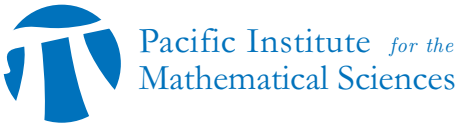

SUBSPACES PRESERVED BY A MATRIX:

PACIFIC INSTITUTE FOR THE MATHEMATICAL SCIENCES VIRTUAL EXPERIMENTAL MATHEMATICS LAB FINAL REPORT

STUDENTS: KAREN FLETCHER, SOHAM LAKHI, BASSAM NIMA

GRADUATE MENTOR: JASON GILBERT

SUPERVISORS: STEVEN RAYAN, ALEX WEEKES, CURTIS WENDLANDT

1. Introduction

Prior to Google search engines often found and ranked websites based on the incidence of keywords which could bury important results below less relevant ones, leaving the user to look through pages of results. Google's PageRank algorithm instead relies of the structure of the internet, hypothesizing that an important page will have many links to it while an unimportant page will receive far fewer links from other websites, assuming the number of links reflects the degree of human interest. The goal of this algorithm was to improve the quality of the search results. Their work seeks to represent the structure of the internet mathematically, and create a mathematical measure of website significance, separate from user behaviour or website content. Mathematical representation opens the problem to computational analysis, allowing us to expand our explorations. A number of mathematical tools are used in the problem; the PageRank algorithm seeks to represent the relationship between pages in a graph, with pages represented by nodes and links represented by edges; it then constructs a transfer matrix that describes the likelihood of randomly navigating from one page to another; and the eigenspace of this matrix encodes information about the structure of the graph.

2. Problem

We were interested in what we can discover about the structure of the internet based on its transfer matrix. We also wanted to know whether we could understand typical user behaviour by examining a Markov Chain which describes random walks via the transfer matrix. We were also interested in comparing the computational advantages of each approach.

3. Mathematical Tools

3.1. **Background and Initial Work.** We began by seeking to learn about and better understand the mathematical tools being used in the problem. We did a literature review to understand PageRank and why it is significant as well as understanding generalized eigenvectors, Jordan chains, and Jordan normal form. Of the articles we reviewed, the most impactful to our work were [1, 2, 3].

3.1.1. *Graphs and Markov Matrices.* The adjacency matrix of a graph can be used to compute its Markov matrix. Given a finite, directed graph $G = (V, E)$, with a set of nodes $V = \{v_1, ..., v_n\}$, and a set of edges $E = \{E_1, \ldots, E_n\}$ the adjacency matrix elements A_{ij} are 1 when an edge exists from v_i directed to v_j and 0 otherwise. Evidently, the diagonal of the adjacency matrix represents the self-loops in the graph, thus the network representing the internet should have a zero-diagonal, as that would model a user refreshing the page, which is unlikely to meaningfully reflect the structure of the internet.

Suppose G is given by Figure 1.

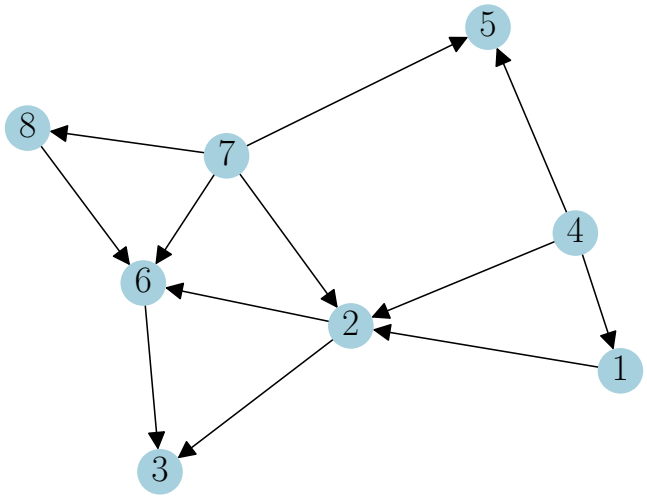

FIGURE 1. A sample directed graph with 8 nodes

Then the adjacency matrix of the graph is

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

The Markov matrix can be computed by dividing each of the adjacency matrix elements by the sum of their row. However, nodes with no outgoing lines (dangling nodes) result in a row of zeroes, which proves to be problematic when computing the Markov matrix $\mathbf{P}$. To rectify this, the zero-rows are replaced with rows of $1/n$ for an $n \times n$ matrix.

Then the Markov matrix of the example above is given by

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 1/2 & 0 & 0 \\ 1/8 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 & 0 & 0 \\ 1/8 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 0 & 0 & 1/4 & 1/4 & 0 & 1/4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

3.1.2. *Random Surfer Model.* One problem that arises is in an internet with isolated loops. A loop consisting of a few web pages (such as a website with no incoming or outgoing external links) results in a rank sink since the rank accumulates within the loop. A random walk initializing within the loop would remain in the loop indefinitely. To solve this, a random surfer model is implemented wherein each iteration there is a probability α that the user accesses a hyperlink and probability $1-\alpha$ that the user navigates to another site, such as by searching or typing in a URL. This can be computed with Equation 1 where $\mathbf{P}(\alpha)$ is the new random surfer Markov matrix, $\mathbf{P}$ is the original Markov matrix, n is the number of web pages and $\mathbf{e}\mathbf{e}^T$ is an $n \times n$ matrix of ones.

$$\mathbf{P}(\alpha) = \alpha\mathbf{P} + \frac{(1-\alpha)}{n}\mathbf{e}\mathbf{e}^T \tag{1}$$

Probability $1-\alpha$ is typically taken as a small value such as 0.15, normally for the sake of computational advantage, as we will explain later.

