mathbf{x} \neq \mathbf{y}$

La sua matrice delle adiacenze è:

$$W = \begin{bmatrix} 0 & 1 & 1 & & & \\ 1 & 0 & 1 & & & \\ 0 & 1 & 0 & 1 & & \\ & & & 0 & 1 & \\ & & & & 0 & 1 \\ 1 & & & & & 0 \end{bmatrix}$$

Ragioniamo sul caso senza teletrasporto, ma anche quello con il teletrasporto è analogo:  $\mathbf{x}$  è tale che  $x_1 = x_2 = x_3 > x_4 = x_5 = x_6$  infatti i nodi 4, 5 e 6 possono essere visitati solo in sequenza (se si raggiunge il 4 si visitano anche gli altri 2), da cui poi si visita per forza il nodo 1: da qui si può visitare il nodo 2 o il 3, dal 2 si può visitare l'1 o il 3, dal 3 si può visitare il 2 o, passando dal 4, 5 e 6, il nodo 1. Sono quindi sensate le uguaglianze fra i 6 elementi di  $\mathbf{x}$ .

Considerando invece la variante senza backtracking si ha che:  $y_1 = y_3 > y_2 = y_4 = y_5 = y_6$ . Infatti, la differenza rispetto a prima è che se visitiamo il nodo 1 o 3

essendo passati dal nodo 2 subito prima, adesso non è possibile rivisitarlo subito. Nel caso con il teletrasporto un ragionamento analogo porta a dire che  $x_1 > x_2 = x_3 > x_4 > x_5 > x_6$  e  $y_1 > y_3 > y_2 > y_4 > y_5 > y_6$ .

In conclusione, abbiamo ottenuto il vettore del non backtracking PageRank e osservato quali possono essere le analogie e le differenze con il PageRank standard. Prima di vedere gli esperimenti, analizziamo nel prossimo capitolo come funziona GMRES, un risolutore di sistemi lineari per grandi matrici sparse che useremo sui dati negli esperimenti.

## Capitolo 3

# GMRES

In questo capitolo spiegheremo come funziona il risolutore di sistemi lineari GMRES [5]. Partiremo dai metodi di proiezione per definire l'algoritmo, faremo alcune osservazioni sul criterio d'arresto e sul restart, infine introdurremo il preconditionamento e in particolare il preconditionamento con ILU per migliorare GMRES.

### 3.1 I metodi di proiezione

Un metodo di proiezione per risolvere un sistema lineare della forma

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

è un metodo che cerca l'approssimazione  $\mathbf{x}_m$  della soluzione nel sottospazio affine  $\mathbf{x}_0 + \mathcal{K}_m$  di dimensione  $m$  imponendo la seguente condizione:

$$\mathbf{b} - A\mathbf{x}_m \perp \mathcal{L}_m,$$

dove  $\mathbf{x}_0$  è un vettore arbitrario (spesso scelto uguale a 0 per comodità),  $\mathcal{K}_m = \mathcal{K}_m(A, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \dots, A^{m-1}\mathbf{r}_0\}$  è il sottospazio di Krylov di dimensione  $m$  con  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$  e  $\mathcal{L}_m$  è un sottospazio di dimensione  $m$ .

#### 3.1.1 Il metodo di Arnoldi

Un esempio di metodo di proiezione su  $\mathcal{K}_m$  utile per arrivare a GMRES è quello di Arnoldi, creato nel 1951 per trasformare una matrice piena in forma di Hessenberg, che più tardi si scoprì avere applicazioni nell'approssimazione di autovalori di grandi matrici sparse:

**Algorithm 1:** Arnoldi

```

1: Si scelga un vettore  $\mathbf{v}_1$  di norma 1
2: for  $j = 1, 2, \dots, m$  do
3:   Si calcoli  $h_{ij} = \langle A\mathbf{v}_j, \mathbf{v}_i \rangle \forall i = 1, 2, \dots, j$ 
4:   Si calcoli  $\mathbf{w}_j = A\mathbf{v}_j - \sum_{i=1}^j h_{ij}\mathbf{v}_i$ 
5:    $h_{j+1,j} = \|\mathbf{w}_j\|_2$ 
6:   if  $h_{j+1,j} = 0$  then
7:     Stop
8:   end if
9:    $\mathbf{v}_{j+1} = \mathbf{w}_j/h_{j+1,j}$ 
10: end for

```

L'algoritmo di Arnoldi fornisce una procedura per trovare una base ortonormale di  $\mathcal{K}_m$ : ad ogni passo l'algoritmo moltiplica  $\mathbf{v}_j$  per  $A$  e ortonormalizza  $A\mathbf{v}_j$  con il metodo di Gram-Schmidt. Se  $\mathbf{w}_j$  diventa 0, l'algoritmo si ferma.

Proviamo che  $\{\mathbf{v}_1, \dots, \mathbf{v}_m\}$  è una base ortonormale di  $\mathcal{K}_m$ :

**Proposizione 3.1.** *Se l'algoritmo di Arnoldi non si è fermato nei primi  $m$  passi, allora i vettori  $\{\mathbf{v}_1, \dots, \mathbf{v}_m\}$  creati dall'algoritmo costituiscono una base ortonormale dello spazio di Krylov:*

$$\mathcal{K}_m = \text{span}\{\mathbf{v}_1, A\mathbf{v}_1, \dots, A^{m-1}\mathbf{v}_1\}.$$

*Dimostrazione.* Innanzitutto, i vettori  $\mathbf{v}_j$  sono ortonormali per costruzione, grazie alla procedura di Gram-Schmidt. Per provare che tali vettori generano l' $m$ -esimo sottospazio di Krylov si può procedere per induzione, mostrando che ogni vettore  $\mathbf{v}_j$  può essere scritto come  $q_{j-1}(A)\mathbf{v}_1$  dove  $q_{j-1}(A)$  è un polinomio di grado  $j-1$ . Il passo base con  $j=1$  è banale:  $\mathbf{v}_1 = q_0(A)\mathbf{v}_1$  con  $q_0(A) = 1$ . Assumiamo che l'ipotesi sia vera per ogni indice  $\leq j$ , proviamola per  $j+1$ . Dall'algoritmo segue che:

$$\mathbf{v}_{j+1} = \frac{\mathbf{w}_j}{h_{j+1,j}} = \frac{1}{h_{j+1,j}}(A\mathbf{v}_j - \sum_{i=1}^j h_{i,j}\mathbf{v}_i)$$

e usando l'ipotesi induttiva per  $\mathbf{v}_1, \dots, \mathbf{v}_j$  si ha:

$$\mathbf{v}_j = \frac{1}{h_{j+1,j}}(Aq_{j-1}(A)\mathbf{v}_1 - \sum_{i=1}^j h_{i,j}q_{i-1}(A)\mathbf{v}_1).$$

