EMBEDDING WITH PAGERANK

by

Disha Shur

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science

School of Electrical and Computer Engineering

West Lafayette, Indiana

May 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. David F. Gleich, Co-Chair

Department of Computer Science

Dr. Mireille Boutin, Co-Chair

School of Electrical and Computer Engineering

Dr. David I. Inouye, Member

School of Electrical and Computer Engineering

Approved by:

Dr. Dimitrios Peroulis

ACKNOWLEDGMENTS

I am extremely grateful to Prof. David Gleich for agreeing to be my advisor. Especially for guiding me through courses, throughout the research in this project, for funding my education at Purdue and most of all, for being patient when I made embarrassingly silly mistakes.

I am earnestly thankful to Prof. Mireille Boutin for giving necessary insights into the problem. This document would not have looked the same without her inputs and ideas.

I would also like to express my gratitude towards Prof. David Inouye who agreed to be on my thesis committee and drove interesting discussions on my research matter.

I have to thank my family for supporting me when I was not doing so well during the last 1.5 years through the phone.

Lastly, I would like to thank the group members of **Networked Matrix** for responding to my questions pertaining to this work, or otherwise. I would also like to thank Pat, Charlie, Akshita, Yash, Dawn and Becky for making sure I had a comfortable stay while at Purdue.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Adjacency matrix, $\mathbf{G}$

Diagonal degree matrix, $\mathbf{D}$

Degree vector, $\mathbf{d}$

Laplacian, $\mathbf{L} = \mathbf{D} - \mathbf{A}$

Symmetric laplacian, $\tilde{\mathbf{L}} = \mathbf{I} - \mathbf{D^{-1/2}GD^{-1/2}} = \mathbf{Q\Lambda Q^T}$

Transition probability matrix, $\mathbf{P} = \mathbf{GD^{-1}}$

Personalization vector, $\mathbf{v_k}$ seeded on the kth node, Personalization matrix, $\mathbf{V}$

PageRank vectors, $\mathbf{x}$: $\mathbf{x} = (1-\alpha)(\mathbf{I} - \alpha\mathbf{P})^{-1}\mathbf{v_k}$

PageRank matrix, $\mathbf{X}$: $\mathbf{X} = (1-\alpha)(\mathbf{I} - \alpha\mathbf{P})^{-1}\mathbf{V}$

eigenvalues of the laplacian, λ,

eigenvalue matrix of the Laplacian $\mathbf{\Lambda}$

eigenvalues of the transition matrix, $\mathbf{I} - \mathbf{\Lambda} = \boldsymbol{\mathcal{E}}$

ABSTRACT

Personalized PageRank with high teleportation probability enables exploring the environment of a seed. With this insight, one can use an orthogonal factorization of a set of personalized PageRank vectors, like SVD, to derive a 2-dimensional representation of the network. This can be done for the whole network or a smaller piece. The power of this method lies in the fact that only a few columns, compared to the size of the networks, can be used to generate a local representation of the part of the network we are interested in. This technique has the potential to be seamlessly used for higher order structures, such as hypergraphs which have found a great deal of use for real-world data. This work investigates the characteristics of personalized PageRank and how it compares to the transition probabilities on the graph in terms of their ability to develop low dimensional representations. A key focus of the thesis are the similarities between the embeddings generated due to PageRank and those generated by spectral methods.

1. INTRODUCTION

PageRank [1], in its most basic form, is a mechanism to rank nodes, surrounding a seed node (or a set of seed nodes), based on their connectivity with the latter. Belying its success for ranking websites is the probability transition matrix of the graph developed from each element on the internet. The formulation of PageRank relies on a probabilistic extension of this transition matrix that maintains a random walk with a probability given by the teleportation factor (also referred to as the damping factor), and resets itself onto some predetermined nodes otherwise. Although originally designed for the internet, we now know that equivalent ranking schemes or at least those based on similar formulations of the probability transition matrix apply to ranking mechanisms in different settings [2] where the ranked elements can be arranged in a graph. Besides ranking, PageRank has been shown to be efficient [3] for recovering clusters as well compared to other ranking mechanisms. Apart from ranking and graph partitioning, PageRank has also been used to develop graph representations [4] and node embeddings ([5]).

