Unit 3 Project

1 Submission

Please create a single LaTeX document with your solutions to the five assigned problems. Show all your work. Make sure your name is on the document and upload the .tex file to the course webpage when you are done.

2 Prerequitites

Please read section 4.9 *Application to Markov Chains* in your text before starting this project.

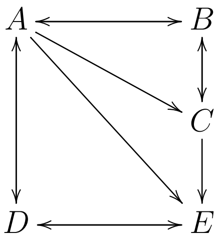
3 Markov Chains and Google PageRank

When Google went online in the late 1990s, one thing that set it apart from other search engines was that its search results listings always seemed to deliver the "good stuff" up front. With other search engines, you often had to wade through screen after screen of links to irrelevant web pages that just happened to match the search text. Part of the magic behind Google is its PageRank algorithm, which quantitatively rates the importance of each page on the web, allowing Google to rank the pages and thereby present to the user the more important (and typically most relevant and helpful) pages first.

Understanding how Google ranks its search results is essential for anyone designing a webpage that they want people to access frequently, since getting listed first in a Google search leads to many people looking at your page. With Linear Algebra, you can understand the first and best-known search algorithm used by Google, now a trillion dollar company!

The PageRank algorithm, developed at Stanford University in 1996 by Google founders Sergey Brin and Larry Page, ranks webpages in Google search results. The basic idea is that a webpage's PageRank should depend on how many other pages link to it, how many pages link to those pages, and so on. A page with many inbound links is likely important, and a page with many inbound links from pages which themselves have many inbound links (and so on) is even more likely to be important.

Example 1: Consider five webpages A, B, C, D, and E that link to each other according to the graph below. For instance, the arrow from A to C indicates that page A links to page C, and the two-way arrow between A and B indicates that A and B link to each other.



The PageRank algorithm assigns ranks $r(A)$, $r(B)$, $r(C)$, $r(D)$, and $r(E)$ that are all between 0 and 1 (inclusive) and sum to 1.

Each page equally distributes its own PageRank along its outbound links. For example, page B has two outbound links, so page B "donates" $\frac{r(B)}{2}$ to both page A and page D. Page D also gives half its PageRank to page A, so the PageRank of A satisfies:

$$r(A) = \frac{r(B)}{2} + \frac{r(D)}{2}$$

Since page B has inbound links from page A (which has four total outbound links) and page C (which has two total outbound links), the PageRank of page B satisfies:

$$r(B) = \frac{r(A)}{4} + \frac{r(C)}{2}$$

The full list of equations is:

$$r(A) = \frac{r(B)}{2} + \frac{r(D)}{2}$$

$$r(B) = \frac{r(A)}{4} + \frac{r(C)}{2}$$

$$r(C) = \frac{r(A)}{4} + \frac{r(B)}{2}$$

$$r(D) = \frac{r(A)}{4} + r(E)$$

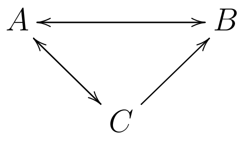
$$r(E) = \frac{r(A)}{4} + \frac{r(C)}{2} + \frac{r(D)}{2}$$

The unique solution of this system of linear equations for which the sum of the PageRanks is 1 is:

$$\begin{bmatrix} r(A) \\ r(B) \\ r(C) \\ r(D) \\ r(E) \end{bmatrix} = \begin{bmatrix} 0.211 \\ 0.105 \\ 0.105 \\ 0.316 \\ 0.263 \end{bmatrix}$$

We always normalize our solution vector so that the sum of the PageRanks is 1.
Interpreting these results, we see that page D would show up first in a search since it has the highest rank (0.316), followed by page E, with a rank of 0.263, and so on. It is interesting to note that even though page E has the most inbound links, it doesn't have the highest PageRank. This is because page E only has one outbound link, to page D, so page D receives the entirety of page E's PageRank, plus a small contribution from page A.

Exercise 1: (6 pts) Use the technique of the example above to find the PageRanks of pages A, B, and C in the graph below.