3.1.3. *Dominant Eigenvalue and Stationary Distribution.* The dominant eigenvalue, that is, the eigenvalue with the largest modulus, will always be real and equal to 1 for a Markov matrix.

$$|\lambda_1| = 1 > |\lambda_2| \geq ... \geq |\lambda_n| \tag{2}$$

The dominant eigenvalue's corresponding eigenvector, denoted by $\boldsymbol{\pi}$ represents the stationary distribution.

$$\boldsymbol{\pi}^T = \boldsymbol{\pi}^T \mathbf{P}(\alpha) \tag{3}$$

When the power method is applied with $\mathbf{P}(\alpha)$, such as $\mathbf{P}(\alpha)^k$ for a very large integer k, the resulting matrix multiplication of an initial distribution converges to the stationary distribution. In other words, $\boldsymbol{\pi}$ gives the likelihoods $\boldsymbol{\pi}_i$ of ending up at page i after a very long time, hence, it can be used to rank each page. And since this represents a probability distribution of the nodes, $\sum_i \boldsymbol{\pi}_i = 1$. Given the ergodicity of the random surfer model, a stationary distribution is guaranteed to exist. If multiple eigenvalues are equal to 1 then the system may oscillate between the dominant states, however for the random surfer model, $\mathbf{P}(\alpha)$ is aperiodic and only one eigenvalue of 1 exists.

3.1.4. *Subdominant Eigenvalues and Convergence.* The second eigenvalue, that is, the eigenvalue with the second largest modulus denoted by λ_2, can be useful in determining convergence rates. For the random surfer model, λ_2 is limited by α.

$$|\lambda_2| \leq \alpha \tag{4}$$

Since $|\lambda_1| = 1$, the eigengap is given by $|\lambda_1| - |\lambda_2| = 1 - |\lambda_2| \geq 1 - \alpha$. [4]

The power method may be used to explore the long term behaviour of the Markov matrix. The rate of convergence of the power method is given by

$$\text{rate of convergence} = \left|\frac{\lambda_2}{\lambda_1}\right|^k \tag{5}$$

It follows that applying the power method to the random surfer model gives

$$\text{rate of convergence} \leq \alpha^k \tag{6}$$

3.1.5. *Necessity of Generalized Eigenvectors.* We can find the basis for an n by n square matrix, if it is diagonalizable, that is there exists an invertible matrix P

such that $P^{-1}AP = D$, where D is a diagonal matrix (a matrix where the only non-zero entries occur on the diagonal). To "diagonalize" a matrix means to find D, in order to do this we must first find P. However, P can be constructed by finding the eigenvectors of the matrix A, and then combining the eigenvectors as column vectors into a matrix. When we do this we find that D is the diagonal matrix with the diagonal values being eigenvalues corresponding to the position of the eigenvectors in $\mathbf{P}$. The eigenvectors form a basis for the vector space of the column vectors of $\mathbf{P}$.

In some cases, the unit vectors along the x, y, and z axis are not ideal to describe a particular scenario, for example if we are standing on a floor covered with parallelogram-shaped tiles, rather than square ones. In this scenario we may wish to use the edges of the tiles as our vectors. If we had a previous set of vectors giving us directions in the room we could apply a linear transformation to them in order to convert them to our new basis, we would have one vector that has undergone a transformation, and one that has not. The vector that maintains its direction through the linear transformation is called an eigenvector. Eigenvectors are important because they do not change direction when acted upon by a matrix, but remain in the same direction. For example, for a matrix A and an eigenvector X we can write $AX = \lambda X$, where multiplying the matrix and eigenvector together produces a vector that is a scalar multiple of the eigenvector (the scalar being called an eigenvalue).

3.1.6. *Calculating generalized eigenvectors of transfer matrices.* It is important to note that in order to diagonalize a matrix we require n linearly independent eigenvectors, otherwise they will not span the space. A square matrix that has duplicate eigenvalues is called degenerate, and these matrices cannot be diagonalized, but we can use generalised eigenvectors to form the Jordan Normal Form of a matrix as an analogue to diagonalization. This process gives us a matrix which is as close as possible to being diagonalizable, with zeros at all spots except the diagonal (entries

are eigenvalues) and some positions of the superdiagonal of the matrix (entries are ones).

In order to calculate generalised eigenvectors one first calculates the eigenvectors and identifies those with their algebraic multiplicity greater than their geometric multiplicity. We take these eigenvectors and create Jordan chains for them, In order to get a set of n linearly independent vectors. These subsequent vectors in the Jordan chains are called generalised eigenvectors and, together with the eigenvectors, can be used to find the Jordan Normal Form of the matrix. In order to find the generalized eigenvectors of a matrix A we need to find vectors where

$$(A - \lambda I)^m x_m = 0 \tag{7a}$$

$$(A - \lambda I)^{m-1} x_m \neq 0 \tag{7b}$$

4. Experimental Work

4.1. **Experimental Tools.** The graph illustrated in Figure 1 was used throughout experimentation using various methods including the power method algorithm, Markov matrix sampling, and Markov chain Monte Carlo.

4.2. **Power Method.** In practice, PageRank is computed using an algorithm known as the Power Method, which is a general method for approximating the dominant eigenvector of a matrix. In the vein of studying linear algebraic models via simulation, we chose to implement the power method, computationally test the theoretical convergence properties, and compare the performance of the power method to that of our implementation for Markov chains.

Here we will describe the basic principles of power method, and the expected convergence. Subsequently, we will describe the computational test of the convergence properties.

The power method can be simply stated as an iterative process (Ref[3], p.38),

$$\boldsymbol{\pi}_{k+1} = \boldsymbol{\pi}_k \mathbf{P} \ , \tag{8}$$

where $\boldsymbol{\pi}$ is stationary vector which approximates the dominant eigenvector, $\boldsymbol{\pi}_k$ is the approximation resulting from the k_{th} iteration of the above procedure, and $\mathbf{P}$ is the transition matrix corresponding to the graph being considered.

