Homework assignment 3

COEN 242 instructor: Indrajit Roy

Goal: The aim of this homework is to help you understand and implement the PageRank graph algorithm in R. Submit your code as a single R file with three functions that solve each of the tasks. Upload your submission on Camino.

Due date: **May 25, 2015 (by midnight)**

Assignments have to be completed individually. Contact the grader if you have questions.

Multiple real-world graphs are present at: http://snap.stanford.edu/data/index.html. Download http://snap.stanford.edu/data/soc-Epinions1.html and use it as a test case. You will need to remove the first few comment lines in the file which describe the graph. The grader may use other directed graphs to test your program.

You may like to read about the PageRank algorithm in Chapter 5 of Mining Massive Datasets: http://infolab.stanford.edu/~ullman/mmds/ch5.pdf

Task 1: Generate adjacency matrix (4 points)

Write a R function called "getAdjMatrix(fname)" which reads the graph present in the file "fname" and returns a single R **sparse** matrix representing the graph as an adjacency matrix. This function should also print the number of vertices and edges in the graph.

Your code should use the R package "igraph" to obtain the adjacency matrix representation of the graph. You can assume that each line in the input file "fname" contains an edge, i.e., the source id followed by the destination id. This is the same format as in soc-Epinions1 file.

Task 2: Calculate transition matrix (15 points)

Write a R function called "getTranstitionMatrix(A)" which takes a sparse adjacency matrix A (NxN) as input and returns a single R list with two components "T" (NxN **sparse** matrix) and "z" (1xN dense matrix). You may need to use the "Matrix" library to perform sparse matrix operation, i.e., use library(Matrix).

1) The transition matrix "T" is calculated as follows: Each element T_{ij} (row "i" and column "j") has value 1/k if vertex "j" has "k" outgoing edges in A, and one of them is to vertex "i". Otherwise T_{ij}=0. [Hint: To obtain T, if an element in A belongs to column C_j, you need to divide it by the sum of the values in C_j].

2) "z" is a 1xN matrix. Each z_j is 0.15/N if the jth column in A has at least one non-zero entry, otherwise it is 1/N. We will use "z" in the next task. It models the case that a person at a certain vertex (or Webpage) may randomly jump to a location with some probability even if there are no direct edges to the final location.

Task 3: Implement PageRank algorithm (6 points)

Write a R function called "myPageRank(T, z, niter)" which takes the arguments "T" (NxN sparse transition matrix), "z" (1xN) matrix, and "niter" (number of iterations), and returns a single Nx1 matrix which represents the PageRank values after "niter" iterations. PageRank can be calculated using the following steps in myPageRank function:

1) Create an initial Nx1 PageRank vector called "xold". Each element of "xold" is 1/N
2) Let's call "xnew" the new PageRank vector. It can be calculated by the formula:
   xnew = (0.85*T*xold) + (z*xold)
3) Now perform step 2 for "niter" number of iterations, i.e., make "xnew" as the old PageRank vector and apply the equation provided in step 2.

Curious about correctness of your code?

R's igraph package provides a PageRank function. Use the function "page.rank.old" that implements the power method and is similar to what you have implemented. Note that there may be some differences between your results and what igraph's PageRank method returns. [There are no points for this step].

R's "igraph" documentation: http://cran.r-project.org/web/packages/igraph/igraph.pdf