Esercizio 1:Sperimentazione sulla dipendenza dal parametro γ del metodo del PageRank
Corso di LSMC, a.a. 2019-2020

Cristian Sopio
559597

28 gennaio 2022

1 Descrizione del problema

Prendiamo in considerazione il problema del calcolo del PageRank per una matrice definita da una matrice di adiacenza e da un vettore di personalizzazione come nel modello di Google destritto nelle slide del corso. Precisamente calcoliamo una approssimazione del vettore y definito da

$$y^T = y^T A$$

dove A è la matrice data da

$$A = \gamma D^{-1}(H + ue^T) + (1 - \gamma)ev^T$$

dove H è la matrice di adiacenza associata al web, e è il vettore di componenti uguali a 1, v è il vettore di personalizzazione u è il vettore di componenti 1 in corrispondenza dei nodi dangling, zero altrove, infine D è la matrice diagonale con elementi diagolnali d_i dove $d = (H + ue^T)e$. Siamo interessati a studiare quindi

1. come cambia il numero di iterazioni, fissata una taglia n della matrice di adiacenza, in funzione del parametro γ e della densità di non-zeri.

2. in che modo la variazione di γ influenzi la permutazione indotta sul vettore di PageRank risultante.

2 Descrizione della sperimentazione

Abbiamo preparato una funtion che realizza il calcolo del vettore PageRank, questa funtion è riportata nel prossimo paragrafo.
Per il primo punto abbiamo preparato lo `Script 1` in cui otteniamo una matrice

che al variare di $\gamma = [0.85, 0.9, 0.95, 0.99, 0.999]$ e delle densità prese in considerazione ($d = [10/n, 5/n, 2/n, 1/n, 0.5/n]$) restituisce il numero di iterazioni eseguite, i risultati ottenuti sono riportati in `Tabella 1`.
Per il secondo punto abbiamo invece usato la `Funtion 2` che fissati i valori di $\gamma_1 = 0.85$ e $\gamma_2 = 0.99$, restituisce la permutazione "differenza" fra gli ordinamenti del PageRank dei due valori e abbiamo riportato i grafici richiesti nell'ultimo paragrafo.

3 Script e function

Si riportano di seguito le function utilizzate nella sperimentazione.
Funtion 1

```
function [y,it] = PageRank1(H, v, gamma, itmax)
%la funzione Pagerank1 prende in entrata una matrica H di adiacenza, un
%vettore di personalizzazione v, un certo valore di gamma, e un numero
%massimo di iterazioni, applica la ricorsione dell'algoritmo del Pagerank
%fermandosi se si raggiunge un errore inferiore a 1.e-13*max(x) dove x è un
%vettore generato random oppure se si raggiunge il numero massimo di
%iterazioni.
%Si diversifica dalla funzione PageRank in quanto restituisce la coppia
%[v,it] con v il vettore di personalizzazione e it il numero di iterazioni
%svolte.
    n = size(H,1);
    usn = 1/n;
    e = ones(n,1);
    d = H*e;
    d = d';
    dang = d==0;
    dh = d + dang*n;
    x = rand(1,n);
    v = v/sum(v);
    dh = 1./dh;
        x = x/sum(x);

    for it=1:itmax

    y = x.*dh;
    y = y*H + usn*sum(dang.*x);
    y = y*gamma+(1-gamma)*v;
    err = max(abs(x-y));
    x = y;
    if err<1.e-13*max(x)
        break
    end
    end
```

```
end
```