Abbiamo espresso  $\mathbf{v}_j$  come un polinomio di grado  $j$ , dunque la tesi è provata.  $\square$

Mostriamo un altro utile risultato per la formulazione di GMRES:

**Proposizione 3.2.** *Sia  $V_m \in \mathbb{R}^{n \times m}$  matrice che ha i vettori  $\mathbf{v}_1, \dots, \mathbf{v}_m$  sulle colonne, sia  $\tilde{H}_m \in \mathbb{R}^{(m+1) \times m}$  la matrice Hessenberg superiore con gli elementi  $h_{ij}$  non nulli trovati con l'algoritmo di Arnoldi. Allora si ha che:*

$$AV_m = V_{m+1}\tilde{H}_m. \quad (3.1)$$

*Dimostrazione.* Usando il codice dell'algoritmo si ha che:

$$\mathbf{v}_{j+1} = \frac{\mathbf{w}_j}{h_{j+1,j}} = \frac{1}{h_{j+1,j}} \left( A\mathbf{v}_j - \sum_{i=1}^j h_{i,j} \mathbf{v}_i \right) \quad \forall j = 1, \dots, m$$

e quindi isolando  $A\mathbf{v}_j$ :

$$A\mathbf{v}_j = h_{j+1,j} \mathbf{v}_{j+1} + \sum_{i=1}^j h_{i,j} \mathbf{v}_i = \sum_{i=1}^{j+1} h_{i,j} \mathbf{v}_i \quad \forall j = 1, \dots, m$$

che conclude la dimostrazione.  $\square$

## 3.2 GMRES

Anche GMRES (Generalized Minimum Residual Method) è un metodo di proiezione, nel quale si sceglie  $\mathcal{L}_m = A\mathcal{K}_m$  e  $\mathbf{v}_1 = \frac{\mathbf{r}_0}{\|\mathbf{r}_0\|_2}$ . Per trovare la soluzione di  $A\mathbf{x} = \mathbf{b}$  proviamo a risolvere il problema ai minimi quadrati

$$J(\mathbf{y}) := \|\mathbf{b} - A\mathbf{x}\|_2$$

con  $\mathbf{x} \in \mathbf{x}_0 + \mathcal{K}_m$ . Per la Proposizione 3.1,  $\mathbf{x} = \mathbf{x}_0 + V_m \mathbf{y}$ . Possiamo riscrivere:

$$\begin{aligned} \mathbf{b} - A\mathbf{x} &= \mathbf{b} - A(\mathbf{x}_0 + V_m \mathbf{y}) = \mathbf{r}_0 - AV_m \mathbf{y} = \\ &= \beta \mathbf{v}_1 - V_{m+1} \tilde{H}_m \mathbf{y} = V_{m+1} (\beta \mathbf{e}_1 - \tilde{H}_m \mathbf{y}), \end{aligned}$$

dove abbiamo usato la Proposizione 3.2 e posto  $\beta := \|\mathbf{r}_0\|_2$ . Visto che  $V_{m+1}$  ha le colonne ortonormali si ha che:

$$J(\mathbf{y}) = \|\beta \mathbf{e}_1 - \tilde{H}_m \mathbf{y}\|_2$$

Dunque, il vettore  $\mathbf{x}_m$  di GMRES si ottiene come

$$\mathbf{x}_m = \mathbf{x}_0 + V_m \mathbf{y}_m, \quad \text{dove}$$

$$\mathbf{y}_m = \underset{\mathbf{y}}{\operatorname{argmin}} \|\beta \mathbf{e}_1 - \tilde{H}_m \mathbf{y}\|_2.$$

Calcolare  $\mathbf{y}_m$  è poco costoso, infatti basta risolvere un problema ai minimi quadrati con matrice Hessenberg superiore di dimensione  $(m+1) \times m$  con  $m$  abbastanza piccolo.

Di seguito l'algoritmo di GMRES:



Definendo  $Q_m = \Omega_m \Omega_{m-1} \dots \Omega_1$  si ha:

$$\begin{aligned}\tilde{R}_m &:= Q_m \tilde{H}_m, \\ \tilde{\mathbf{g}}_m &:= Q_m(\beta \mathbf{e}_1) = \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_{m+1} \end{pmatrix}\end{aligned}$$

e poiché  $Q_m$  è una matrice unitaria:

$$\min \|\beta \mathbf{e}_1 - \tilde{H}_m \mathbf{y}\|_2 = \min \|\tilde{\mathbf{g}}_m - \tilde{R}_m \mathbf{y}\|_2.$$

**Proposizione 3.3.** *Siano  $\Omega_1, \dots, \Omega_m$  le matrici di rotazione usate per trasformare  $\tilde{H}_m$  in una matrice triangolare superiore e  $\tilde{R}_m, \tilde{\mathbf{g}}_m$  come definiti sopra. Sia  $R_m$  la matrice triangolare superiore di dimensione  $m \times m$  ottenuta eliminando l'ultima riga di  $\tilde{R}_m$  e  $\mathbf{g}_m$  il vettore ottenuto eliminando l'ultima componente di  $\tilde{\mathbf{g}}_m$ . Allora:*

1.  $\text{rg}(AV_m) = \text{rg}(R_m)$ . Inoltre se l'elemento  $r_{mm} = 0$  allora  $A$  è non invertibile;
2. Il vettore  $\mathbf{y}_m := \text{argmin}_{\mathbf{y}} \|\beta \mathbf{e}_1 - \tilde{H}_m \mathbf{y}\|_2$  è  $\mathbf{y}_m = R_m^{-1} \mathbf{g}_m$ ;
3.  $\mathbf{b} - A\mathbf{x}_m = V_{m+1} Q_m^T (\tilde{\mathbf{g}}_m - \tilde{R}_m \mathbf{y}_m)$  e quindi  $\|\mathbf{b} - A\mathbf{x}_m\|_2 = |\gamma_{m+1}|$ .