Example 1 (con't): Returning to our example, we may write the system of linear equations we found in matrix form:

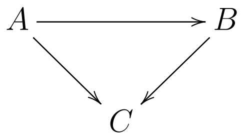
$$\begin{bmatrix} 0 & 0.5 & 0 & 0.5 & 0 \\ 0.25 & 0 & 0.5 & 0 & 0 \\ 0.25 & 0.5 & 0 & 0 & 0 \\ 0.25 & 0 & 0 & 0 & 1 \\ 0.25 & 0 & 0.5 & 0.5 & 0 \end{bmatrix} \vec{r} = \vec{r}, \qquad \text{where} \quad \vec{r} = \begin{bmatrix} r(A) \\ r(B) \\ r(C) \\ r(D) \\ r(E) \end{bmatrix}$$

Denote the 5×5 matrix above by P. The columns of P describe the outbound links from each webpage (the first column of P, for example, indicates that A has outbound links to B, C, D, and E, and equally distributes its PageRank among them). Because of this, the sum of the entries in each column of P is 1. In other words, P is a stochastic matrix, and can be thought of as the *transition matrix* of a Markov chain. Additionally, the vector $\vec{r}$ of PageRanks we are searching for is an eigenvector of P corresponding to the eigenvalue $\lambda = 1$. Since the entries of $\vec{r}$ sum to 1, $\vec{r}$ is actually a steady-state vector of P!

This Markov connection is not accidental. Suppose we start on a random page (A through E) and begin randomly clicking links. If $\vec{x}_0$ in $\mathbb{R}^5$ is the probability vector for our initial location (i.e., the components of $\vec{x}_0$ are the probabilities that we begin at page A, or page B, etc.), then $P^{50}\vec{x}_0$ is the probability vector for our location after 50 random clicks. As we'll see later, in "nice" cases the sequence of vectors $P^k\vec{x}_0$ approaches the steady-state (PageRank) vector $\vec{r}$ as $k \to \infty$, no matter what $\vec{x}_0$ is. In this way, we can interpret the PageRank vector as a vector of probabilities for our location after many random clicks. In Example 1, $r(C) = 0.105$ means that if we start on a random page and click a large number of random links, then there is about a 10.5% chance that our final destination is page C.

We'll see later that this realization makes it possible to approximate the PageRank vector in cases where it is infeasible to calculate the PageRank vector directly, but we are still left with a question: can we always frame the problem of finding PageRanks in terms of finding the steady-state vector of a related stochastic matrix? The following example will help settle the question.

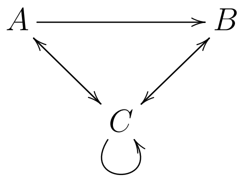
Example 2: Consider the simple graph of links below:



The corresponding matrix is:

$$P = \begin{bmatrix} 0 & 0 & 0 \\ 1/2 & 0 & 0 \\ 1/2 & 1 & 0 \end{bmatrix}$$

The third column of P is the zero vector, so P is *not* the transition matrix of a Markov chain! If we access a random page (A, B, or C) and click links randomly, then we always end up stuck at C after at most two clicks. To remedy this, **whenever we have a page that doesn't link to any other pages, we modify the link diagram so that the page links to *all* pages, even itself.** In this example, we draw arrows from C to all three pages:

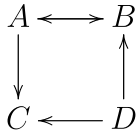


The new matrix is:

$$\widetilde{P} = \begin{bmatrix} 0 & 0 & 1/3 \\ 1/2 & 0 & 1/3 \\ 1/2 & 1 & 1/3 \end{bmatrix}$$

We call $\widetilde{P}$ the *altered transition matrix*. In general, if there are n webpages, then we form $\widetilde{P}$ from P by replacing any columns of zeroes by the vector $(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n})$. An altered transition matrix is always a stochastic matrix (why?), and should be used to find PageRanks if at least one webpage does not link to other pages.

Exercise 2: (12 pts) Consider the link diagram below.

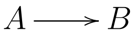


a. (4 pts) Show that the method you used in Exercise 1 to find PageRanks doesn't work here. [Note that D has no inbound links, so $r(D) = 0$.]

b. (2 pts) Find the altered transition matrix $\widetilde{P}$.

c. (6 pts) Find the PageRank vector by finding the unique steady-state vector of $\widetilde{P}$.

Exercise 3: (8 pts) Consider the very simple link diagram below. Ralph has a $\frac{2}{3}$ chance of starting at page A and a $\frac{1}{3}$ chance of starting at page B. Once per minute, he clicks on a random link, if possible.



a. (2 pts) What is the probability that Ralph will be on page B after four minutes?

