Mathematics 204
Lab 4: Google PageRank
Due: Wednesday, December 7 at 10pm

Please work on this lab activity in a group of 2 or 3 students. When you are ready, scan your work into a single PDF and submit through Blackboard with only one submission per group. Just make sure everyone's name is included below.

Names:

This lab has a total of 40 points with point totals given in bold next to the questions.

When I recently googled the phrase "linear algebra," Google told me there were about 179 million results and that the Wikipedia page was the one most likely to be useful to me. How does Google decide to recommend that page above all the others? Well, Google computes a quantity called PageRank for each page on the Internet. Pages with a higher PageRank are deemed to be more valuable. The key to computing PageRank lies in an algorithm that we'll study in this lab.

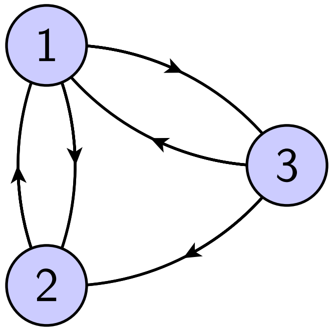
One could determine the importance of a web page by allowing people to vote for their favorite web pages. However, Google wants their rankings to be free of any human bias so it examines the structure of the Internet itself to rank pages. Here's the idea: if my home page is valuable, many other pages will link to it. Of course, I could game the rankings by creating lots of pages that link to my home page. So Google views a page as being more valuable when valuable pages link to it. For instance, if three of my friends link to my home page, that means three people think I have a valuable page. But if `nytimes.com`, `amazon.com`, and `beyonce.com` link to me, that probably means more people will find my page to be valuable so I should have a higher PageRank.

Here's how to compute PageRank. Each webpage has a certain amount of PageRank that we denote by x_i. Each page divides its PageRank into equal portions, one for each outgoing link. Each page then gives one portion of its PageRank to each page it links to. A page's PageRank is then the sum of all the PageRank it receives from pages that link to it. Let's look at an example.

If you visit the page `http://gvsu.edu/s/0To`, you will find some Sage code that will be helpful for this lab. Be sure to evaluate the first cell and then don't use that cell anymore.

1. Suppose that our Internet only has three pages with links as shown below. Page 2 only links to page 1 so it gives all of its PageRank x_2 to Page 1. Page 3 has two outgoing links so it divides its PageRank x_3 into two equal pieces and gives half its PageRank $x_3/2$ to Page 1. The PageRank x_1 is therefore

$$x_1 = x_2 + \frac{1}{2}x_3.$$



[2] Find similar expressions for x_2 and x_3.

[2] If we write $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$, find the "Google matrix" G such that $\mathbf{x} = G\mathbf{x}$.

[2] Notice that the equation $\mathbf{x} = G\mathbf{x}$ is the condition that $\mathbf{x}$ be an eigenvector of G. What is the associated eigenvalue?

[2] Find the vectors $\mathbf{x}$ that satisfy $\mathbf{x} = G\mathbf{x}$.

[2] Notice that the solutions form a 1-dimensional subspace. Since we're just interested in the relative size of the entries, we can choose any one of them as the PageRank vector. Which of the three web pages has the highest PageRank and which has the smallest?

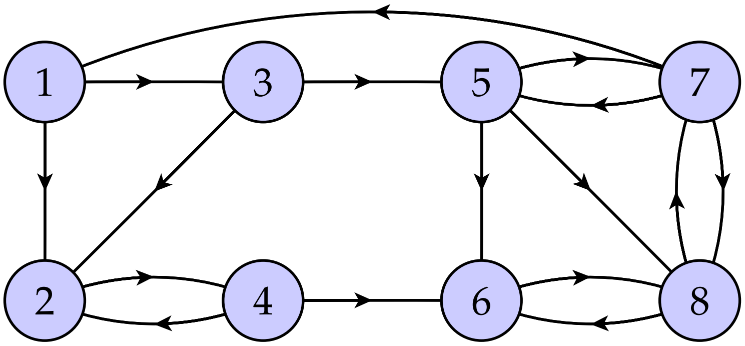
[2] Explain why G is a stochastic matrix. Why will this always be the case for any internet, assuming that each page has outgoing links?