*Dimostrazione.* Vediamo la dimostrazione punto per punto:

1. Dalla definizione di  $Q_m$  e dalla Proposizione 3.2 si ha che:

$$AV_m = V_{m+1} \tilde{H}_m = V_{m+1} Q_m^T Q_m \tilde{H}_m = V_{m+1} Q_m^T \tilde{R}_m$$

La matrice  $V_{m+1} Q_m^T$  è unitaria, quindi  $\text{rg}(AV_m) = \text{rg}(\tilde{R}_m)$  e inoltre  $\text{rg}(\tilde{R}_m) = \text{rg}(R_m)$  poiché  $\tilde{R}_m$  e  $R_m$  differiscono solo di una riga, fatta da tutti 0. In particolare, se  $r_{mm} = 0$ , allora  $\text{rg}(R_m) \leq m - 1$  e quindi  $\text{rg}(AV_m) \leq m - 1$ , ma poiché  $\text{rg}(V_m)$  è massimo si ha che  $A$  non è di rango massimo e quindi è non invertibile;

2. Come accennato prima dell'enunciato:

$$\|\beta \mathbf{e}_1 - \tilde{H}_m \mathbf{y}\|_2^2 = \|\tilde{\mathbf{g}}_m - \tilde{R}_m \mathbf{y}\|_2^2 = |\gamma_{m+1}|^2 + \|\mathbf{g}_m - R_m \mathbf{y}\|_2^2$$

e il minimo è raggiunto quando il secondo addendo dell'ultimo membro è uguale a 0. Poiché  $R_m$  è invertibile, si ha che  $\mathbf{y} = \mathbf{y}_m = R_m^{-1} \mathbf{g}_m$  minimizza la norma;

3. Per  $\mathbf{x} = \mathbf{x}_0 + V_m \mathbf{y}$  e per la Proposizione 3.2, possiamo scrivere:

$$\begin{aligned}\mathbf{b} - A\mathbf{x} &= \mathbf{b} - A(\mathbf{x}_0 + V_m \mathbf{y}) = V_{m+1}(\beta \mathbf{e}_1 - \tilde{H}_m \mathbf{y}) = \\ &= V_{m+1} Q_m^T Q_m (\beta \mathbf{e}_1 - \tilde{H}_m \mathbf{y}) = V_{m+1} Q_m^T (\tilde{\mathbf{g}}_m - \tilde{R}_m \mathbf{y})\end{aligned}$$

e quindi  $\mathbf{b} - A\mathbf{x}_m = V_{m+1}Q_m^T(\tilde{\mathbf{g}}_m - \tilde{R}_m\mathbf{y}_m)$ . Dunque, dal punto 2 si ha che:

$$\mathbf{b} - A\mathbf{x}_m = V_{m+1}Q_m^T(\gamma_{m+1}\mathbf{e}_{m+1}),$$

cioè  $\|\mathbf{b} - A\mathbf{x}_m\|_2 = |\gamma_{m+1}|$ .

□

Implementiamo la ricerca di  $|\gamma_{m+1}|$  ad ogni iterazione di GMRES. Assumiamo di aver già applicato le prime  $m$  rotazioni  $\Omega_1, \dots, \Omega_m$  alla matrice Hessenberg superiore  $\tilde{H}_m$  e al vettore  $\beta\mathbf{e}_1$ . Essi diventeranno  $\tilde{R}_m$  e  $\tilde{\mathbf{g}}_m$  definiti come sopra. Aggiungiamo ora l' $(m+1)$ -esima colonna a  $\tilde{H}_m$  e l' $(m+2)$ -esima componente a  $\tilde{\mathbf{g}}_m$ . Essi diventano:

$$\tilde{H}_{m+1} = \begin{pmatrix} & h_{1,m+1} & \\ \tilde{R}_m & \vdots & \\ & h_{m+1,m+1} & \\ 0 \cdots 0 & h_{m+2,m+1} & \end{pmatrix}$$

$$\mathbf{g}_{m+1} = \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_{m+1} \\ 0 \end{pmatrix}$$

con  $\tilde{H}_{m+1} \in \mathbb{R}^{(m+2) \times (m+1)}$  e  $\mathbf{g}_{m+1} \in \mathbb{R}^{m+2}$ . Applichiamo le  $m$  rotazioni  $\Omega_1, \dots, \Omega_m$  anche sull'ultima colonna di  $\tilde{H}_{m+1}$ . Per ottenere la matrice triangolare superiore  $\tilde{R}_{m+1}$  dobbiamo fare solo un'ultima rotazione con:

$$\Omega_{m+1} = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & c_{m+1} & s_{m+1} \\ & & & -s_{m+1} & c_{m+1} \end{pmatrix}$$

Grazie a quest'ultima rotazione,  $\tilde{H}_m$  diventa  $\tilde{R}_m$  triangolare superiore,  $\mathbf{g}_{m+1}$  diventa:

$$\tilde{\mathbf{g}}_{m+1} = \begin{pmatrix} \gamma_1 \\ \vdots \\ c_{m+1}\gamma_{m+1} \\ -s_{m+1}\gamma_{m+1} \end{pmatrix}. \quad (3.2)$$

Riassumendo, abbiamo detto che GMRES va interrotto se la norma del residuo, data da  $|\gamma_{m+1}|$ , è abbastanza piccola. Dalla (3.2) si ottiene la seguente relazione iterativa per trovare il criterio di arresto:

$$\gamma_{j+1} = -s_j\gamma_j. \quad (3.3)$$

In particolare, se  $s_j = 0$  allora la norma del residuo è uguale a 0, quindi abbiamo trovato la soluzione esatta.

*Osservazione 3.4.* Osservando il codice dell'Algoritmo 2 si può notare che esso si interrompe solo quando  $h_{j+1,j} = 0$  per qualche  $j$ . Proviamo nella seguente proposizione che questo è equivalente ad aver trovato la soluzione esatta.

