HW 6

Bayard Walsh

May 2023

1

1.1

Following the definitions of $M_{i,j}$ and $A_{i,j}$ given earlier, we have the two matrices below:

$$M = \begin{matrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{matrix}$$

$$A = \begin{matrix} 0 & 0 & 1 & 0 \\ 1/2 & 0 & 0 & 0 \\ 1/2 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{matrix}$$

Note that for A every column sums exactly to 1

1.2

see code

1.3

see code

1.4

First we will find the adjacency matrix M in this case with no dampening parameter. We have this below for a matrix of size: $(N \cdot N)$:

$$M = \begin{matrix} 1 & 1 & \dots & 1 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{matrix}$$

I will justify this matrix with a few observations below

First: every page links to Facebook. Therefore every page has a link pointing to another page, so we can disregard the "dangling nodes" formula

Second: Every page links to Facebook, and Facebook is 1. Therefore every $\{1,1\} \cdots \{1,n\}$ index must equal 1

Third: No pages link to any other page. Therefore for every point where $i \neq 1$ must be 0

Now we have justified M. Next I will justify that $A = M$ (without considering α or the Google matrix). This is seen by the fact that A takes every 1 node in M, say $m_{i,j}$ and divides it by the number of pages that page j points to. As each node only points to one page, we have 1/1 for every 1, meaning that we have $A = M$ (all zeroes remain unchanged). In other words, all columns are already normalized by the problem.

Now we will find A with the damping parameter α, or translate to the Google matrix. As we know the A matrix is all composed of 1 or 0 we can address those two cases based on the equation:

For the 1 case, we have $\alpha(1) + \frac{(1-\alpha)}{n}$ or $\alpha + \frac{(1-\alpha)}{n}$
For the 0 case, we have $\alpha(0) + \frac{(1-\alpha)}{n}$, or $\frac{(1-\alpha)}{n}$

We now have the google matrix with respect to a given α

$$Google = \begin{matrix} \alpha + \frac{(1-\alpha)}{n} & \alpha + \frac{(1-\alpha)}{n} & \cdots & \alpha + \frac{(1-\alpha)}{n} \\ \frac{(1-\alpha)}{n} & \frac{(1-\alpha)}{n} & \cdots & \frac{(1-\alpha)}{n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{(1-\alpha)}{n} & \frac{(1-\alpha)}{n} & \cdots & \frac{(1-\alpha)}{n} \end{matrix}$$

Now we will solve for x, y given the following page rank vector:
$\pi = [x, y, y \cdots y]^T$, such that $\pi = Google \cdot \pi$
First, we have

$$x = x(a + \frac{1-a}{n}) + (n-1)y(a + \frac{1-a}{n})$$

by matrix multiplication. Next:

$$x = (a + \frac{1-a}{n})(x + (n-1)y)$$

By normalization property, we have $(x + (n-1)y) = 1$ Therefore we have our x which is:

$$x = a + \frac{1-a}{n}$$

Next, we have

$$y = x(\frac{1-a}{n}) + (n-1)y(\frac{1-a}{n})$$

by matrix multiplication. Next

$$y = (\frac{1-a}{n})(x + (n-1)y)$$

Again, as we know the page rank vector is normalized, so we have $(x+(n-1)y) = 1$ Therefore we have our y which is

$$y = (\frac{1-a}{n})$$

Note that based on the page rank vector composition $\pi = [x, y, y \cdots y]^T$, we can find any page rank value given we know the value of x (page rank of Facebook) and y (page rank of any other page)

2

see code for technical work/ graphs

Observations: Based on the performance of varying sigmas, $\sigma = .001$ performed the worst, whereas the other extreme side of magnitude $\sigma = 10000$ performed poorly, but not as bad as the order of magnitude would suggest (somewhat similar to $\sigma = 1.5$ performance. Sweet spot at roughly $\sigma = .075$