It should be noted here that for any matrix which satisfies the convergence requirements of the method, which the random surfer matrix does, for any initial vector $\boldsymbol{\pi}_0$, $\boldsymbol{\pi}_k$ will converge the to dominant eigenvector as $k \to \infty$.

As for the convergence of the method, it is known (Ref [3], p.41) that the asymptotic rate of convergence should be

$$R = \left|\frac{\lambda_2}{\lambda_1}\right|^k = |\lambda_2|^k \ , \tag{9}$$

where k is the number of iterations, λ_1 and λ_2 are the largest and second-largest eigenvalues, respectively, and $\lambda_1 = 1$ for the matrices were are interested in, allowing us to eliminate it from the above expression.

4.2.1. *Power Method Convergence.* In order to test the theoretical predictions for the rate of convergence of the power method, we applied the method to our test graph and computed the error in the approximation as a function of the number of iterations.

Here we define the error to be

$$\varepsilon_k = ||\boldsymbol{\pi}_k - v|| = \sqrt{\sum_j \left[(\boldsymbol{\pi}_k)_j - v_j\right]^2} \ , \tag{10}$$

where $\boldsymbol{\pi}_k$ is the approximate stationary distribution after k iterations; v is the dominant eigenvector of the transition matrix $\mathbf{P}$, computed via conventional eigenvector decomposition; and the index j refers to the j_{th} component of each vector. The measure used to define the error is simply the Euclidean norm of the difference between the two vectors; it should be noted that this decision was made for the sake of convenience, without theoretical justification, and that other measures could be used instead.

To compute the rate of convergence of ε, we assume the error takes the form

$$\varepsilon_k = a_2 \cdot (a_1)^k \ , \tag{11}$$

where a_1 and a_2 are arbitrary coefficients. Based on the theory, we would expect that $a_1 = \lambda_2$. To facilitate the determination of the coefficients, we use the expression

$$\log \varepsilon_k = k \log a_1 + \log a_2 = k b_1 + b_2 \ , \tag{12}$$

and estimate $a_1 = 10^{b_1}$ via a least-squares fit of $\log \varepsilon_k$.

One final detail which must be mentioned is that we expect the rate of convergence to vary for different $\boldsymbol{\pi}_0$, even if they all converge. For this reason, the method is applied to a large number of randomly distributed $\boldsymbol{\pi}_0$, and the error for each iteration is computed by averaging over all vectors.

The results of this experiment are shown below in Figure 2.

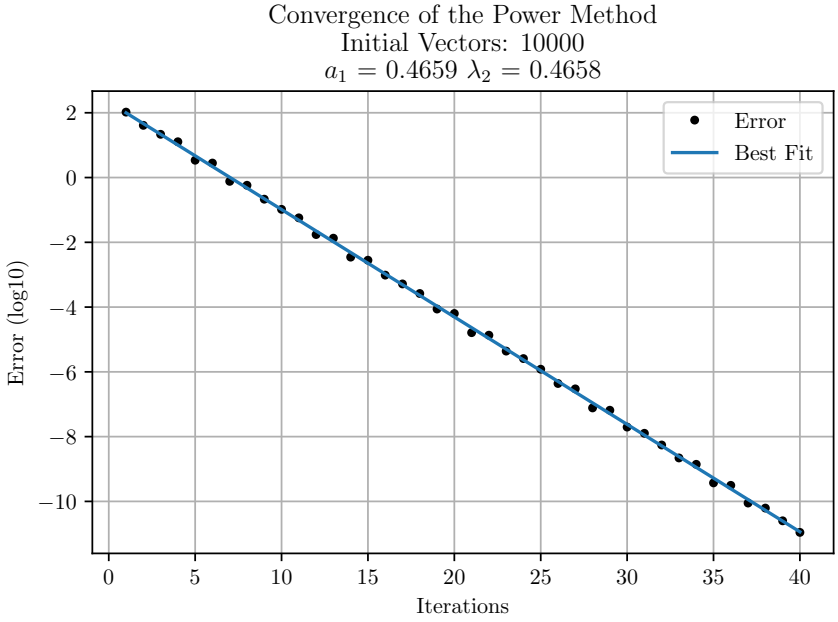

FIGURE 2. Error between approximate stationary distribution and dominant eigenvector, as defined by equation (13). Best fit according to equation (11) shows that $a_1 \approx \lambda_2$, as predicted. Each value is obtained by averaging the error over 10000 initial vectors ($\boldsymbol{\pi}_0$), generated via uniform random distribution.

4.3. **Markov Matrix Sampling Simulation.** Another method to estimate the PageRank of a network is by sampling from the Markov matrix, $\mathbf{P}(\alpha)$, to generate a Markov chain, $X = \{X_1, X_2, ...X_M\}$. An $M = 5000$ Markov chain simulation was done where the first entry, X_1 was set to a uniformly random $x \in \{1, 2, ..., n\}$. Then, X_i was chosen using the X_{i-1}th row vector probability distribution of $\mathbf{P}(\alpha)$. A sample simulation is shown in Figure 3.

