Computer Assigment

In the file Stanweb.dat, you will find in compressed form the connectivity matrix for the webpages of Stanford University. Specifically in the first column are contained the nodes while in the second the node with which is connected. Using the notation of the tutorial pagerank.pdf do the following:

a) Find the vector π with

i) the Power method

ii) solving the corresponding system

as they described in paragraph 5.1 and 5.2 of the tutorial. For both methods consider as $\alpha = 0.85$ and stopping criterion $\tau = 10^{-8}$ and the vector a having 1 if it is corresponds to a node with no out links and 0 otherwise. Are the results the same for both methods? Which method seems to be faster? Use Gauss Seidel method for the iterative solution of the system.

b) Do the previous task with $\alpha = 0.99$. Your remarks on the convergence speed. Did the ranking of the first 50 nodes changed?

c) When we use the power method do all the cmponents of π converge at the same speed to their limits? If not which of the converge faster: those that correspond to important nodes or to non important ? Do you observe the same behavior when you find π through the solution of the linear system?

A typical way to raise the PageRank of a page is to use ?link farms?, i.e., a collection of ?fake? pages that point to yours in order to improve its PageRank. Our goal in this problem is to do a little analysis of the design of link farms, and how their structure affects the PageRank calculations. Consider the web graph. It contains n pages, labeled 1 through n: Of course, n is very large. As mentioned in, we use the notation $G = \alpha P + \frac{1-\alpha}{n} I$ for the transition matrix. Let π_i denote the PageRank of page i and $\pi = (\pi_1, \pi_2, \ldots \pi_n)$ denote the vector of PageRanks of all pages. Note: For a page that has k outgoing links, we put $1 = k$ for the corresponding entries of P: However, when a webpage has no outgoing links, we add a 1 as the corresponding diagonal element of P for making its row-sum one. Note that this makes G a valid transition probability matrix.

a) You now create a new web page X (thus adding a node to the web graph). X has neither in-links, nor out-links. Let $\tilde{\pi} = (\tilde{\pi}_1, \tilde{\pi}_2, \ldots \tilde{\pi}_n)$ denote the vector of new PageRanks of the n old web pages, and x denote the new PageRank of page X: In other words, $(\tilde{\pi}_1, \tilde{\pi}_2, \ldots, \tilde{\pi}_n, x)$ is the PageRank vector of the new web graph. Write $\tilde{\pi}$ and x in terms of r: Comment on how the PageRanks of

the older pages changed due to the addition of the new page (remember n is a very large number). Hint: Use the stationary equations to calculate PageRank, not the iterative approach.

b) Unsatisfied with the PageRank of your page X; you create another page Y (with no in-links) that links to X: What are the PageRanks of all the $n + 2$ pages now? Does the PageRank of X improve?

c) Still unsatisfied, you create a third page Z: How should you set up the links on your three pages so as to maximize the PageRank of X?

d) You have one last idea, you add links from your page X to older, popular pages (e.g.: you add a list of ?Useful links? on your page). Does this improve the PageRank of X? Does the answer change if you add links from Y or Z to older, popular pages?

e) Describe what steps you might take to raise the PageRank of X further. You do not need to prove anything here, just summarize your thoughts based on the previous parts. For extra credit though, you can prove what the structure for a link farm with m nodes should be to optimize the PageRank of X.