PageRank

1 General description

PageRank is an algorithm for the analysis of Internet hyperlinks, used by the Google search engine to assign a weight to each element in a set of documents interconnected by hyperlinks, with the aim of measuring relative importance within the set. Let there be a set of N resources (web pages). Each page in this set may contain links to other pages. More links will be directed to some pages than to others. In other words, a user (who surfs the Internet, accessing various pages by chance) will access some resources with a higher probability, and other resources will be accessed with a lower probability.

We can say that the pages that will be visited with a higher probability are more importantthan the others (they contain more information, are more cited scientific works than others, etc.).

A search engine will have to redirect the most important pages to the foreground (so that if a user searches for a pin (certain information, depending on the keywords typed by him) in the fan cart (the set of all web pages), then this task should not be proverbially impossible. Thus, a search engine needs a way to measure the importance of a resource within a set, and an algorithm to calculate this importance. An algorithm that calculates this factor is PageRank , and the unit of measure used is the PageRank index. We will denote PR(R) the PageRank index of the resource R.

To understand how this algorithm works, let's imagine that we want to calculate how important page A is, for example. Denote by M (A) the set of all pages from which page A can be reached by a single click. Let B M (A). Obviously, the higher the probability that a user reaches page B, the higher the probability of reaching A. Also, the higher the number of links owned by page B (we will mark this number with L(B)), the lower the probability that the next visited page will be A. If we take into account the other pages in M (A), as well as the probability that a user will continue surfing the Internet (this probability is given by a coefficient d), then the formula to calculate PR(A) (considering that PR is known (B),

$\forall B \in M(A)$) is:

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

At the web address [1] (in the Computation section) you will find the pseudocode for different algorithms to determine PR coefficients.

An important method used in the PageRank algorithm is based on fuzzy logic membership functions. Fuzzy logic ([2]) is a logic that extends the classical, Boolean logic. Thus, if in boolean logic a proposition takes values from a set whose cardinal is 2, in fuzzy logic the truth value of a proposition. These values are given by so-called member functions.

Iterative Algoritm

Let's assume the existence of a program that receives as input data a collection of N web resources and determines a graph, represented by an adjacency list. This graph will be displayed in a file, like this: on the first line the number N will be given, and on the following lines the lists of neighbors will be given: a line will start with the number of the node (or this parameter) for which the neighbors are given , will follow the number of nodes with which node i is adjacent, and the following numbers represent the nodes with which i is adjacent. On the last 2 lines, the val1 and val2 values are given (one per line; you will use them to solve the following requirements of other themes). To solve the homework requirements, you will have to build the adjacency matrix of this graph (denoted by A: A(i, j) = 0, if node i is not adjacent to node j, and 1 otherwise).

Implement the Iterative algorithm described at address [1]. Remarks:

1. Any web page analyzed contains at least one link to another web page. This means that the matrix K (from the Iterative algorithm) is invertible.

2. There are some longer pages that have links to themselves (to allow easier navigation), so not all the elements on the main diagonal of matrix A are 0. In the analysis performed, these links have no meaning, so they will not count. So $A(i, i) = 0, \forall i \in \{1, 2, \ldots, N\}$.

3. The algorithm will be implemented in the file Iterative.m; the Iterative function will receive as arguments, in this order: the name of the file from which it will read the graph, the d parameter (see its description above), the eps parameter (the error that appears in the calculation of the PR vector). It will have the PR vector as output date.

4. All input files will contain the number N on the first line, then the adjacency matrix on the next N lines.

Algebraic Algorithm

To implement the Algebraic algorithm from the address [1]. The algorithm will be implemented in the file Algebraic.m; the Algebraic function will receive as arguments, in this order: the name of the reinforcement file (which has the structure of the input file from Requirement 1), the parameter d (which has the same interpretation as in Requirement 1). It will have the PR vector as output date.

Observation. To calculate the inverse of a matrix, the Gram-Schmidt algorithm will be used: let T be an invertible matrix (with n rows and n columns) for which T −1 is required to be determined. We have: T = [t1t2 . . . tn], and T −1 = [x1x2 . . . xn] and T · T −1 = In. so,

T · [x1x2 . . . xn] = [e1e2 . . . en]

T · xi = ei (∗) where ei is the i column of the unit matrix In.

To find T −1, the system ( ) will be solved for each i separately, using the optimized Gram-Schmidt algorithm.

Degree of Membership

Let the following member function be: $$u(x) = \begin{cases} 0, & x \in [0, val_1); \\ a \cdot x + b, & x \in [val_1, val_2]; \\ 1, & x \in (val_2, 1] \end{cases}$$

where val1 and val2 are given in the input file, as described in Requirement 1; a and b are values calculated by you so that u(x) is a continuous function. This function indicates the degree of belonging of the page whose PageRank is x to the majority of important pages.

Write the file PageRank.m; the PageRank function receives as input data, in this order, a file name, the d parameter, the eps parameter. All these parameters have the above interpretation. PageRank.m will write a new file, whose name is given by the name of the file received as a parameter, to which the string .out is concatenated. It will write in the new file, on the first line, the number N (number of analyzed web pages), calculate the PR vector using the first algorithm and write it in the .out file on N lines, then calculate the PR vector using the second algorithm and -will write it in the .out file; an empty row will be left between N and the first vector, and an empty row between the two vectors. After this step, the PR vector calculated by the second algorithm will be sorted in descending order (using any sorting algorithm, this sorted vector will be denoted PR1). A free space will be left in the output file, then N lines of the form will be displayed:

i j F

where i represents indices in the vector PR1 (they will be displayed in the order: 1, 2, 3, . . . , N), j represents the node whose PageRank is PR1(i), and F = u(PR1(i)). Practically, a ranking of the most important pages will be made, a ranking in which the place obtained (i.e. number i), the number of the page that obtained this place, and the degree of belonging of this page to the majority of important pages are of interest.

2 Examples of input data and results

$d = 0.85$, $eps = 0.001$; The content of the graf1 file is:

```
7
1 3 2 3 4
2 3 3 4 5
3 5 1 2 5 6 7
4 5 1 2 3 6 7
5 3 4 6 7
6 3 1 4 5
7 4 1 2 3 5
0.001
0.95
```

The PageRank.m function will write the graf1.out

```
file:7
0.137730
0.142355
0.156163
0.176215
0.147967
0.119785
0.119785

0.137419
0.142396
0.156189
0.176532
0.147754
0.119855
0.119855

1 4  0.184965
2 3  0.163529
3 5  0.154641
4 2  0.148994
5 1  0.143750
6 7  0.125242
7 6  0.125242
```

3 Web

1. http://en.wikipedia.org/wiki/PageRank
2. http://en.wikipedia.org/wiki/Fuzzy_logic
3. http://www.cs.huji.ac.il/ csip/CSIP2007-intro.pdf