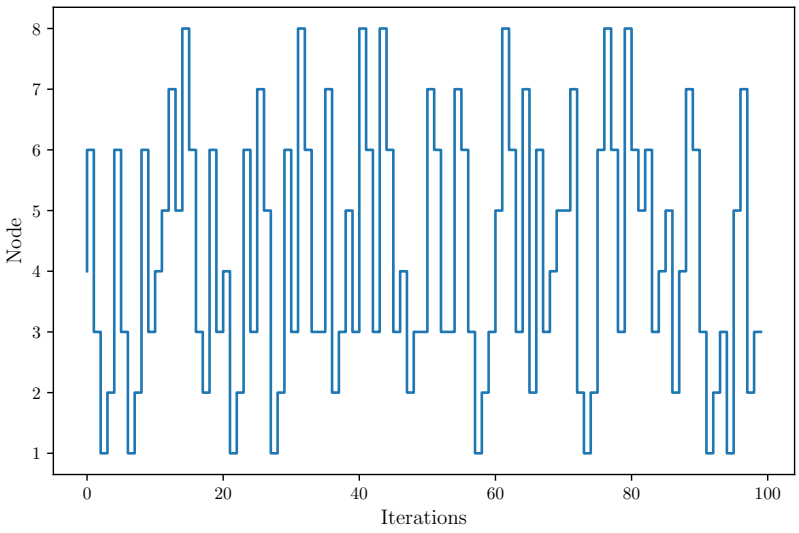

FIGURE 3. The first 100 iterations of a 5000 length Markov chain, sampled from the random surfer model transition matrix of the graph in 1.

An estimation of $\boldsymbol{\pi}$ could be computed by taking the normalized count of the nodes, as demonstrated in Figure 4. Clearly, as the Markov chain length increases, the normalized count approaches the dominant eigenvector

$$\boldsymbol{\pi}^T = (0.08,\ 0.15,\ 0.29,\ 0.06,\ 0.09,\ 0.20,\ 0.06,\ 0.07).$$

At 5000 iterations, the normalized count is

$$\hat{\boldsymbol{\pi}}^{\boldsymbol{T}} = (0.08,\ 0.15,\ 0.30,\ 0.06,\ 0.09,\ 0.20,\ 0.06,\ 0.07),$$

yielding an error of

$$\textbf{error} \approx (0,\ 0,\ 0.03,\ 0,\ 0,\ 0,\ 0,\ 0).$$

The transitions between each node were also accounted for as demonstrated in the matrix plots in Figure 6. At 5000 iterations the matrix plot in Figure 6 (f) closely resembles the matrix plot of $\mathbf{P}(\alpha)$ in Figure 5.

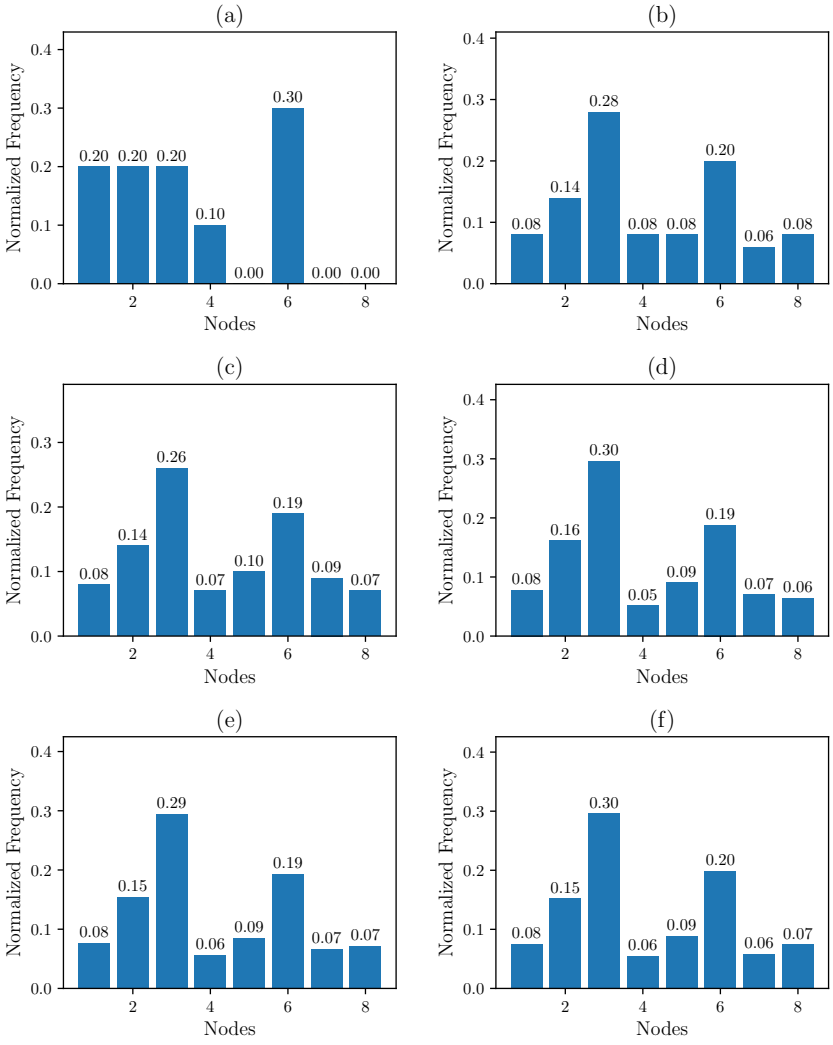

FIGURE 4. Result node distribution from Markov matrix sampling with varying simulation times: (a) 10, (b) 50, (c) 100, (d) 500, (e) 1000, (f) 5000 iterations.

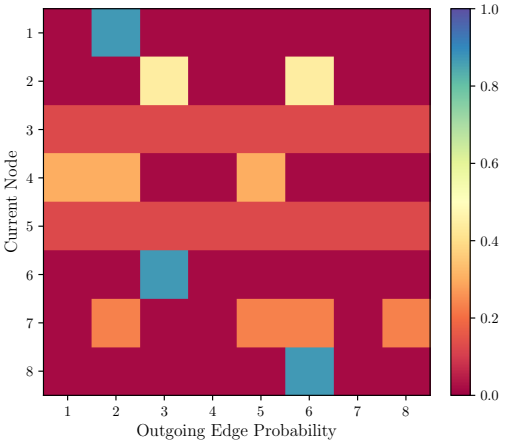

FIGURE 5. The transition matrix of random surfer model applied to the graph from Figure 1, with $\alpha = 0.85$.