**Proposizione 3.5.** *Sia  $A$  una matrice invertibile. Allora GMRES si interrompe al passo  $j$  se e solo se  $\mathbf{x}_j$  è la soluzione esatta del problema.*

*Dimostrazione.* ( $\Rightarrow$ ) Poiché  $A$  è invertibile, si ha che  $r_{jj} = h_{jj} \neq 0$  per il punto 1 della Proposizione 3.3. Supponiamo  $h_{j+1,j} = 0$ , allora per la sua definizione  $s_j = 0$ . Quindi, per la (3.2),  $\mathbf{x}_j$  è la soluzione esatta.

( $\Leftarrow$ ) Se la soluzione è esatta al passo  $j$ , allora per la (3.2) si ha  $s_j = 0$  e per come è definito  $h_{j+1,j}$  anch'esso è uguale a 0.  $\square$

### 3.2.2 GMRES con restart

GMRES diventa molto costoso se  $m$  diventa troppo grande. Una soluzione a questo problema è la variante di GMRES con restart:

**Algorithm 3:** GMRES con restart

- 1: Si calcoli  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ ,  $\beta := \|\mathbf{r}_0\|_2$  e  $\mathbf{v}_1 = \mathbf{r}_0/\beta$
- 2: Si generi la base ortonormale con Arnoldi e la matrice  $\tilde{H}_m$  come prima, usando  $\mathbf{v}_1$  come primo autovettore
- 3: Si calcoli  $\mathbf{y}_m$  che minimizzi  $\|\beta\mathbf{e}_1 - \tilde{H}_m\mathbf{y}\|_2$  e  $\mathbf{x}_m = \mathbf{x}_0 + V_m\mathbf{y}_m$
- 4: Se  $\mathbf{x}_m$  non è soddisfacente si ponga  $\mathbf{x}_0 = \mathbf{x}_m$  e si torni al punto 1

Il discorso su  $|\gamma_{m+1}|$  fatto in precedenza vale ancora, quindi abbiamo lo stesso criterio d'arresto. L'unico problema di questa variante è che l'algoritmo può stagnare quando la matrice  $A$  non è definita positiva.

## 3.3 Precondizionamento di GMRES

Un problema dei risolutori di sistemi lineari iterativi come GMRES è che quando sono usati su sistemi lineari molto grandi possono convergere in maniera anche molto lenta. I preconditionatori sono stati introdotti per ridurre la difficoltà del problema e aumentare l'efficienza dei risolutori. La parte più importante del preconditionamento è la scelta stessa del preconditionatore, che dipende dal problema che stiamo affrontando. Una scelta di preconditionatore non appropriata potrebbe non migliorare affatto la risoluzione del problema.

Spieghiamo brevemente i diversi tipi di preconditionamento per GMRES, assumendo che il sistema da risolvere sia sempre  $A\mathbf{x} = \mathbf{b}$ . Esistono tre modi per

precondizionare il problema: precondizionamento da sinistra, precondizionamento da destra e precondizionamento diviso:

- Nel precondizionamento da sinistra cerchiamo di risolvere il sistema:

$$M^{-1}A\mathbf{x} = M^{-1}\mathbf{b},$$

dove  $M$  è il precondizionatore scelto opportunamente. L'algoritmo è analogo a quello di GMRES sostituendo  $M^{-1}A$  e  $M^{-1}\mathbf{b}$  rispettivamente al posto di  $A$  e  $\mathbf{b}$ . Osserviamo che la base ortonormale creata dall'algoritmo è una base di:

$$\text{span}\{\mathbf{r}_0, M^{-1}A\mathbf{r}_0, \dots, (M^{-1}A)^{m-1}\mathbf{r}_0\}.$$

Il problema più grande di questo tipo di precondizionamento è che la norma del residuo che studiamo non è più quella iniziale, il nuovo residuo è:

$$M^{-1}(\mathbf{b} - A\mathbf{x}_m),$$

dove la moltiplicazione per  $M^{-1}$  potrebbe portare a un'interpretazione errata della convergenza del metodo.

- Nel precondizionamento da destra cerchiamo di risolvere il sistema:

$$AM^{-1}\mathbf{u} = \mathbf{b},$$

dove  $M$  è il precondizionatore scelto opportunamente e  $\mathbf{u} = M\mathbf{x}$ . In questo caso, ad ogni passo  $j$  la matrice  $M^{-1}$  agisce su una combinazione lineare di vettori della base ortonormale, dando:

$$\mathbf{x}_m = \mathbf{x}_0 + M^{-1}\left(\sum_{i=1}^m \eta_i \mathbf{v}_i\right).$$

Notiamo che stavolta la base ortonormale creata è una base di:

$$\text{span}\{\mathbf{r}_0, AM^{-1}\mathbf{r}_0, \dots, (AM^{-1})^{m-1}\mathbf{r}_0\}.$$

In questo tipo di precondizionamento la norma del residuo non cambia rispetto al problema non condizionato, infatti:

$$\mathbf{r}_m = \mathbf{b} - A\mathbf{x}_m = \mathbf{b} - AM^{-1}\mathbf{u}_m.$$

- Il precondizionamento diviso considera un precondizionatore  $M = LU$  e affronta il sistema:

$$L^{-1}AU^{-1}\mathbf{u} = L^{-1}\mathbf{b},$$

dove  $\mathbf{u} = U\mathbf{x}$ . Questo tipo di precondizionamento coinvolge entrambi i precedenti, infatti ad ogni iterazione  $U^{-1}$  moltiplica i vettori della base ortonormale come nel precondizionamento da destra, mentre  $L^{-1}$  agisce sul residuo come nel precondizionamento da sinistra.

### 3.4 Fattorizzazione ILU

Consideriamo una matrice  $A$  sparsa. La sua fattorizzazione ILU (Incomplete LU) consiste nel trovare una matrice sparsa  $L$  triangolare inferiore e una matrice sparsa  $U$  triangolare superiore in modo che  $R = LU - A$  soddisfi alcune condizioni. In generale cerchiamo  $LU$  che approssimi  $A$ .

Per trovare queste due matrici ragioniamo sulla sparsità di  $A$ . Ci sono due strade: la prima è quella di mantenere le posizioni degli elementi zeri e non zeri anche nelle matrici  $L$  e  $U$ , più conveniente quando  $A$  ha una forma particolare, ad esempio tutti i non zeri stanno su alcune diagonali. La seconda è quella di far diventare zero tutti gli elementi non zeri piccoli in  $L$  e  $U$ , sotto una certa tolleranza.