PageRank can either be characterized by studying its probabilistic nature, in terms of the teleportation factor, or the geometric series in terms of powers of the probability transition matrix. Both these perspectives have been widely discussed and documented rigorously [6]. These perspectives underscore the various useful properties that PageRank might have depending on the teleportation factor and the spectral characteristics of the probability transition matrix (or the symmetric Laplacian). The geometric power series generated by PageRank , for example, has been compared in terms of its efficiency for clustering, with exponential power series of the heat kernel [7] and the wave kernel [8]. It has also been used to develop techniques for graph representations [4] in terms of spectral characteristics of the resultant series (as opposed to the node embeddings developed using the probabilistic perspective). A consequent line of work concerned with PageRank is its ability to produce embeddings for the graph.

The different perceptions of PageRank - power series of transition probabilities and diffusion - highlight its nature of incorporating the environment of the seed nodes which hints at its efficiency as an embedding technique. From the formulation of PageRank, it is to be expected that the teleportation probability, that is the probability with which it either explores the surrounding, or stays focused on the seed nodes, holds most of the strength in PageRank's ability to create representations.

This thesis offers insights regarding the role of α, the teleportation probability, in embeddings generated by PageRank as compared to those generated by the transition matrix of the graph. The personalization vectors in this work is comprised of just one node. Thus it can be interpreted as random sampling of nodes from the given graph, referred to as the seed nodes, and each PageRank procedure is seeded on one such node.

The columns obtained from the PageRank operation on each of the seed nodes are then augmented into one matrix that we denote as $\mathbf{X}$. For example, for a 10 node graph with 4 nearest neighbours and a seed set consisting of 4 randomly selected nodes, the matrix would look as follows.

$$\mathbf{X} = \begin{bmatrix} 0.065 & 0.058 & 0.065 & 0.065 \\ 0.061 & 0.091 & 0.061 & 0.061 \\ 0.056 & 0.075 & 0.056 & 0.056 \\ 0.061 & 0.091 & 0.061 & 0.061 \\ 0.18 & 0.086 & 0.18 & 0.11 \\ 0.11 & 0.12 & 0.11 & 0.11 \\ 0.11 & 0.086 & 0.11 & 0.18 \\ 0.14 & 0.22 & 0.14 & 0.14 \\ 0.14 & 0.11 & 0.14 & 0.14 \\ 0.06 & 0.05 & 0.06 & 0.06 \end{bmatrix} \tag{1.1}$$

The observant reader will notice that the 1st and 3rd columns are equal. This happened because the corresponding seed nodes sampled were same.

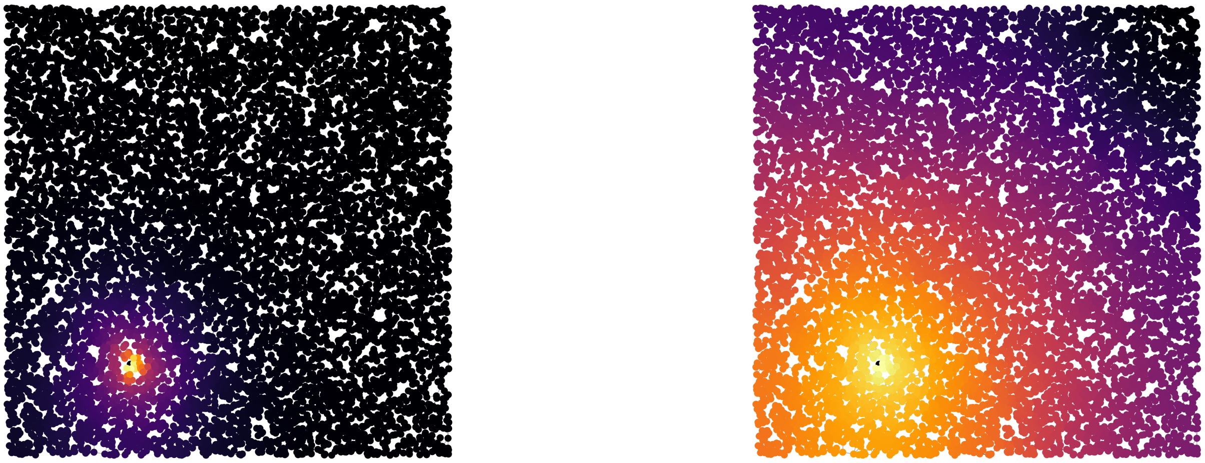
The primary focus of this thesis revolves around the embedding generated by the singular vectors of $\mathbf{X}$. The tipping point of inspiration for this work was an observation that the embeddings generated using a high value of α resembled the spectral embeddings that result from the eigenvectors of the symmetric Laplacian. Moreover, at lower values of α, the element-wise logarithm operation on the PageRank created a smoothed version of the spectral picture. We discuss more about this under the Motivation section. This led us to investigate the ”spectrum” of effect that α has on the PageRank and discover relations with other existing methods of embedding development.