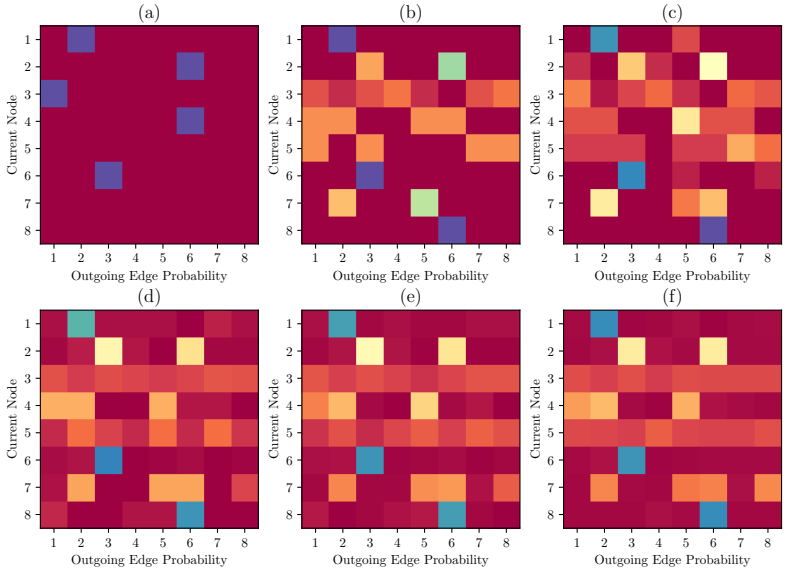

FIGURE 6. Simulated transition matrices from Markov matrix sampling with varying simulation times: (a) 10, (b) 50, (c) 100, (d) 500, (e) 1000, (f) 5000 iterations.

4.4. **Markov Chain Distributions.** Now we explore the distributions of the node counts presented in the previous section, when many of these Markov chains are generated. To illustrate, Figure 7 shows the first 100 iterations of several Markov chains that were used to compute distributions.

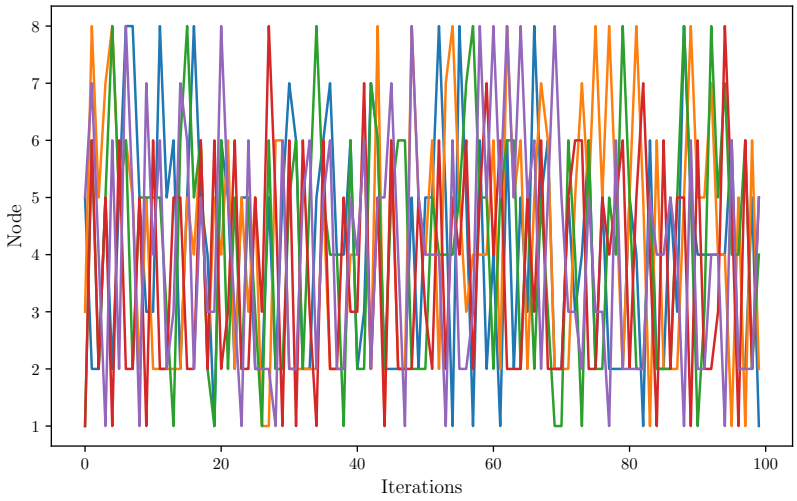

FIGURE 7. Markov chains generated with the methods described in previous sections. For clarity, only 5 of 2000 Markov chains are plotted.

The counts for these Markov chains are shown in Figure 8, where a Gaussian function is fitted using the means and standard deviations. The mean squared error (MSE) of the stationary distribution may be a useful indicator of the spread in the simulation, and eventually was used to measure convergence. We calculate the MSE using

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{\pi}_i - \hat{\boldsymbol{\pi}}_i)^2 \tag{13}$$

where n is the number of nodes, $\boldsymbol{\pi}_i$ is the ith element of the stationary distribution and $\hat{\boldsymbol{\pi}}_i$ is the ith normalized count from the simulation.

