Morkunas, Edward

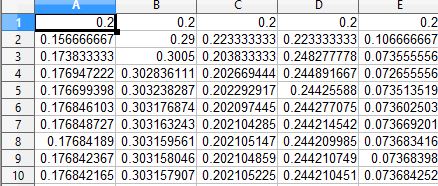
Timothy Indrieri

Brandon Artner

Page Rank

Page Rank is an algorithm designed to understand the efficiency of how a user browses through the internet. Consider your average Joe clicking on a website with an embedded link. He clicks on that embedded link and goes to another website with another link. This will continue forever so there is a term called the dampening factor which will be used to understand the way a user thinks in our case with Joe the dampening factor is one. However, our example will use a dampening factor of .15 because the user is different in the way he selects his hyperlinks. In this experiment we are also interested in how the PageRank algorithm converges because we want to know which website is on the top of the “Ranking” list. We have implemented an algorithm that uses the information available to the public with the number of websites and where those websites link to.

The algorithm was initialized by taking in a file extension .txt that contained a matrix filled with ones and zeroes. We then filled a two dimensional array to represent the network. We also created separate lists for the count of outgoing links, a list of page ranks, and a list containing tuples to indicate where each website links to. Next we used a while loop with a flag to indicate that the page ranks converged. This is calculated by a method defined at the top of the program. The method checks the floating point values with a threshold to check if they are converging after a number of steps. The threshold is .005. In the while loop we have a temp list that gets updated as we compute the ranks of each page. The incoming links of each ranks page are known through the list as stored tuples. We use the indices to access the initialized page ranks and compute the values. After the computations are made we then multiply each value in the temp list by the dampening factor, which is set to .15, and add 0.75. Then we perform a check to see if the new numbers are within the threshold of the old numbers. The final step of each loop is to update the actual page rank list so that we know we have finished a stage of the loop.



The image above shows our expected results, calculated through finding general equations for a 6x6 matrix and expanding out the steps through Excel. We notice that the numbers converge after 3 steps, however we extended the number of stages to make sure that it converges.

We know that our pagerank algorithm will terminate because we apply a dampening factor of 0.15 that should decay our ranks every iteration of the loop. As we continue to calculate each rank, it should reach that threshold and eventually terminate. Here is two example outputs of our program:

