Page Rank

STA 325: Lab 4, Fall 2017

Today's agenda: Page rank

Programming partner's: You should have a programming partner for each lab, and you should switch off who is programming, and use each other for help. We will spend about 30–50 minutes per week on lab exercises and you will be expected to bring you laptops to class to work on these exercises in class. Myself and the TA's will be in class to help you.

Lab Tasks

1. Read in the file titled pageRank.RData. Use the ls() command and ensure that the link matrix myData is present in the environment.
2. Count the number of links for each page (at each row) and place these into a diagonal matrix. Call this matrix M. Invert the matrix M and store the inverse into the variable MInv.}
3. Based on the link matrix and the matrix M compute the Broken Rank. What is the problem here? Hint: Before performing the calculation remember to transpose your link matrix since the you need the links to go from $j \to i$
4. Create a dampening parameter d and set this to 0.85. Next, count the number of webpages present in the link matrix and store this in the variable n. Next, initialize a matrix E with dimensions equal to the link matrix that contains only 1s. Now combine these elements to compute the Page Rank vector for the link matrix. How is this different from the Broken Rank? Hint: Before performing the calculation remember to transpose your link matrix since the you need the links to go from $j \to i$.
5. Consider the following test data below

```
myNewData <- matrix(c(0,1,1,0,0,0,1,0,1,1,1,1,0,1,0,0,0,0,0,0,1,0,1,0,0,1,1,0,0,0,0,0,1,0,0,0),
            nrow=6,ncol=6,byrow=TRUE)
```

Write a function myPageRank that takes as its inputs a link matrix and a dampening parameter and outputs the Page Rank vector. Now test this on the the link matrix myNewData and the dampening parameter d. Hint: Before performing the calculation remember to transpose your link matrix since the you need the links to go from $j \to i$.