Laboratorio Sperimentale di Matematica Computazionale (Parte I) Lezione 5

Gianna Del Corso <gianna.delcorso@unipi.it>

15 Marzo 2017

1 PageRank

Scopo dell'esercitazione è l'implementazione dell'algoritmo PageRank implementando il metodo delle potenze od utilizzando il metodo di Arnoldi (cioè la funzione `eigs` di Matlab).

Dato un grafo diretto con n nodi espresso attraverso la sua matrice di adiacenza G, si costruisca la matrice P ottenuta dividendo gli elementi di ogni riga per l'outdegree del nodo corrispondente. Sia $\bar{P} = P + \mathbf{d}\mathbf{v}^T$ con $d_i = 1$ se il nodo i non ha nessun arco uscente e $d_i = 0$ altrimenti e con $\mathbf{v} = 1/n\,\mathbf{e}$, ed $\mathbf{e}$ vettore con tutte le componenti 1.

Sia

$$\hat{P} = (1-\alpha)\bar{P} + \alpha\,\mathbf{e}\mathbf{v}^T. \tag{1}$$

2 Dati

Per i test iniziali si può utiliazzare la matrice di adiacenza di un grafo con 10 nodi contenuta nel file `QMATRIX.mat`. Le funzioni che lavorano sulla struttura sparsa della matrice di adiacenza del grafo possono essere testati su `stanford-web.dat` che può essere letta con `load stanford-web.dat` . La lista può essere convertita in matrice con il comando `A=spconvert(stanford_web)`.

1. Si scriva una funzione

```
function Phat=build_matrix(A, alpha)
% A: matrice di adiacenza di un grafo
% alpha: parametro di "teletrasporto" valore tipico 0.15
% Phat=(1-alpha)*Pbar+\alpha evT
```

che costruisce la matrice definita stocastica $\hat{P}$ in (1).

2. Si scriva la funzione

```
function [pr]=pagerank(A, alpha)
% A: matrice di adiacenza di un grafo
% alpha: parametro di "teletrasporto" valore tipico 0.15
% pr: vettore di pageRank costruito come l'autovettore sinistro
% di Phat relativo a lambda=1
```

La funzione, presa la matrice di adiacenza di un grafo `A`, ed un valore del parametro `alpha`, deve costruire la matrice stocastica $\hat{P}$ come in (1). La funzione deve poi restituire il vettore di pagerank, cioe' l'autovettore sinistro di $\hat{P}$ corrispondente a $\lambda = 1$. Per calcolarlo si può usare il comando `eigs` con i parametri opportuni. Si testi la funzione sulla piccola matrice `QMATRIX.mat` fornita.

Si faccia un plot delle componenti vettore PageRank, cioe' dell'autovettore sinistro di $\hat{P}$, per i seguenti valori di $\alpha = 0.15, 0.1, 0.01, 10^{-8}, 10^{-12}$. Quali conclusioni si possono trarre a proposito della dipendenza di PageRank da α?

3. Si scriva la funzione

```
function [y]=P_times_x(P, x, alpha)
% P: matrice ottenuta dalla matrice di adiacenza
% scalando le righe per l'outdegree
% alpha: parametro di "teletrasporto"
% x: vettore
```

La funzione deve implementare il prodotto $y = \hat{P}\mathbf{x}$, senza esplicitamente costruire $\hat{P}$, ma lavorando sulla matrice sparsa P (si veda la formula riportata nei lucidi).

Fate qualche prova sulla piccola matrice `QMATRIX.mat` per verificare la correttezza della vostra implementazione.

3 Calcolo di Pagerank con il Metodo delle Potenze

1. Si scriva la funzione

```
 function [pr]=pr_powermethod(A, alpha)}
% A: matrice di adiacenza di un grafo
% alpha: parametro di "teletrasporto"
% pr: vettore di PageRank
```

Il vettore `pr` di PageRank è calcolato applicando il metodo delle potenze a $\hat{P}$, cioè costruendo la successione di vettori

$$\mathbf{x}^{(\mathbf{i})} = \hat{\mathbf{P}}^{\mathbf{T}}\mathbf{x}^{(\mathbf{i-1})},$$

a partire da un vettore $\mathbf{x}^{(0)}$ qualsiasi di norma-1 unitaria[1]. Si utilizzi la funzione `P_times_x` implementata al punto 2 per eseguire il prodotto matrice vettore senza costruire esplicitamente $\hat{P}$. Come criterio di arresto si usi un controllo sulla norma del residuo congiuntamente ad un controllo sul numero di iterazioni. Dopo aver testato la funzione sulla matrice piccola `QMATRIX.mat`, si testi sulla matrice di Stanford e si faccia un plot del residuo in funzione del numero di iterazione.

2. Si implementi il calcolo di PageRank utilizzando il comando `eigs`. Per non dover memorizzare la matrice piena $\hat{P}$, si chiami la funzione `eigs` con una handle alla moltiplicazione matrice-vettore[2]. Si noti che poichè in `eigs` non si può controllare la norma dei vettori nella moltiplicazione matrice-vettore in quanto `eigs` implementa un metodo di Arnoldi che non assicura che le norme-1 rimangono invariate, non si può utilizzare il la funzione `P_times_x(P, x, alpha)`, ma occorre implementarne un'altra, sempre senza memorizzare la matrice $\hat{P}$ tenendo conto che $\hat{P}$ è una modifica di rango uno di P. Si consiglia di provare le funzioni su matrici piccole per verificarne il funzionamento prima di applicare la funzione alla matrice Stanford.

3. Si testino i due algoritmi usando i valori $\alpha = 0.15$ e $\alpha = 0.01$. Si confronti la velocità di convergenza dei due algoritmi e il tempo impiegato (usano di tic e toc di Matlab). Si confrontino i vettori ottenuti di pagerank per i due valori del parametro di teletrasporto.

[1] Si noti che, per il fatto che $\hat{P}$ è stocastica, non è necessaria la normalizzazione ad ogni passo perchè partendo da un vettore di norma unitaria tutti i vettori intermedi hanno norma-1 unitaria.

[2] La sintassi è la seguente `eigs(@(x)Afun(x, P, alpha), n, 1)` dove `y=Afun(x, P, alpha)` è la funzione che implementa la moltiplicazione$\mathbf{y} = \hat{P}^T\mathbf{x}$.