La `Funtion 1` applica il metodo del PageRank e restituisce l'autovettore y in buona approssimazione e il numero di iterazioni che sono state necessarie.
Function 2

```
function punto2es1(n,dn)
%esegue l'algoritmo di PageRank
%su una matrice generata casualmente
%con due diversi valori di gamma
%e restituisce la permutazione 'differenza'
%fra gli ordinamenti del PageRank dei due valori
%In input:
%   n dimensione della matrice da generare
%   dn densità moltiplicata per la taglia della matrice da generare
d=dn/n;
gamma1=0.85;
gamma2=0.99;
H = sprand(n,n,d) ~= 0;
[v1,it1] = PageRank1(H, ones(1,n), gamma1, 1000);
[v2,it2] = PageRank1(H, ones(1,n), gamma2, 1000);
[z1,p1]=sort(v1,'descend');
[z2,p2]=sort(v2,'descend');
for i=1:n
    w(i)=find(p1==p2(i));
end
f1=figure(1);
hold on
plot(w(1:(n/1000)));
plot(1:(n/1000),'r');
hold off


f2=figure(2);
hold on
plot(w(1:(n/100)));
plot(1:(n/100),'r');
hold off


f3=figure(3);
hold on
plot(w(1:(n/10)));
plot(1:(n/10),'r');
hold off
```

```
f4=figure(4);
hold on
plot(w(1:n));
plot(1:n,'r');
hold off

end
```

La `Funtion 2` restituisce il vettore permutazione differenza ed esegue i plot richiesti in `Figura 1`.

Si riportano ora gli script usati nella sperimentazione.

```
%Script1 esercizio PageRank1
%Lo script esegue l'algoritmo di PageRank
%per 5 diversi valori di gamma su una matrice di adiacenze generata casualmente con 5 densit
%Restituisce poi una matrice 5x5 in cui sono riportate
%le iterazioni necessarie, in cui
%all'elemento (i,j) della matrice corrisponde
%la sperimentazione con la i-esima densità e la j-esima gamma
n=100000;
d=[10/n,5/n,2/n,1/n,0.5/n];
gamma=[0.85,0.9,0.95,0.99,0.999];
I=zeros(5,5);
for i=1:5
    for k=1:5
    H = sprand(n,n,d(i)) ~= 0;
    [v,it] = PageRank1(H, ones(1,n), gamma(k), 1000);
    I(i,k)=it;
    end
end
```

`Script 1` esegue l'algoritmo di PageRank sulla matrice di adiacenza al variare di γ e della densità di non-zeri.

4 Grafici, tabelle e commenti

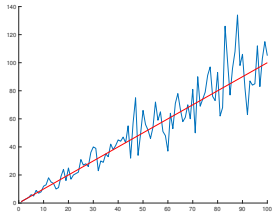
La tabella 1 riporta i valori delle iterazioni al variare di γ e della densità.

d \ γ	0.85	0.90	0.95	0.99	0.999
10/n	26	27	28	30	29
5/n	39	41	43	45	46
2/n	61	74	79	91	88
1/n	170	80	524	76	1000
0.5/n	33	36	72	39	1000

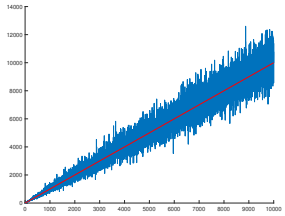
Tabella 1: Tabella che riporta ripetitivamente alla densità (righe) ed al valore di γ, (colonne) i vari numeri di iterazioni (con 1000 il valore massimo di iterazioni)

La tabella 1 riporta i valori delle iterazioni al variare di γ e della densità. Possiamo osservare che per densità come le prime tre in esame i valori rimangono presso che controllati, invece per le ultime due densità, molto basse, anche ripetendo molte volte la sperimentazione vengono risultati sempre molto diversi, quindi si vede che è legato al caso.