[4] Does G satisfy the hypothesis of the Perron-Frobenius theorem? Explain why or why not. What does this tell us about the behavior of a Markov chain constructed using G?

[4] The first cell on the page of Sage cells loaded a function `markov(A, x0, n)` that prints n vectors in the Markov chain defined by a stochastic matrix A and initial state vector $\mathbf{x}_0$. Starting with the initial state vector $\mathbf{x}_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, generate a Markov chain with 20 vectors $\mathbf{x}_1, \ldots, \mathbf{x}_{20}$. What do you notice about the vectors $\mathbf{x}_k$? How does this demonstrate the Perron-Frobenius theorem?

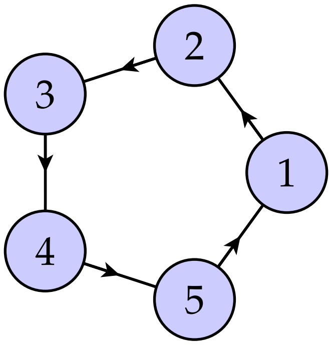
Comment: Google currently knows about 100 trillion (10^{14} pages) so the real Google matrix is 100 trillion $\times$ 100 trillion. That's big, but real applications of linear algebra frequently use huge matrices. Finding the PageRank vector by finding a basis for the eigenspace E_1 using row reduction is not computationally feasible. However, the Perron-Frobenius theorem tells us that any Markov chain will converge to the PageRank vector!

2. Consider the Internet shown below.



[4] The Google matrix for this internet was defined to be `G8` when you evaluated the first cell. Create a Markov chain with a sufficient number of terms to converge. What do you find for the PageRank vector? Which page is most valuable? Which page is least valuable?

3. **[3]** Now consider the internet shown below and find its Google matrix G.



[4] What happens to a Markov chain that begins with the initial state $\mathbf{x}_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$?

[3] By inspecting this internet, state what you feel the PageRank vector should be. Is any page more important than the others?

4. As the last example shows, the Perron-Frobenius theorem may not always apply given the structure of the internet. This means that a Markov chain may not converge or it may converge to a vector with zero entries. For this reason, Google modifies the Google matrix so that the Perron-Frobenius theorem applies. Here's what happens.

   If there are n web pages, define the matrix

$$H_n = \begin{bmatrix} \frac{1}{n} & \frac{1}{n} & \dots & \frac{1}{n} \\ \frac{1}{n} & \frac{1}{n} & \dots & \frac{1}{n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n} & \dots & \frac{1}{n} \end{bmatrix}.$$

   This would correspond to an Internet in which every page is linked to every other page, including itself. We now define a modified Google matrix G' by choosing a parameter α, which is a trade secret but is suspected to be close to $\alpha = 0.85$, and defining

$$G' = \alpha G + (1 - \alpha)H_n,$$

   or, if $\alpha = 0.85$,

$$G' = 0.85G + 0.15H_n.$$

   In other words, G' is obtained by mixing 85% of G with 15% of H_n. You may use the Sage command `modified_google_matrix(G)` to produce G'.

   [2] Go back and revisit the Internet having five pages linked together in a cycle. Construct the modified Google matrix G' and explain why the Perron-Frobenius theorem now applies.

   [2] What do you now find for the PageRank vector using G'? Does this agree with what you think the PageRank vector should be?

[2] Revisit the second Internet whose Google matrix is `G8`, construct its modified Google matrix, and its PageRank vector. Which page is most important and which is least important? Do the results using the modified Google matrix differ significantly from the original results obtained just using `G8`?