Analizziamo meglio solo la variante di ILU che useremo nel quarto capitolo: eseguiamo la fattorizzazione LU con l'eliminazione di Gauss e mentre procediamo rendiamo zero tutti gli elementi il cui modulo è minore di una certa tolleranza fissata. Inoltre, per evitare di trovare pivot uguali a zero in questa procedura o un condizionatore instabile (cioè con la norma di  $M^{-1} = U^{-1}L^{-1}$  molto grande), effettuiamo ad ogni passo una permutazione delle colonne che fissa come pivot l'elemento più grande. Salvando il vettore di permutazione inversa delle colonne e applicandolo, gli elementi di  $L$  e  $U$  saranno mantenuti con il loro ordine originale. La procedura, spiegata brevemente in queste righe, si chiama ILUTP (Incomplete LU Threshold Pivoting) ed è una variante di ILU del secondo tipo.

## Capitolo 4

# Esperimenti numerici

### 4.1 Reti stradali

Passiamo agli esperimenti, eseguiti sui dati disponibili su [6]. I dati presi in considerazione descrivono le reti stradali di Anaheim (USA), Birmingham (UK), Chicago (USA) e Berlino (Germania). Ciascuna rete stradale è vista come un grafo in cui i nodi sono gli incroci e gli archi sono le strade. Le strade a senso unico sono rappresentate da un singolo arco, le strade a doppio senso sono rappresentate da due archi reciproci, con verso opposto. Nella Tabella 4.1 sono riportati per le quattro città il numero di nodi  $n$ , il numero di dangling nodes  $k$  e il numero di archi dopo aver corretto i dangling nodes  $\tilde{m}$ .

	$n$	$k$	$\tilde{m}$
Anaheim	416	0	914
Birmingham	14639	0	33937
Chicago	12982	3	77964
Berlino	12981	45	612515

Tabella 4.1: Descrizione del dataset.

Come lavoro preliminare, sono state estratte dai dati la matrice  $A$  delle adiacenze e la matrice  $W$  delle adiacenze con i dangling nodes corretti. Da esse sono state ricavate la matrice sorgente  $L$  e la matrice bersaglio  $R$ , dalle quali parte lo studio del non backtracking PageRank.

Per risolvere i sistemi lineari abbiamo utilizzato GMRES senza restart, con tolleranza  $10^{-6}$  e numero massimo di iterazioni uguale a 100. Il parametro di teletrasporto  $\alpha$  è stato inizializzato a 0.85. Gli esperimenti sono stati svolti su MATLAB Version: 9.9.0.1467703, su un Lenovo Ideapad 330 con Microsoft Windows 11, un processore Intel core i7 con 1.8 GHz e 8.00 GB di RAM.

La prima analisi ha voluto mettere a confronto PageRank (PR) e non backtracking PageRank (NBT PR), quindi abbiamo risolto il sistema lineare con le matrici  $I - \alpha W^T D^{-1}$  e  $I - \alpha \mathcal{B}^T \mathcal{D}^{-1}$ . Riportiamo in Figura 4.2 i grafici di dispersione

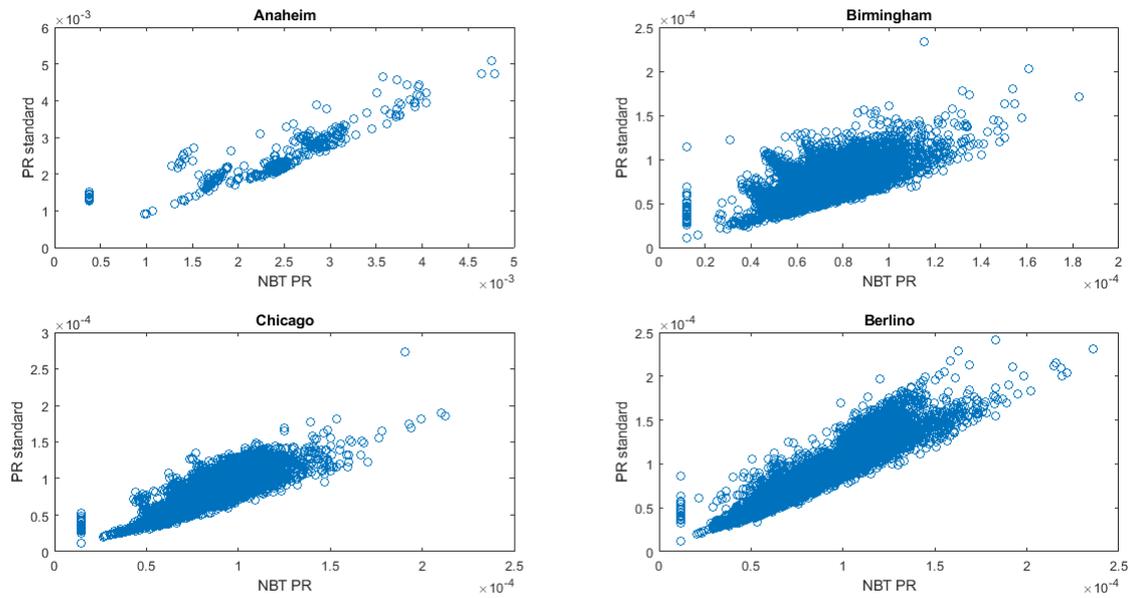


Figura 4.2: Grafici di dispersione fra non backtracking PageRank (NBT PR) proiettato indietro nello spazio dei nodi e PageRank (PR standard) messi a confronto nelle quattro reti stradali.

fra il vettore del PageRank e il vettore del non backtracking PageRank proiettato nello spazio dei nodi. Vediamo che in tutti gli esempi presi in considerazione, i grafici tendono a seguire l'andamento della diagonale, segno che i due vettori sono abbastanza correlati. Nonostante ciò, ci sono diversi nodi che si posizionano ben lontani dalla diagonale: fra le due classificazioni ci sono davvero delle differenze, come già mostrato alla fine dello scorso capitolo. Abbiamo fatto anche un altro test: ordinando i due vettori e confrontando le 10 componenti più grandi, vediamo che solo alcune rientrano fra le prime 10 con entrambe le classificazioni, più precisamente 6 per Anaheim, 5 per Birmingham, 6 per Chicago e 5 per Berlino.