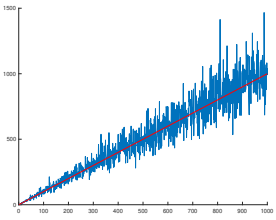
Nelle prossime figure vediamo, soprattutto nel plot completo che l'area è più ampia rispetto alla bisettrice del quadrante nella zona centrale del grafico, e tende a restringersi ai bordi del vettore. Pertanto, la variazione del valore di γ influenzerebbe maggiormente la disposizione delle pagine che, nel PageRank, occupano una posizione centrale, rivelandosi meno determinante sulle pagine classificate come decisamente importanti o, al contrario, del tutto poco importanti. Tale dato non cambia al variare delle prime tre densità delle matrici di adiacenza: per tutti e tre i valori di densità, invece per le ultime due densità, più basse si vede che la differenza è concentrata maggiormente all'inizio del vettore e quindi sulle pagine più importanti.



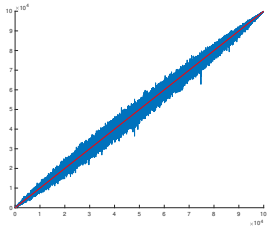
(a) *prime* 100 *componenti*



(b) *prime* 10000 *componenti*

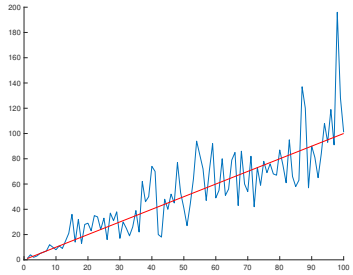


(c) *prime* 1000 *componenti*

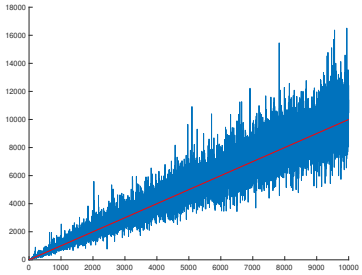


(d) *tutte le componenti*

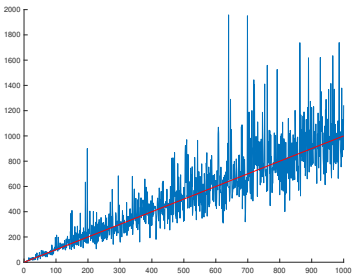