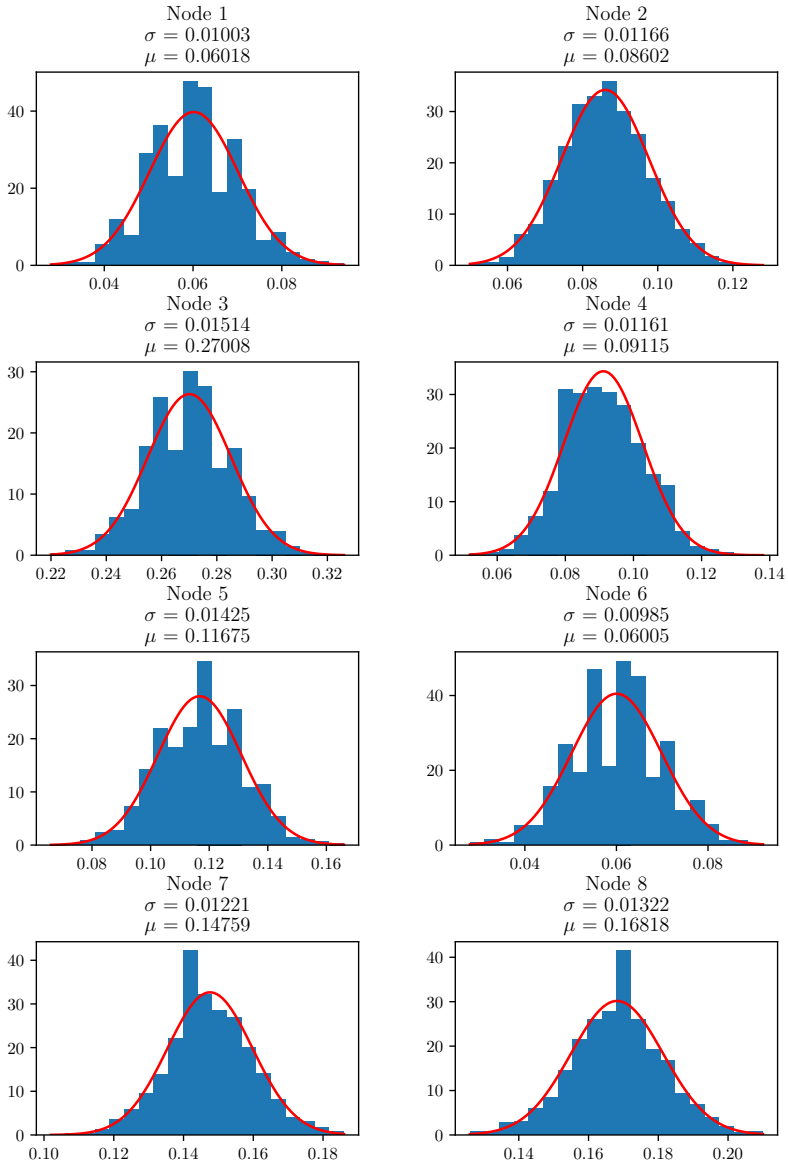

FIGURE 8. Sample distributions of each node's $\boldsymbol{\pi}_i$ estimation with 2000 samples of Markov chains of length 500. A Gaussian function is fitted using the standard deviation σ and mean μ.

When the simulation is run with many samples of Markov chains, we obtain a similar Gaussian distribution of the MSEs, as illustrated in Figure 9. Here a Gaussian if fitted using the distribution's mean, μ, and standard deviation, σ.

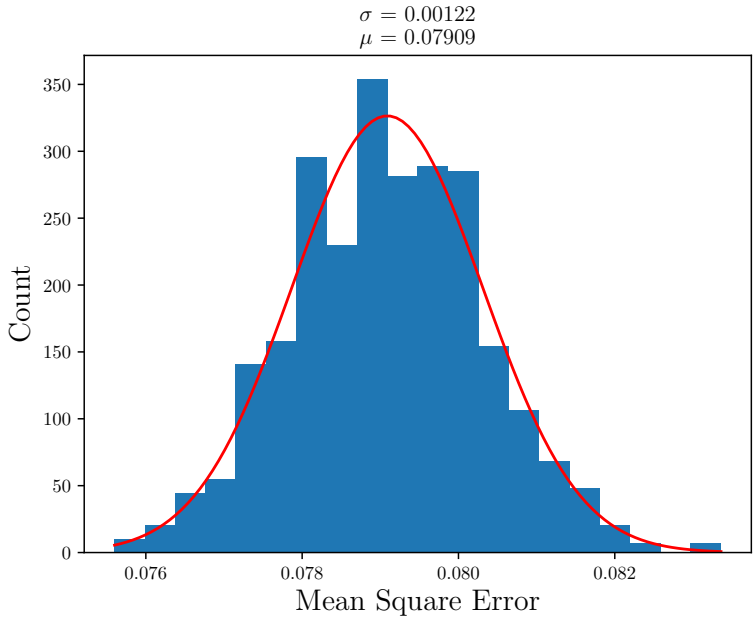

FIGURE 9. The MSE distribution of the previous simulation.

When the length of the simulation is changed, the MSE's mean and standard deviation follows a downward trend, as shown in Figure 10 and Figure 11. Here the standard deviation of MSE is fitted with

$$\sigma = bt^k, \tag{14}$$

and the average MSE is fitted with

$$\mu = ad^t + c, \tag{15}$$

where a, b, c, d, k are fitting parameters and t is the number of iterations in the simulation (*i.e.* Markov chain length).

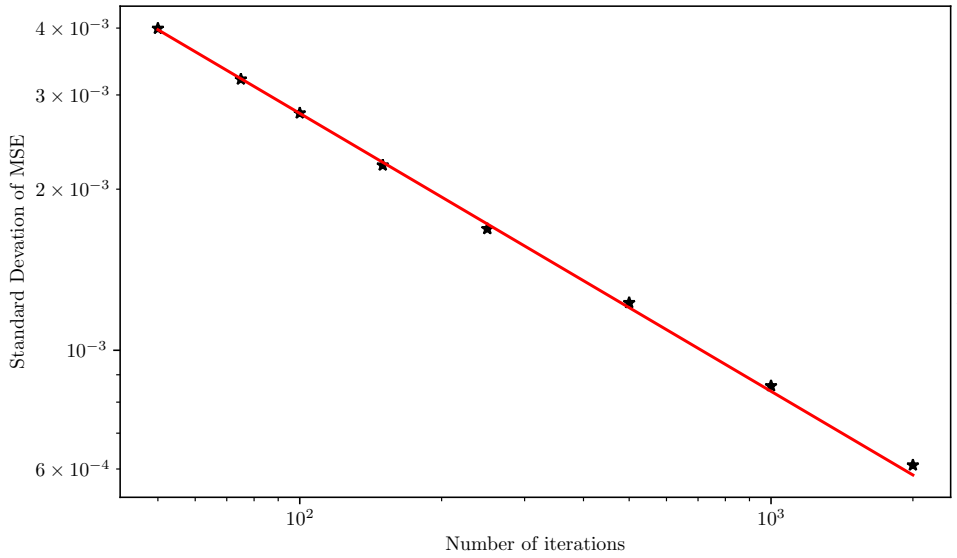

FIGURE 10. The MSE standard deviation versus various simulation times.

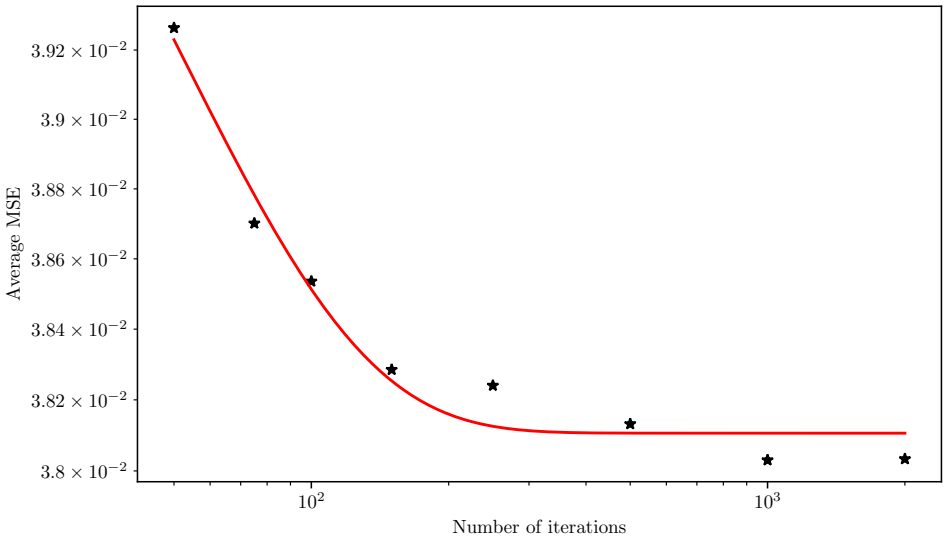

FIGURE 11. The average MSE versus various simulation times.

4.4.1. *Relationship Between MSE Convergence Rates and the Second Eigenvalue.* Naturally, we ask: does the MSE distribution's change over simulation time have a relationship with the second eigenvalue, analogous to the relationship between power method's rate of convergence and the second eigenvalue? This simulation

was run with multiple graphs using multiple simulation lengths to compare the fitting parameters a, b, c, d, k to the second eigenvalue of $\mathbf{P}(\alpha)$. The matrix, $\mathbf{P}(\alpha)$, was chosen by randomly shuffling the rows of the test random surfer model matrix we have used up to now. In other words, the outgoing edges are swapped between nodes. The simulation time was approximately 12 hours for 100 networks with varying simulation times each having 1000 samples of Markov chains.

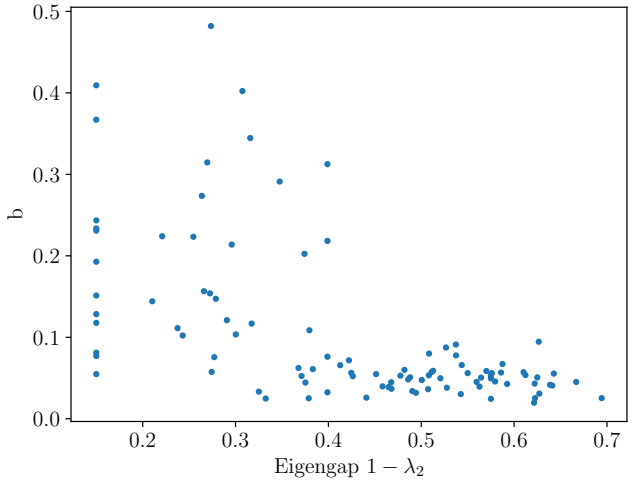

FIGURE 12. b versus the eigengap. Pearson R = -0.593.

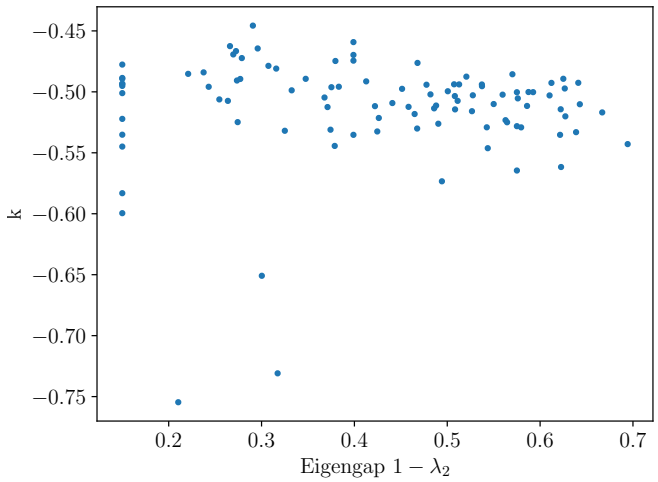

FIGURE 13. k versus the eigengap. Pearson $R = 0.036$.

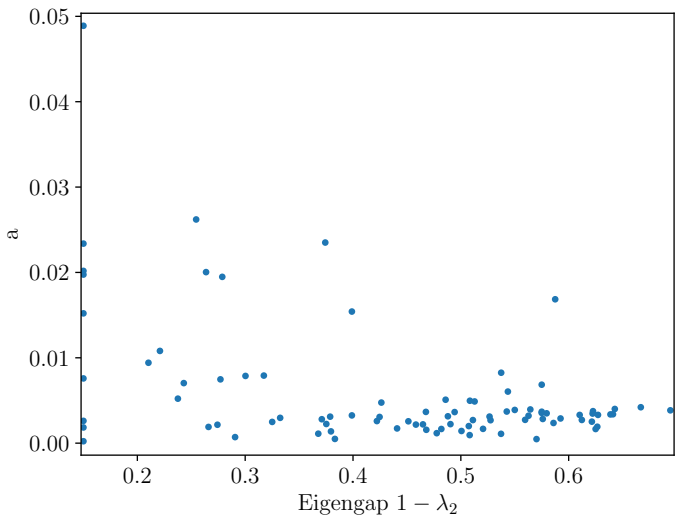

FIGURE 14. a versus the eigengap. Pearson R = -0.166.