PageRank has also been used on higher order networks such as hypergraphs for clustering in semi-supervised learning. [9] defines PageRank for hypergraphs and uses it to for a clustering algorithm. A more localized method for clustering via PageRank is offered by [10]. Developing hypergraph representations for semi-supervised learning have used non-linear methods with hyperedge expansion techniques([11],[12]). Non-linearity based learning models are known to be inexplicable and hyperedge expansion techniques have shown to cause loss of structural information in the resulting embeddings. Interestingly, the method discussed in this work offers a theoretical alternative that does not suffer with these issues. Efficient algorithms for calculating PageRank on hypergraphs already exist, but have not been used for hypergraph representation. If the success of PageRank embeddings on graph is to serve as a hint, for analogous definitions, they should also be effective on hypergraphs. Since most real-world data is better modelled by a hypergraph, this technique be more efficient and explicable at representing the higher order connections.

1.1 Motivation

Until 2016, when PageRank was a public-facing metric, it used the ranks in a logarithmic scale. Inspired by this, we ran some experiments of our own for a synthetic - 10000 node random distributed graph. As a first observation we visualized the PageRank values for PageRank seeded on a random node, with and without log. Figures 1.1a and 1.1b show the diffusion of PageRank values for each case. Figure in 1.2a provides another perspective that we will discuss in later sections. It shows the diffusion of values in powers of the transition matrix of the graph. The black dot shows the seed. The figure in 1.1b hints that the logarithm of PageRank might be a better indicator, than PageRank itself, of the environment around the seed node in the graphs.

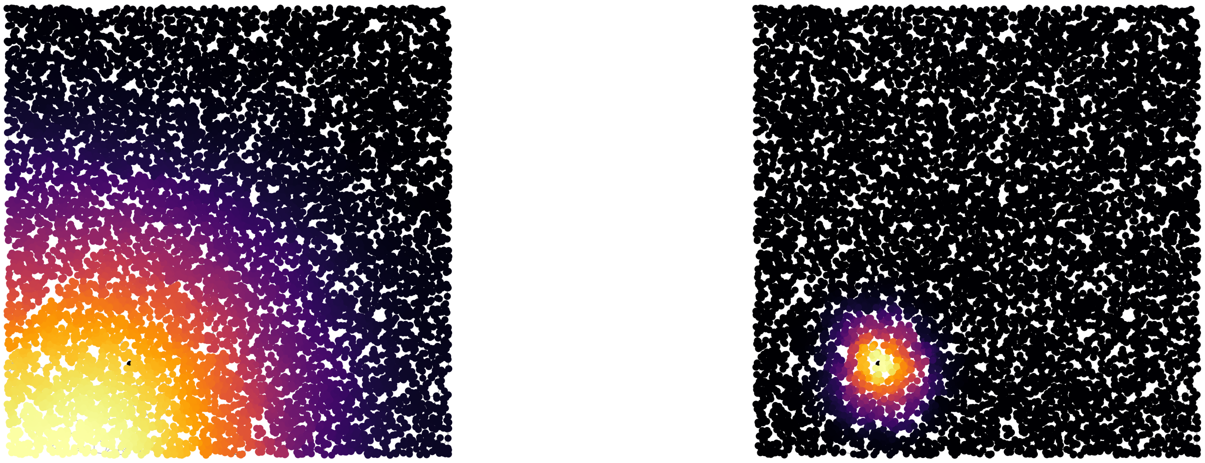
The next perspective is offered by an insight shared in [13]. Given a set of seed nodes, the PageRank on the entire set can be approximated by taking expectation of the same operation on each individual nodes in the set, which was further found to be equivalent to the first left



(a) PageRank values for a 10000 node graph with 6 nearest neighbour with $\alpha = 0.999$

(b) Log of PageRank values for a 10000 node graph with 6 nearest neighbour with $\alpha = 0.999$

Figure 1.1. Diffusion of PageRank and log of PageRank



(a) Normalized $(D^{-1/2}GD^{-1/2})^pD^{-1/2}GD^{-1/2}z$ where z is an indicator vector for the seed used in PageRank above and $p = 2000$

(b) Normalized $(D^{-1/2}GD^{-1/2})^pD^{-1/2}GD^{-1/2}z$ where z is an indicator vector for the seed used in PageRank above and $p = 100$

