# Project II

**Instructions:**

- This assignment may be completed in groups of up to 3 people.
- Your write-up along with a print-out of your code should be submitted on gradescope by 11pm (Pacific Time) on Sunday 8th November. (One submission per group! When you submit in gradescope, select group submission and then select the members of your group. The other members of the group will receive email notification that a submission has been made in their name.)
- 
- Your matlab code should be fully commented so it is easy to read and understand, but your write-up should stand alone without us having to read your code.

## Introduction

Read Examples 8.1 and 8.3 in the text about Google's Page Rank Algorithm for ranking webpages according to their importance. In this project you will be asked to implement the page rank algorithm. Here are the important points.

**The Link Matrix $A$.** Suppose you have $n$ webpages that link to each other. Let $N_j$ be the number of different webpages *to which* webpage $j$ links. The link matrix is defined as

$$A_{ij} = \begin{cases} \frac{1}{N_j} & \text{if webpage } j \text{ links to webpage } i \\ 0 & \text{otherwise} \end{cases}$$

Notice, each column of $A$ sums to 1 (unless there is a webpage that links to no other pages in which case the corresponding column is all 0's) but the rows sum to more or less than 1. Matrices whose entries are all non-negative and whose columns sum to 1 are called *stochastic matrices* and arise in the study of finite state Markov processes.

**A Property of the Link Matrix.** Suppose all webpages link to at least one other webpage, so the matrix $A$ is stochastic. Notice, if $x$ is a vector whose entries sum to 1 then, because the columns of $A$ all sum to 1, the vector $Ax$ will also have the property that its entries sum to 1. We can see this as follows.

$$\sum_i (Ax)_i = \sum_i \sum_j A_{ij} x_j = \sum_j x_j \left( \sum_i A_{ij} \right) = \sum_j x_j = 1.$$

**The Importance Vector.** If we let $x = [x_1, x_2, \ldots, x_n]^T$ be a vector whose $i$'th entry is the importance we give to webpage $i$, then we want the vector $x$ to satisfy

$$x = Ax.$$

In other words, $x$ should be an eigenvector with corresponding eigenvalue 1.

**Some Questions.** Some questions arise:

- Does such a vector $x$ exist? In other words, is 1 an eigenvalue of $A$?
- Is $x$ unique up to magnitude? In other words, is the eigenspace corresponding to 1 one-dimensional?

- Is 1 the eigenvalue of $A$ of largest magnitude?
- Are the entries in $x$ all positive?

The theory of stochastic matrices tells us that, under certain conditions, the answers are all affirmative. But to apply the theory we need that a) the matrix $A$ be stochastic (so all webpages link to at least one other page) and b) the conditions are met. The following alternative way to think about the importance vector gives us some insight into appropriate ways to deal with these two issues.

**The Importance Vector as an Equilibrium Distribution.** There is another important way to think about this situation that provides a lot of intuition. Suppose, for now, that all webpages link to at least one other webpage, so the matrix $A$ is stochastic. Consider a vector $x_0$ whose entries are all non-negative and sum to 1. We can think of $x_0$ as a *distribution* of people on the webpages. For example, in a toy example where $n = 6$, if

$$x_0 = \begin{bmatrix} 1/6 & 1/2 & 0 & 0 & 1/3 & 0 \end{bmatrix}^T$$

then one sixth of the people are at webpage 1, one half at webpage 2, none are at webpages 3, 4 and 6 and one third are at webpage 5. Now, imagine the people all surf the web at random, so they move from the webpage they are at, at random, to another webpage, by clicking a link in the page where they are. The new distribution will then be

$$x_1 = Ax_0.$$

They all then move at random to a new website by clicking again on some link. The new distribution will be

$$x_2 = Ax_1.$$

Etc. If the distribution settles down, it will settle down to a distribution $x$ that satisfies

$$x = Ax.$$

In other words, the equilibrium distribution will be the same as the importance vector; the importance of a webpage will be equal to the limiting proportion of people who are on that page.

**Dealing with Non-stochasticity.** Suppose there is a webpage that doesn't link to any other webpage. (See the figure on the left in Figure 8.3 in the text.) In terms of the matrix, this would mean there is a column all of whose entries are equal to 0's, so the matrix is not stochastic. What do the people on this webpage do every iteration?

- If we assume they stay on that page then, in order for $x_1 = Ax_0$ to be the distribution of people at the next iterate, we need to insert a 1 on the diagonal in the corresponding column of $A$. Notice this makes the matrix stochastic. If we do this, and other webpages link to this page, then this webpage acts as a sink; whenever anyone gets there they never leave. This means the importance vector will weight this webpage with a 1 and all other webpages with a 0. This is not very informative.
- So, instead we imagine that when a person comes to this page, they then jump at random to any other webpage. That corresponds to making every entry in the corresponding column of $A$ be equal to $1/n$. This also makes the matrix stochastic and is probably closer to reality and gives us a more meaningful importance vector.