Figura 1: $n = 100000$, $d = 10/n$



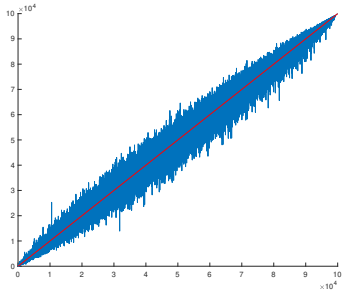
(a) *prime* 100 *componenti*



(b) *prime* 10000 *componenti*

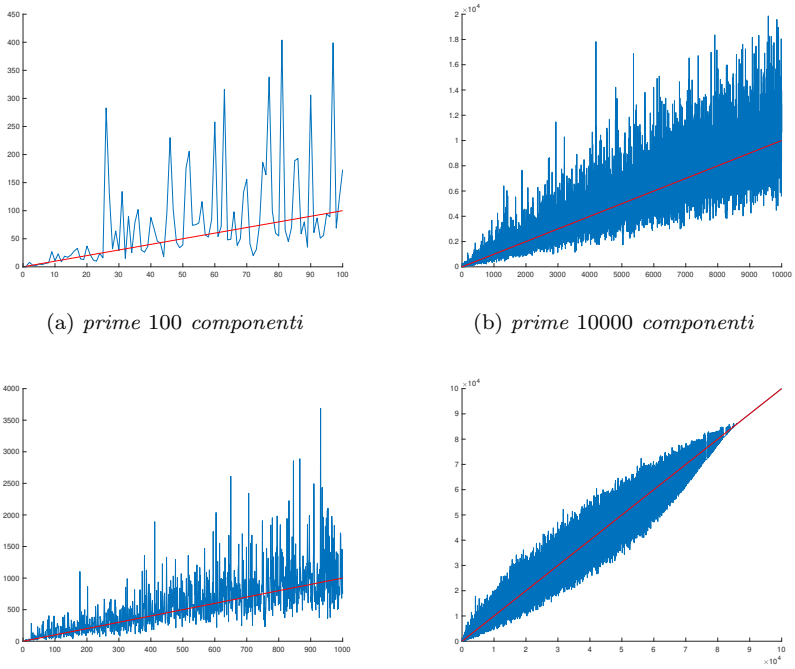


(c) *prime* 1000 *componenti*



(d) *tutte le componenti*

Figura 2: $n = 100000$, $d = 5/n$



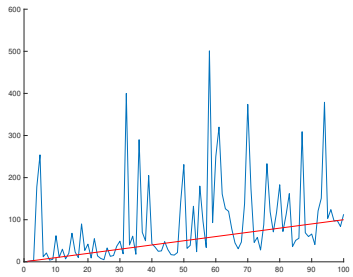
(a) *prime* 100 *componenti*

(b) *prime* 10000 *componenti*

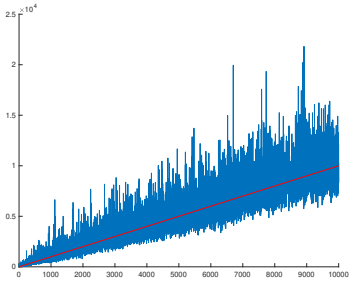
(c) *prime* 1000 *componenti*

(d) *tutte le componenti*

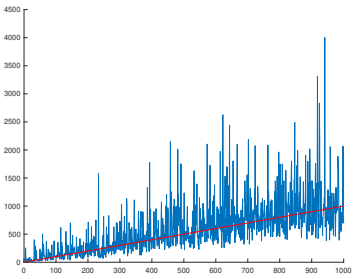
Figura 3: $n = 100000$, $d = 2/n$



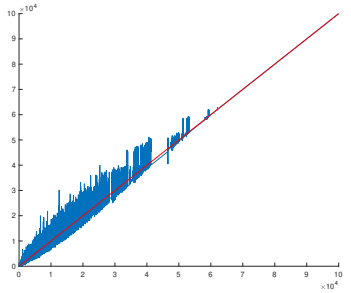
(a) *prime* 100 *componenti*



(b) *prime* 10000 *componenti*

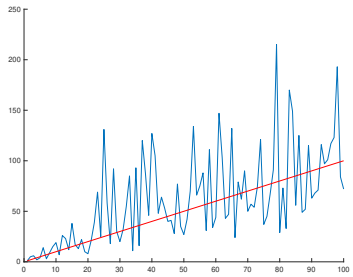


(c) *prime* 1000 *componenti*

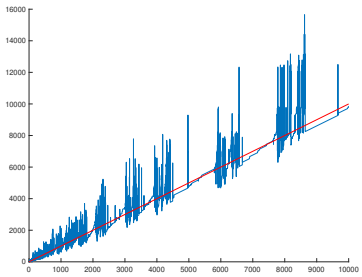


(d) *tutte le componenti*

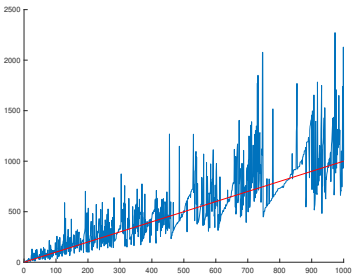
Figura 4: $n = 100000$, $d = 1/n$



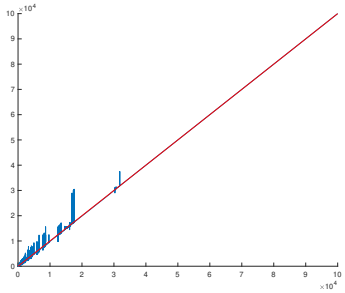
(a) *prime* 100 *componenti*



(b) *prime* 10000 *componenti*



(c) *prime* 1000 *componenti*



(d) *tutte le componenti*

Figura 5: $n = 100000$, $d = 0.5/n$