

Laboratorio Sperimentale di Matematica Computazionale (Parte I) Lezione 6

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

17 Marzo 2017

1 HITS

Scopo dell'esercitazione è studiare la relazione tra l'ordinamento dato da Degree, con quello di HITS e di Pagerank. Dapprima si implementa HITS secondo lo schema iterativo o con il calcolo dell'autovettore dominante con il comando `eigs`. Nella seconda parte si analizza un altro approccio nel quale il grado di un nodo (indegree e outdegree) viene usato per filtrare i nodi che sono più connessi e sul grafo più piccolo ottenuto mantenendo solo questi nodi si calcola Pagerank e HITS. La motivazione si basa sull'osservazione che in generale i nodi poco connessi hanno bassi punteggi di Pagerank e authority.

Esercizio 1. Si consideri un grafo diretto con n nodi e sia A la sua matrice di adiacenza. Si scriva una funzione

```
function [h, a]=Iterate_HITS(A)
% A matrice di adiacenza di un grafo
% h e a punteggi di Hub e authority
```

che partendo da due vettori $\mathbf{h}_0, \mathbf{a}_0$ restituisca due vettori \mathbf{h} e \mathbf{a} contenenti i punteggi di Hub e Authority calcolati utilizzando la formula iterativa. Alla fine di ogni iterazione si normalizzino i due punteggi ottenuti. Come criterio d'arresto si utilizzi la norma tra due iterazioni successive combinato con il numero di iterazioni effettuate.

Per un piccolo grafo (ad esempio QMATRX della volta scorsa) si confrontino i risultati ottenuti con quelli ottenuti calcolando l'autovettore relativo all'autovalore di modulo massimo di $A^T A$ e di $A A^T$ con il comando `eigs` di Matlab.

1.1 Confronto diverse strategie di ranking

Esercizio 2. Si scriva una funzione

```
function [top_idx, pr_idx, auth_idx]=rank_compare(A, n)
% A matrice di adiacenza di un grafo
% n: intero positivo
% top_idx: vettore di indici dei primi n nodi ordinati in base
% a somma di indegree e outdegree
% pr_idx: vettore di indici dei primi n nodi ordinati in base a pagerank
% auth_idx: vettore di indici dei primi n nodi ordinati in base
% al punteggio di authority
```

La funzione deve implementare il seguente algoritmo

- Calcolare l'in-degree e out-degree di ogni nodo.
- Ordinare i nodi in base alla somma di outdegree e indegree
- Prendere i primi n nodi e restituire in `top_idx` la lista ordinata.
- Creare un nuovo grafo dove solo questi nodi sono inclusi. Il nuovo grafo dovrà mantenere solo i link presenti tra questi n nodi.
- Calcolare i punteggi di PageRank (usando una delle funzioni implementate la volta scorsa) e di authority (usando `Iterate_HITS`) su questo nuovo grafo e restituire le permutazioni indotte dall'ordinamento per valori decrescenti di Pagerank (`pr_idx`) e di authority (`auth_idx`)
- Si costruisca una tabella che mostri le prime 10 componenti dei tre vettori di ranking.

Esercizio 3. Si scriva una funzione

```
function [hc, ac]=centrality(A)
% A : matrice di adiacenza
% hc: misura di centralita' come hub
% ac: misura di centralita' come authority
```

che presa la matrice A di adiacenza di un grafo di dimensione n restituisca le due misure di centralità `hc` e `ac` calcolate come segue.

- Si costruisca la matrice simmetrica

$$B = \begin{bmatrix} O & A \\ A^T & O \end{bmatrix}$$

- Si calcoli l'esponenziale di matrice di B utilizzando il comando Matlab `expm(B)`. Si ricorda che per un grafo non orientato, la diagonale della matrice $E = \exp(B) = I + B + \frac{B^2}{2} + \frac{B^3}{3!} + \dots$ può essere interpretata come una misura della centralità di un nodo, e quindi della sua importanza.
- Si calcolino i due punteggi di centralità come hub e authority rispettivamente dati da `hc(i)=E(i, i)` per $i = 1, 2, \dots, n$ e `ac(i)=E(i, i)` con $i = n + 1, \dots, 2n$.

- Sulla matrice di dimensione 1000 costruita nell'esempio precedente, si calcoli il punteggio `ac` e si confronti l'ordinamento indotto da questa misura con quelli ottenuti con le altre strategie di ranking dell'esercizio precedente.

2 Dati

Si utilizzi la matrice `stanford_web.mat` messa a disposizione la scorsa volta. Si raccomanda di provare inizialmente le funzioni implementate su grafi più piccoli.