Esercizio 0: Sperimentazione sui tempi di CPU e sul numero di iterazioni 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 $d = (H + ue^T)e$. Il vettore y viene approssimato col metodo delle potenze descritto nelle note del corso.
Si intende quindi studiare

1. Studiare i tempi di esecuzioni della CPU in funzione della dimensione della matrice di adiacenza, fissato γ al variare di varie densità di non-zeri.

2. Vogliamo vedere sempre fissando un parametro di γ al variare della densità della matrice di adiacenza, ripetendo mille esperimenti quante iterazioni sono necessarie affichè il metodo iterativo degli autovalori dell'algoritmo del page rank congerga.

2 Descrizione della sperimentazione

Per il primo punto abbiamo fissato $\gamma = 0.85$ e abbiamo disegnato i grafici sovrapposti di come cambiamo i tempi di esecuzione usando varie densità. Per

il secondo punto invece abbiamo fissato $n = 10000$ e abbiamo calcolato il valor medio e la varianza, riportandoli in `tabella 1`, del numero di iterazioni al variare della densità della matrice di adiacenza.

3 Script e function

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

```
function y = PageRank(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. Restituendo in uscita una buona approssimazione
%dell'autovettore y.
    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);
    for it=1:itmax
        dh = 1./dh;
        x = x/sum(x);
    end
    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)
        return
    end
end
```

La `Funtion 1` esegue l'algoritmo di PageRank e restituisce l' autovettore y in buona approssimazione.

Funtion 2

```
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 2` analoga alla prima soltando che restituisce anche il numero di iterazioni che sono state necessarie.

Function 3

```
function A=sperimentazione(n)
%Script che data una dimensione n, esegui 'nprove' sperimentazioni con
%matrice di adiacenza generate a caso e restituisce una matrice che ha come
%prima riga le densità riportate nel vettore d, nella seconda riga le medie
%delle iterazioni svolte durante le nprove iterazioni (al variare della densità indicizzate
%le varianze durante le nprove svolte (al variare delle densità indicizzate
%su k).
d=[10/n,5/n,2/n,1/n,0.5/n];
```

```
nprove = 1000;
I=zeros(5,nprove);
A=zeros(3,5);
for k=1:5
    for j=1:nprove
        H = sprand(n,n,10/n)~=0 ;
        [v,it] = PageRank1(H, ones(1,n), d(k), 1000);
        I(k,j) = it;

    end
    A(2,k)=mean(I(k,:));
    A(3,k)= var(I(k,:));
    A(1,k)=d(k);
end
end
```

La `Funtion 3` Prende in input la taglia n della matrice e esegue la sperimentazione sul numero di iterazioni necessarie per avere la convergenza. Infine restituisce una matrice A composta, per riga, dalle denistà, dalle medie e dalle varianze del numero di iterazioni.

```
% Primo script
% script file: sperimentazione sui tempi di esecuzione al variare della
% densità scelta nel vettore d. Tale script esegue il plot di
% (dimensione,tempi di esecuzione)
N=[10000:5000:100000];
tempi = zeros(length(N),1);
d=[10/n,5/n,1/n,0.1/n];
for k=1:4
for i=1:length(N)
  n = N(i);
for j=1:20
H = sprand(n,n,d(k))  ~=0;
tic;
v = PageRank(H, ones(1,n), 0.85, 1000); t = toc;
tempi(i) = tempi(i) + t;
end
  tempi(i) = tempi(i)/20;
end
plot(N,tempi)
hold on
end
legend('10/n','5/n','1/n','0.1/n');
```

`Script 1` disegna sovrapposti i grafici al variare della densità dei tempi della CPU al variare della taglia n della matrice di adiacenza.

4 Grafici, tabelle e commenti

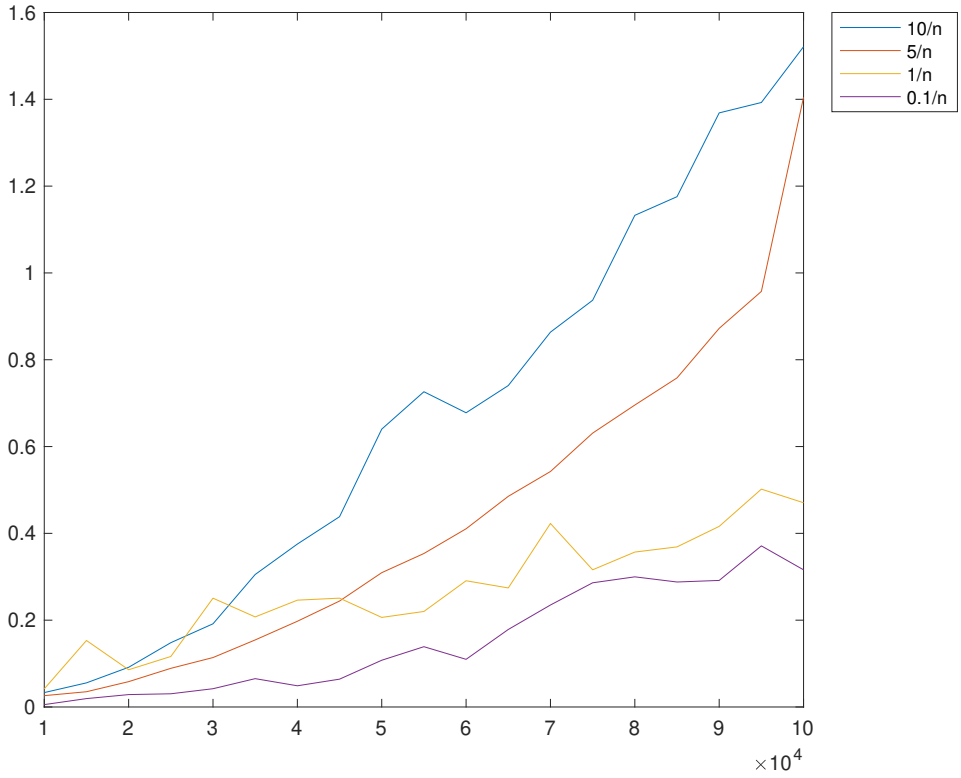

Figura 1: Figura che riporta il grafico delle sperimentazioni con densità $10/n$, $5/n$, $1/n$, $0.1/n$

Nel grafico ottenuto si può notare che l'andamento dei grafici ottenuti per vari valori del parametro è poco più che lineare e lascia intuire che i tempi di cpu crescono proporzionalmente alla dimensione della matrice, inoltre vediamo che ha un chiaro effetto sui tempi usare una desità di non-zeri molto bassa, in particolare per la densità $0.1/n$.

Densità	0.001000	0.0005	0.0002	0.0001	0.00005
Valor medio	25.987	40.812	91.763	98.982	80.367
Varianza	0.468831	357.112656	2269.508831	3011.503676	4135.868311

Tabella 1: Tabella che riporta le varie densità delle matrici di adiacenza di taglia 10000, i valori medi e le varianze del numero di iterazioni in una sperimentazione con 1000 matrici di adiacenza.

Fissata la taglia della matrice di adiacenza abbiamo che per una densità molto alta di non-zeri le iterazioni rimangono molto controllate intorno al valor medio e la varianza è pressochè nulla. Invece dimuendo la densità abbiamo varianze molto alte, dovute dal caso.