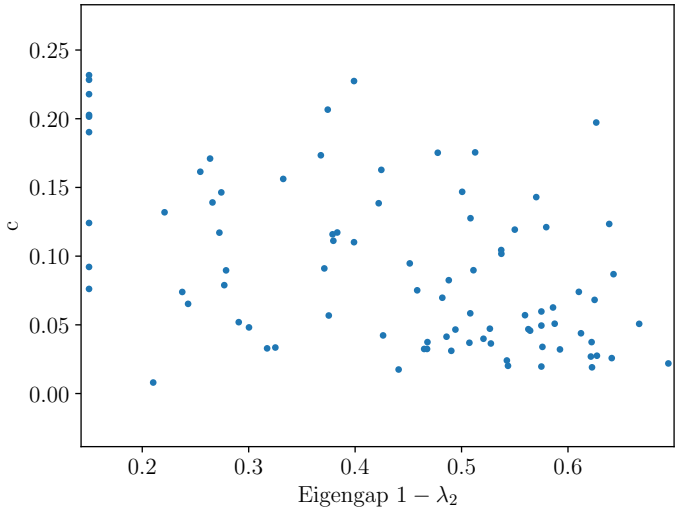

FIGURE 15. c versus the eigengap. Pearson $R = 0.242$.

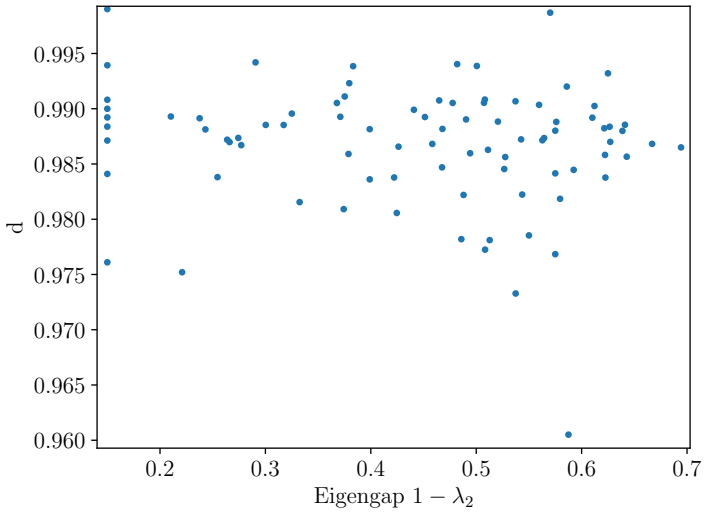

FIGURE 16. d versus the eigengap. Pearson $R = 0.013$.

Regarding the results of these experiments, we may make several interesting observations. First, we note that the eigengaps for all 100 networks and their Markov matrices obeyed Equation 4, since $1 - \alpha = 0.15$ and $1 - \lambda_2 \geq 0.15$. Second we note that there is no obvious correlation between the eigengap and the convergence rate of the simulation, as the above figures show either a relatively random distribution of parameters with respect to the eigengap, or that the parameter is invariant with respect to the eigengap.

Next, we note that the parameter d from equation (15), used to fit the Markov chain MSE, is analagous to the parameter a_1 from equation (11), which we used to quantify the convergence of the power method. This allows some comparison of the convergence between the two. From our experiments, we found the rate of convergence for the power method was close to λ_2, while the rate of convergence for the Markov simulations was roughly 0.985, regardless of eigengap, based on an inspection of Figure 16. This shows that it takes far fewer iterations to get an accurate approximation using the power method. We are unable to provide a precise statement of the computational advantage, as it would require more knowledge of the cost of generating each step of the simulation, but we can say with some

confidence that the requirement to average many simulations to get a reasonable approximation via simulation would ensure the relative superiority of the power method.

5. Conclusion

To summarize, we explored the subject of subspaces preserved by a matrix, specifically the subspace spanned by the eigenvectors of Markov matrices. In this process, we sought to understand generalized eigenvectors, their applications to the subject of ranking webpage significance, and how modelling the problem in terms of linear algebra compares to modelling in terms of Markov chains.

To better understand each of these concepts, we developed programs which implemented each approach, verified that the models were consistent, and attempted to compare the efficiency of different approaches to ranking webpages.

From these studies we found that the two models were, in fact, consistent, and that the power method appeared to have much preferable convergence properties, explaining why it is such a popular choice in practice.

6. Future Work

It is of interest to investigate what additional information about the graph is encoded by the eigenspace of its Markov matrix. We have seen that the dominant eigenvector represents an expected visitation to each node via a random walk, but are unsure what meaning the other eigenvectors has.

To explore this question, one could compute the eigenspaces for a large number of valid Markov matrices of the same size, and examine them for any apparent structure. Another possibility would be to look for symmetries in the transition matrix $\mathbf{P}$, for example, conditions on a matrix A such that

$$A\mathbf{P} = A \ , \tag{16}$$

which would give insight into the family of graphs which are described by a given eigenspace.

References

[1] Ying Bao, Guang Feng, Tie-Yan Liu, Zhi-Ming Ma, and Ying Wang. Ranking websites: a probabilistic view. *Internet Mathematics*, 3(3):295–320, 2006.

[2] Amy N Langville and Carl D Meyer. Deeper inside pagerank. *Internet Mathematics*, 1(3):335–380, 2004.

[3] Amy N Langville and Carl D Meyer. *Google's PageRank and beyond: The science of search engine rankings*. Princeton university press, 2006.

[4] Taher H Haveliwala and Sepandar D Kamvar. The second eigenvalue of the google matrix. Technical report, Citeseer, 2003.