I coefficienti di correlazione sono:  $\rho = 0.95$  per Berlino,  $\rho = 0.90$  per Chicago,  $\rho = 0.89$  per Anaheim e  $\rho = 0.81$  per Birmingham. Vediamo nel grafico in Figura 4.4 all'inizio della pagina seguente la correlazione fra i due vettori al variare di  $\alpha$  per il dataset di Birmingham.

Siamo poi passati allo studio dei tempi: per matrici dei dati piccole, i tempi dei due algoritmi al variare di  $\alpha$  erano sempre confrontabili, come si vede in Tabella 4.3.

$\alpha$	0.1	0.25	0.5	0.75	0.85	0.99
PR standard	0.0408	0.0172	0.0058	0.0024	0.0054	0.0528
NBT PR	0.0113	0.0126	0.0028	0.0037	0.0116	0.0257

Tabella 4.3: Tempi (s) Anaheim.

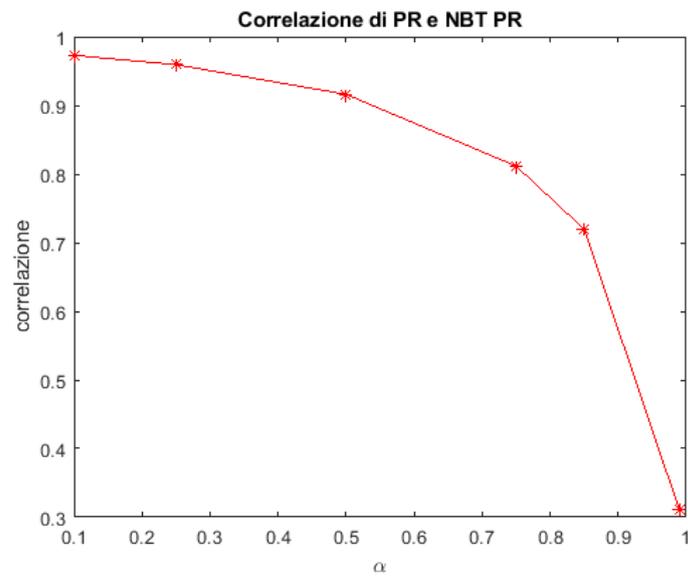


Figura 4.4: Coefficiente di correlazione fra PR e NBT PR per Birmingham al variare di  $\alpha$ .

I tempi per un network più grande, come quello di Berlino, erano invece di ordini di grandezza diversi, come si evince dal grafico in Figura 4.5 e come riportato in Tabella 4.6: nasce da qui la necessità di preconditionare il problema per farlo convergere più in fretta.

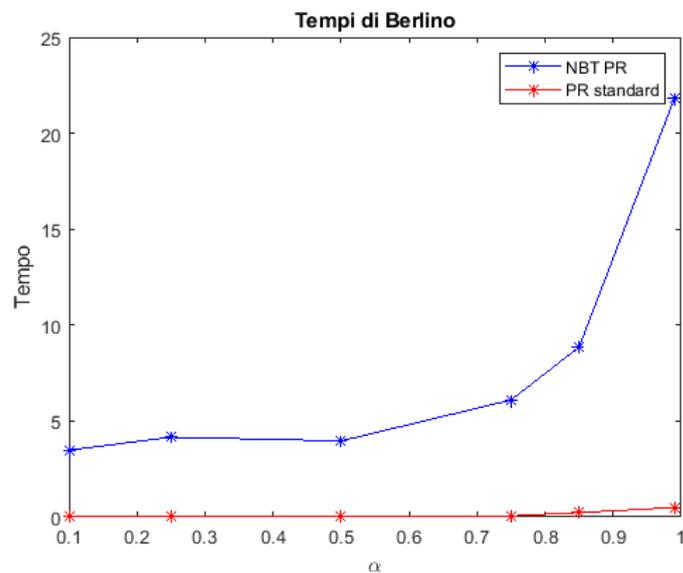


Figura 4.5: Tempo impiegato dal PageRank e dal non backtracking PageRank a confronto per  $\alpha \in [0.1, 0.25, 0.5, 0.75, 0.85, 0.99]$ , grafico riferito ai dati in Tabella 4.6.

$\alpha$	0.1	0.25	0.5	0.75	0.85	0.99
PR standard	0.0551	0.0401	0.0330	0.0575	0.2312	0.4963
NBT PR	3.4882	4.1596	3.9648	6.0992	8.8639	21.8138

Tabella 4.6: Tempi (s) Berlino.

## 4.2 Precondizionamento

Abbiamo quindi provato due tipi di precondizionamento. Il primo si basa sull'approssimazione dell'inversa della matrice  $G = I - \alpha \mathcal{B}^T \mathcal{D}^{-1}$ . Per scegliere come approssimare tale inversa, abbiamo usato il seguente teorema:

**Teorema 4.1.** *Sia  $X = \alpha \mathcal{B}^T \mathcal{D}^{-1} \in \mathbb{R}^{m \times m}$  con  $\alpha \in (0, 1)$ ,  $\mathcal{B} \in \mathbb{R}^{m \times m}$  a valori non negativi e  $\mathcal{D} \in \mathbb{R}^{m \times m}$  matrice diagonale con il vettore  $\mathcal{B}\mathbf{1}$  sulla diagonale, corretto con una costante  $k > 0$  al posto degli 0. Se  $\lambda \in \mathbb{C}$  è autovalore di  $X$ , allora  $|\lambda| < 1$ .*

*Dimostrazione.* Il vettore  $\mathcal{B}\mathbf{1}$  ha come elementi le somme per righe della matrice  $\mathcal{B}$ .  $\mathcal{D}$  è invertibile, grazie alla correzione degli elementi uguali a 0.

Facendo  $\mathcal{B}^T \mathcal{D}^{-1}$  ogni colonna viene normalizzata rispetto alla propria 1-norma o divisa per  $k$  se la sua norma era uguale a 0.