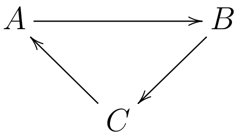
b. (6 pts) Now suppose that whenever Ralph accesses page B, after one minute he will randomly (with equal probability) jump to either page A or stay on page B. Find the probability that Ralph will be on page B after four minutes.

Finding a steady-state vector of an altered transition matrix works well when the number of webpages is small, but is completely unfeasible when attempting to rank millions of webpages. In such a case, the best we can do is to approximate the PageRank vector. The following definition and theorem from your text help us to do this.

Definition: A stochastic matrix P is *regular* if for some k every entry of P^k is positive.

Theorem: If P is a regular stochastic matrix, then P has a unique steady-state vector $\vec{r}$. Additionally, if $\vec{x}_0$ is a probability vector, then the Markov chain $\{P^k\vec{x}_0\}$ approaches $\vec{r}$ as $k \to \infty$.

Exercise 4: (8 pts) Consider the graph of links below.



The PageRank vector is, unsurprisingly, $\vec{r} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$.

a. (2 pts) Find the altered transition matrix $\widetilde{P}$.

b. (4 pts) Let $\vec{x}_0 = \begin{bmatrix} 0.2 \\ 0.3 \\ 0.5 \end{bmatrix}$. Find $\widetilde{P}\vec{x}_0$, $\widetilde{P}^2\vec{x}_0$, $\widetilde{P}^3\vec{x}_0$, and $\widetilde{P}^{100}\vec{x}_0$.

c. (2 pts) Does the sequence $\{\widetilde{P}^k\vec{x}_0\}$ approach $\vec{r}$?

If the altered transition matrix $\widetilde{P}$ is not regular, then $\widetilde{P}$ may not have a unique steady-state vector, and the limit of a Markov chain $\{P^k\vec{x}_0\}$ may not be a steady-state vector of $\widetilde{P}$. To fix this, we build a new matrix out of of $\widetilde{P}$ whose entries are all positive.

If $\widetilde{P}$ is $n \times n$, define M to be the $n \times n$ matrix whose columns are all $(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n})$, and let $0 < \alpha < 1$. Define the *Google matrix* G_α as follows:

$$G_\alpha = \alpha\widetilde{P} + (1-\alpha)M$$

Theorem: The Google matrix G_α is a stochastic matrix whose entries are all positive.

Proof: Since $\widetilde{P}$ was stochastic, the columns of $\widetilde{P}$ sum to 1, so that the columns of the Google matrix sum to $\alpha * 1 + (1-\alpha) * (\frac{1}{n} + \cdots + \frac{1}{n}) = \alpha + (1-\alpha) * n * \frac{1}{n} = 1$.

Since $0 < \alpha < 1$, α and $1-\alpha$ are positive. And since the entries of $\widetilde{P}$ were nonnegative, the entries of G_α have the form (pos.)(nonneg.) + (pos.)(pos.) = (nonneg.) + (pos.), and hence are positive. ■

The theorem implies that the Google matrix G_α is a regular stochastic matrix, and thus for any probability vector $\vec{x}_0$, the Markov chain $\{G_\alpha^k \vec{x}_0\}$ converges to the unique steady-state vector of G_α. If α is close to 1, then G_α is very close to $\widetilde{P}$, and its steady-state vector is close to a steady-state vector of $\widetilde{P}$ (assuming $\widetilde{P}$ has a unique steady-state vector, then this is the PageRank vector $\vec{r}$). If $\vec{x}_0$ is any probability vector, then for a large value of k, $G_\alpha^k \vec{x}_0$ should be a good approximation to the PageRank vector $\vec{r}$. The convergence of $\{G_\alpha^k \vec{x}_0\}$ to the steady-state vector of G_α is typically quicker for smaller values of α, but of course for larger values of α the steady-state vector of G_α is usually closer to a steady-state vector of $\widetilde{P}$. Google hasn't publicly revealed what value of α it uses, but it's thought to be around 0.85.

Exercise 5: (6 pts) Refer back to Exercise 4.

a. (2 pts) Find the Google matrix $G_{0.5}$.

b. (2 pts) Verify that in this case the steady-state vector of $G_{0.5}$ is the same as the PageRank vector $\vec{r}$.

c. (2 pts) Note that $G_{0.5}^5 \begin{bmatrix} 0.2 \\ 0.3 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 0.332 \\ 0.339 \\ 0.329 \end{bmatrix}$. Explain why it's not surprising that this vector is very close to the PageRank vector $\vec{r}$.