Let's call this modified version of $A$, the matrix $B$.

**Dealing with Cyclic Paths.** If there is a cyclic path as depicted on the right in Figure 8.3 in the text (with or without the webpage 1 linking to the cycle) the equilibrium distribution may not be unique and an initial distribution may not settle down to an equilibrium distribution (why not?). This means the eigenvalue corresponding to 1 will not be one-dimensional. To deal with this, we can modify $B$ by taking a weighted average of $B$ and a rank one matrix

$$C_{u,\alpha} = \alpha A + (1-\alpha)ue^T$$

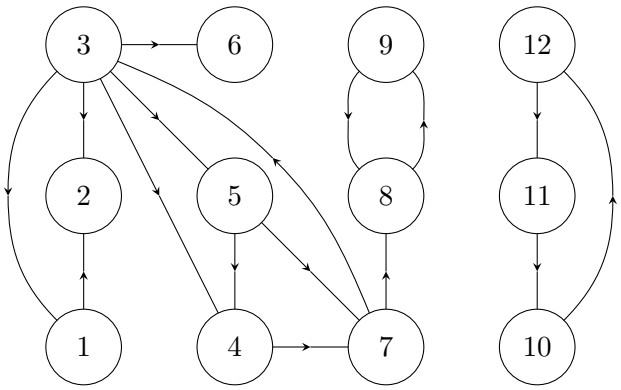
where $e = [1, 1, \ldots, 1]^T$ and the entries in $u$ are all non-negative and sum to 1. This corresponds to everyone who is surfing the web having probability $1-\alpha$ of jumping to a website according to the distribution in $u$.

**Finding the Importance Vector.** Once we have a stochastic matrix $C_{u,\alpha}$ that has no cycles, we can start with any initial distribution $x_0$ (i.e. any vector with all non-negative entries that sum to 1) and keep multiplying it by $C_{u,\alpha}$ to get $x_k = C_{u,\alpha}^k x_0$. Notice, this is a simplified version of the power method; we don't need to normalize the iterates at every stage, since the stochasticity of $C_{u,\alpha}$ ensures all the iterates have the same $l_1$-norm as $x_0$. The theory of stochastic matrices guarantees the iterates converge and thereby proves that the eigenvalue 1 is the eigenvalue of largest magnitude. In fact, it can be shown that all other eigenvalues have magnitude less than $\alpha$. This indicates that the algorithm should converge faster when $\alpha$ is small. However, that places less weight on linkages between webpages and more weight on people jumping to a random point on the webpages.

## Tasks to Complete

(1) Write a matlab file that implements the page rank algorithm and finds the importance vector associated with a link matrix $L$ that is stochastic and has no cycles. Write your file flexibly, so that it will work whatever size $L$ has.

(2) **Toy Example:** Consider the toy example of a network of 12 websites depicted below.




(a) Write down the link matrix $A$.
(b) Modify the link matrix so it has no columns all of which are 0. This is the matrix $B$.

(c) Modify $B$ so that it has no cyclic paths. This is the matrix

$$C_{u,\alpha} = \alpha B + (1-\alpha)ue^T$$

where $e = [1, 1, \ldots, 1]^T$.

(d) Run your page rank m-file on $C_{u,\alpha}$. Explore the page rankings you get for different choices of $\alpha$ and $u$. Explain. Does the importance vector depend on the vector $u$ and/or $\alpha$? How and why? Does the rate of convergence depend on $u$ and/or $\alpha$? How and why?

(e) Try running your page rank file with the matrix $A$. Explain your results. Does it matter what initial distribution you start with?

(f) Try running your page rank file with the matrix $B$. Explain your results. Does it matter what initial distribution you start with?

(3) Posted on blackboard under Content is a matlab file that contains the incidence matrix $M$ of a small network of 1000 webpages. In other words $M_{ij} = 1$ if there is a link on website $j$ to website $i$, otherwise it is 0. Load this matrix into your workspace in Matlab.

(a) Create the link matrix $A$ from $M$ and then modify $A$ to create $B$ and then $C_{u,\alpha}$. Run your Page Rank m-file to rank the pages in order. Use $\alpha = 0.95$ and $u = (1/1000)[1, 1, \ldots, 1]^T$. List the top 10 pages with their corresponding importances in order.

(b) *Fun part!* Create one or more new websites numbered 1001, 1002, etc to add to the network of websites in (3). Your goal is to get website number 1001 into the top 10% of all websites spending the least amount of money. The creation of any website costs \$1000. In addition, you can pay other websites to link to any one of your webpages. The cost of this depends on the ranking of the website you are paying. If that website has rank $i$, then the cost is $(1000 - i + 1)^2$ dollars. You can add whatever links you want to your own websites for free. Show the cheapest solution you found to get your website into the top 10%, the cost associated with your method, and explain how you decided to do what you did.