Usando ora il teorema dei cerchi di Gerschgorin per colonne possiamo stimare ogni autovalore di  $\mathcal{B}^T \mathcal{D}^{-1}$  con:

$$|\lambda_i| \leq 1, \text{ se } \mathcal{B}_{ii} = 0,$$

$$\left| \lambda_i - \frac{t}{n} \right| \leq \frac{n-t}{n}, \text{ se } \mathcal{B}_{ii} = t \text{ e } \sum_{j=1}^m \mathcal{B}_{ij} = n.$$

Dalla seconda, poiché  $|\lambda_i - \frac{t}{n}| \geq \left| |\lambda_i| - \frac{t}{n} \right|$  si ha che  $|\lambda_i| - \frac{t}{n} \leq \frac{n-t}{n}$ , cioè  $|\lambda_i| \leq 1$ . Moltiplicando infine per  $\alpha \in (0, 1)$  si ha la tesi, cioè  $\forall \lambda$  autovalore di  $X$ ,  $|\lambda| < 1$ .  $\square$

Ne segue che  $X^N \xrightarrow{N \rightarrow +\infty} 0$  e quindi possiamo approssimare l'inversa di  $G$  con una matrice  $\tilde{G}$  tale che  $G\tilde{G} = I - X^N$ , con  $N$  numero naturale sufficientemente grande. Effettivamente sul piccolo dataset di Anaheim abbiamo verificato la convergenza degli autovalori di  $G\tilde{G}$  verso 1 come si vede in Figura 4.7.

Il problema di questo tipo di precondizionamento è che salire a  $N \geq 3$  comporta effettuare una moltiplicazione per  $G$ , che è molto grande,  $N - 1$  volte, infatti:  $\tilde{G} = 2I - G$  per  $N = 2$ ,  $\tilde{G} = 3I - 3G + G^2$  per  $N = 3$  e così via. Dato che volevamo abbassare i tempi, abbiamo deciso di prendere  $N = 2$ , che come vedremo migliora il problema ma non lo risolve del tutto.

Il secondo tipo di precondizionamento è il già citato ILU: abbiamo usato la variante ILUTP e provato alcune tolleranze, decidendo infine di fissare droptol = 0.1. Abbiamo usato la funzione predefinita di MATLAB ilu con tali parametri e salvato

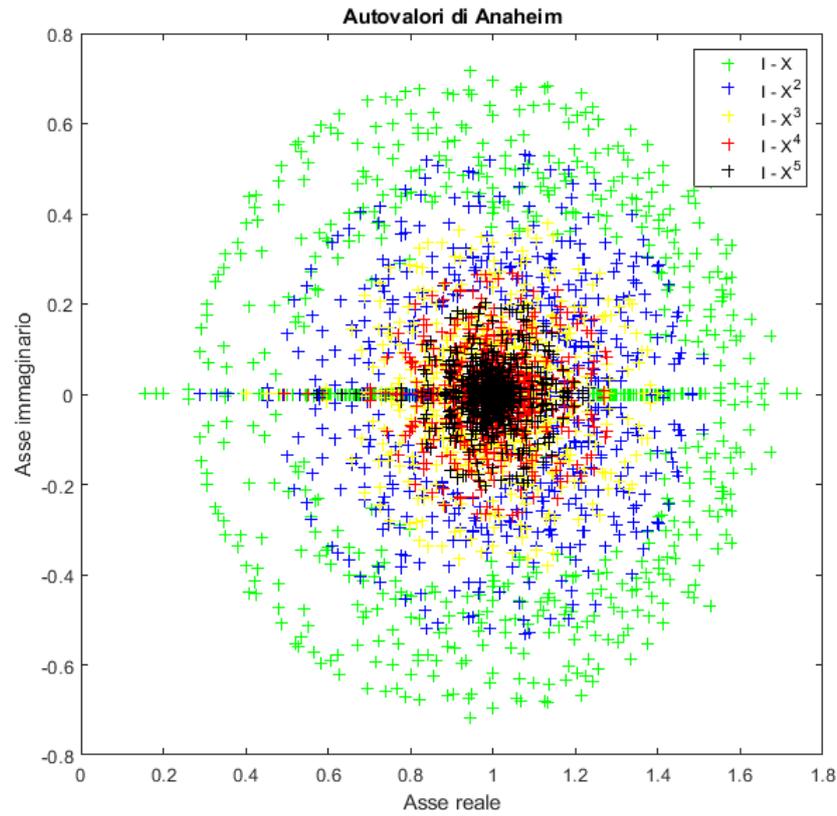


Figura 4.7: Autovalori di  $G\tilde{G} = I - X^N$  per il dataset di Anaheim.

le matrici  $L$  e  $U$  ottenute. Osserviamo subito che l'approssimazione dell'inversa non richiede memoria aggiuntiva poiché si può scrivere in funzione di  $G$ . Invece ILU crea due nuove matrici da memorizzare,  $L$  e  $U$ . In Tabella 4.8 riportiamo il numero di non zeri di queste due matrici e di  $G$ : sono dello stesso ordine di grandezza, quindi ha senso procedere con questo preconditionamento, la quantità di memoria occupata non aumenta eccessivamente.

	G	L	U
Anaheim	2.840e+03	2.297e+03	2.048e+03
Birmingham	9.783e+04	8.203e+04	8.411e+04
Chicago	4.104e+05	1.458e+05	3.085e+05
Berlino	2.885e+07	6.540e+05	3.040e+06

Tabella 4.8: Elementi non zeri di  $G$ ,  $L$  e  $U$ .

Abbiamo usato entrambi i preconditionatori mettendo in input per GMRES rispet-

tivamente la funzione  $(2I - G)\mathbf{x}$  per approssimare l'inversa e la funzione  $U^{-1}L^{-1}\mathbf{x}$  per ILU.

Abbiamo provato a scrivere anche  $G$  come funzione nella quale non venivano esplicitati i prodotti fra matrici (non ottenendo  $G$  esplicitamente quindi) per abbassare il costo delle operazioni. Nonostante ciò, abbiamo deciso di salvare esplicitamente  $G$  come matrice sparsa per due motivi: il primo è che GMRES è più lento con una funzione in input invece che con una matrice sparsa, come si vede nella Tabella 4.9.

	$G$	$G(\mathbf{x})$
Tempo (s)	0.2897	1.5653

Tabella 4.9: A confronto i tempi di GMRES per risolvere il non backtracking PageRank non preconditionato, dando come input la matrice  $G$  o la funzione  $G(\mathbf{x})$ , sul dataset di Chicago.

