INFORMATION RETRIEVAL
Assignment-2
DESIGN DOCUMENTATION

Semester 1 2017
13/11/17

Team:

Neetu 2015A7PS0079H

Naga Deepti Kottamasu 2015A7PS0031H

Description:

This assignment is aimed at implementing PageRank and Topic Specific PageRank. Working models for the following techniques have been built using PYTHON.

The following tasks have been completed:

- Successful implementation of the algorithm on a reasonably sized dataset.
- Handling Spider traps and Dead ends
- **Topic – specific pagerank**
- Visualization of the nodes based on pageranks

Randomized_algorithm_dataset with 742 nodes has been used as the primary dataset for testing.

Dataset has been used from :
http://www.cs.toronto.edu/~tsap/experiments/datasets/index.html

Any dateset from the given source(which follows a certain format) can be used as data for the implemented pagerank algorithm.

HTML Documentation file:

html/index.html

High Level Architecture for the functions common to all implementations
Steps-

Libraries/Packages used:

*** Pandas**
*** Pickle**
*** csv**
*** time**
*** numpy**
*** scipy**

Data collection - Reading the dataset.
The dataset consists of 3 files:

1. adj_list
2. nodes
3. inv_adj_list
4. adj_matrix

adj_matrix has been calculated from adj_list and stored seperately.

Constructing the matrix- An adjacency matrix has been created using the given dataset and is populated with the 1/0 if a node points to the other node/otherwise.

The code used to convert adj_list into adj_matrix is :
list2matrix.c

Computing the page rank- Page rank of each node is calculated.
The user is provided with two options:

1. Perform a general search
2. Perform a topic specific search

Page rank is calculated seperately for each option.

Obtaining the top 10 result- Top 10 search results either based on the topic chosen(topic-specific) or in general are displayed.

General Discussion of the algorithm used:

Pagerank Algorithm:

PageRank is a function that assigns a real number to each page in the Web (or at least to that portion of the Web that has been crawled and its links discovered).
The intent is that the higher the PageRank of a page, the more "important" it is.

In the general case, the PageRank value for any page **u** can be expressed as:

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)},$$

i.e. the PageRank value for a page **u** is dependent on the PageRank values for each page **v** contained in the set $\mathbf{B_u}$ (the set containing all pages linking to page **u**), divided by the number $L(v)$ of links from page **v**.

But , a dampening factor beta is also introduced, to address the problem of spider traps and dead ends.

Various studies have tested different damping factors, but it is generally assumed that the damping factor will be set around 0.85.

So, the equation becomes:

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

After, the algorithm is finished running, pageranks are returned in the form of an eigenvector:

$$\mathbf{R} = \begin{bmatrix} PR(p_1) \\ PR(p_2) \\ \vdots \\ PR(p_N) \end{bmatrix}$$

R is the solution of the equation:

$$\mathbf{R} = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} \ell(p_1,p_1) & \ell(p_1,p_2) & \cdots & \ell(p_1,p_N) \\ \ell(p_2,p_1) & \ddots & & \vdots \\ \vdots & & \ell(p_i,p_j) & \\ \ell(p_N,p_1) & \cdots & & \ell(p_N,p_N) \end{bmatrix} \mathbf{R}$$

such that:

$$\sum_{i=1}^{N} \ell(p_i,p_j) = 1$$

In our implementation, we have used POWER ITERATION method, to compute pagerank, which basically follows the given algorithm:

Given a web graph with n nodes, where the nodes
are pages and edges are hyperlinks
• Assign each node an initial page rank
• repeat until convergence:
calculate the page rank of each node

Topic-specific pagerank:

Suppose S is a set of integers consisting
of the row/column numbers for the pages we have identified as belonging to a certain topic (called the teleport set). Let e S be a vector that has 1 in the components in S and 0 in other components. Then the topic-sensitive Page-Rank for S is the limit of the iteration

$v' = \beta M\, v + (1-\beta) e_S / |S|$

Here, M is the transition matrix of the Web, and |S| is the size of set.