Il secondo motivo è che possiamo usare ILU solo se abbiamo una matrice, non possiamo farlo su una funzione.

Confrontiamo adesso le tre risoluzioni con GMRES del NBT PR, NBT PR preconditionato con  $\tilde{G}$ , NBT PR preconditionato con ILU.

	NBT PR		NBT PR $\tilde{G}$		NBT PR ILU	
	tempo	iterazioni	tempo	iterazioni	tempo	iterazioni
Anaheim	0.0279	43	0.0254	22	0.0227	7
Birmingham	0.3924	47	0.1895	24	0.0948	8
Chicago	0.6431	44	0.3107	22	0.1905	10
Berlino	7.9803	49	4.9633	25	1.6113	11

Tabella 4.10: Le tre implementazioni di non backtracking PageRank messe a confronto per le varie città, con  $\alpha = 0.85$ .

Osservando i tempi si vede che il preconditionamento velocizza GMRES e che abbassa significativamente il numero di iterazioni necessarie per arrivare alla convergenza. Inoltre il preconditionamento effettuato tramite ILU si comporta meglio di quello effettuato con l'approssimazione dell'inversa.

In conclusione, confrontiamo sul dataset di Berlino le iterazioni (Tabella 4.11) e i tempi (Figura 4.12) di non backtracking PageRank e NBT PR preconditionato con ILU al variare di  $\alpha$ .

Ricordando che  $\alpha = 0.75$  e  $\alpha = 0.85$  rappresentano le scelte più ragionevoli per algoritmi di questo tipo, prestiamo particolare attenzione alla notevole differenza di tempi e di iterazioni per tali valori.

$\alpha$	0.1	0.25	0.5	0.75	0.85	0.99
PR standard	6	9	15	31	49	100
NBT PR	6	9	15	32	49	100
NBT PR ILU	6	7	6	9	11	29

Tabella 4.11: Iterazioni impiegate dai tre algoritmi per arrivare a convergenza sul dataset di Berlino.

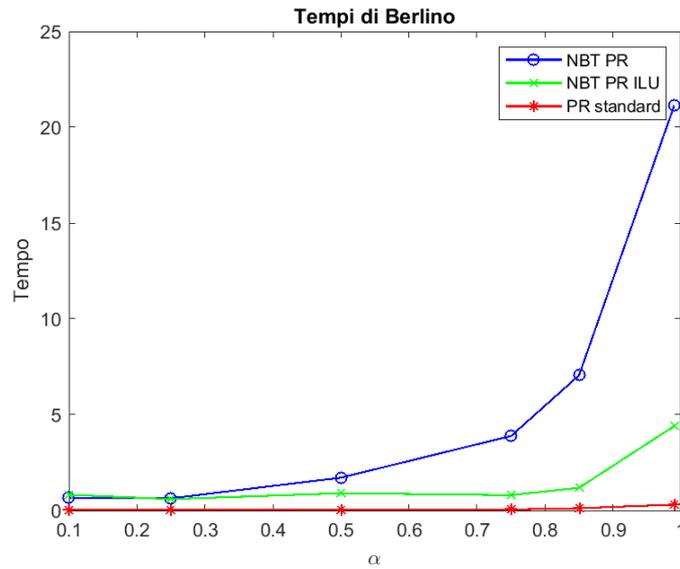


Figura 4.12: Variante del grafico in Figura 4.5 con in più i tempi del non backtracking PageRank preconditionato con ILU.

Possiamo riassumere questo capitolo osservando che il non backtracking PageRank può essere implementato in maniera efficace per dare una classificazione diversa ai dati osservati. Anche se non è il fine di questa tesi stabilire quale classificazione sia più veritiera, nell'applicazione alle reti stradali la nuova classificazione è intuitivamente ragionevole, considerando che i mezzi di trasporto raramente effettuano del backtracking. Entrambi i preconditionamenti introdotti migliorano il problema, ma a livello computazionale ILU è più performante.

# Conclusioni

In questa tesi abbiamo trattato il non backtracking PageRank, partendo con dei richiami sul PageRank e delle nozioni di base essenziali per sviluppare l'algoritmo; introducendo l'Edge PageRank, ovvero il PageRank nello spazio duale e dimostrando che proiettando nello spazio dei nodi il vettore ottenuto i due algoritmi danno la stessa classificazione. Abbiamo poi aggiunto la componente di non backtracking definendo la matrice di Hashimoto. Abbiamo infine mostrato sotto quali condizioni il PageRank e il non backtracking PageRank danno la stessa classificazione, fornendo dei controesempi che facessero vedere come invece si possono ottenere classificazioni diverse. Siamo poi passati alla parte di implementazione: per arrivare a descrivere GMRES abbiamo iniziato dai metodi di proiezioni e in particolare da quello di Arnoldi. Abbiamo riportato dei risultati teorici su GMRES e su come si possa preconditionare. Infine, nell'ultimo capitolo, abbiamo eseguito degli esperimenti sulle reti stradali di alcune città e abbiamo classificato gli incroci delle loro strade con PageRank, non backtracking PageRank standard e preconditionato tramite un'approssimazione dell'inversa e tramite ILU. Abbiamo messo a confronto i risultati ottenuti e cercato il modo più efficace di implementare il non backtracking PageRank.

# Bibliografia

- [1] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: Bringing order to the web.,” tech. rep., Stanford InfoLab, 1999.
- [2] L. Eldén, *Matrix methods in data mining and pattern recognition*. SIAM, 2007.
- [3] G. M. Del Corso, A. Gulli, and F. Romani, “Fast pagerank computation via a sparse linear system,” *Internet Mathematics*, vol. 2, no. 3, pp. 251–273, 2005.
- [4] F. Arrigo, D. J. Higham, and V. Noferini, “Non-backtracking pagerank,” *Journal of Scientific Computing*, vol. 80, no. 3, pp. 1419–1437, 2019.
- [5] Y. Saad, *Iterative methods for sparse linear systems*. SIAM, 2003.
- [6] “Transportation networks for research core team. transportation networks for research..” <https://github.com/bstabler/TransportationNetworks>. Accessed: 2021-09-01.