Figure 1.2. Diffusion of transition probabilities

singular vector of the augmented PageRank matrix. This translates to sampling a random node at a time to carry out personalized PageRank and augmenting these PageRank columns into a matrix, whose expectation over the columns then gives the personalized PageRank on the entire set. For example, for a 10 node graph with 4 nearest neighbors, and a seed set consisting of 4 randomly selected nodes, the expected value of PageRank with $\alpha = 0.9$ on each individual seed, its left most singular vector and PageRank on the complete seed set has the following values respectively. The numbers are rounded to 4 significant figures. Every vector is normalized to 1.

$$\begin{aligned} U &= \Big[0.06369, 0.06842, 0.06099, 0.06842, 0.1411, 0.1107, 0.1214, 0.1644, 0.1369, 0.06369\Big] \\ \mathbb{E}[X] &= \Big[0.06367, 0.06852, 0.06105, 0.06852, 0.1407, 0.1107, 0.1214, 0.1647, 0.1368, 0.06367\Big] \\ Y &= \Big[0.06305, 0.07103, 0.06270, 0.07102, 0.1261, 0.1116, 0.1261, 0.1717, 0.1336, 0.06305\Big] \end{aligned}$$

Further, an orthogonal component to the PageRank can be obtained using the singular value decomposition of the matrix obtained from augmenting the individual PageRank columns. This suggests that the vectors thus obtained can be used as 2-dimensional embeddings. Figure 1.3 shows the embeddings generated by the above procedure. In a nutshell, (if this procedure works, we will see the conditions later in the thesis), this procedure offers the advantage of using only a small fraction of the nodes to develop the embeddings and create a low dimensional representation of the entire structure!

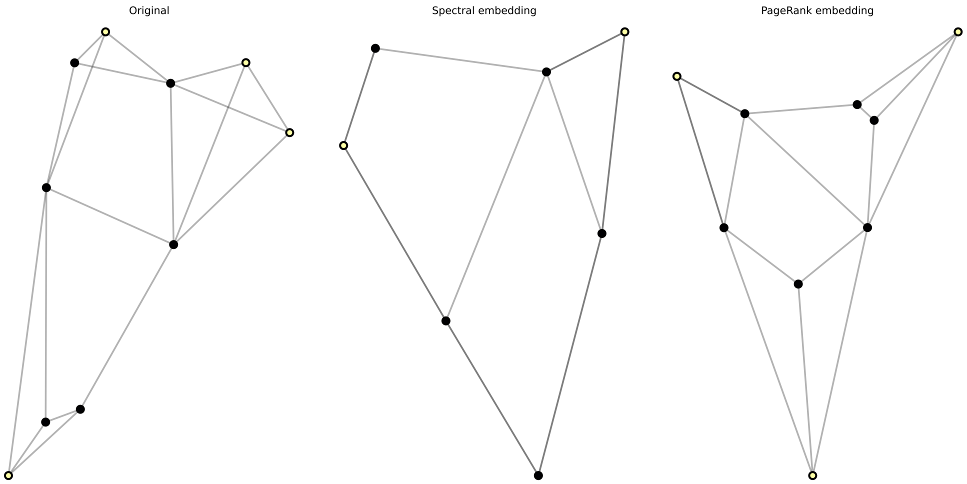
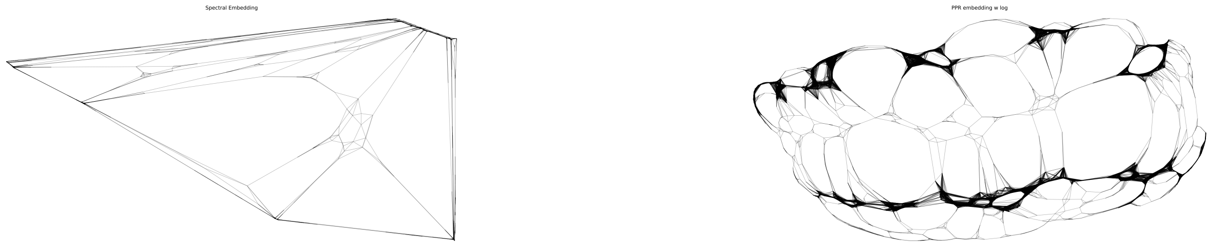


Figure 1.3. Original, spectral and PageRank representation of 10 node nearest neighbour graphs with 4 neighbors with $\alpha = 0.9$ where the light coloured nodes are the sampled seed nodes.

However, for larger graphs, this did not turn out to be completely true as the norm of difference between the singular vector and the PageRank decreased with increasing teleportation probabilities. For a 1000 node graph with 6 nearest neighbours, the difference between U and Y for $\alpha = 0.9$ in norm-1 was 168.93% which came down to 39.3% with the log operation before the singular vectors and for $\alpha = 0.99999$, it became 0.018%.

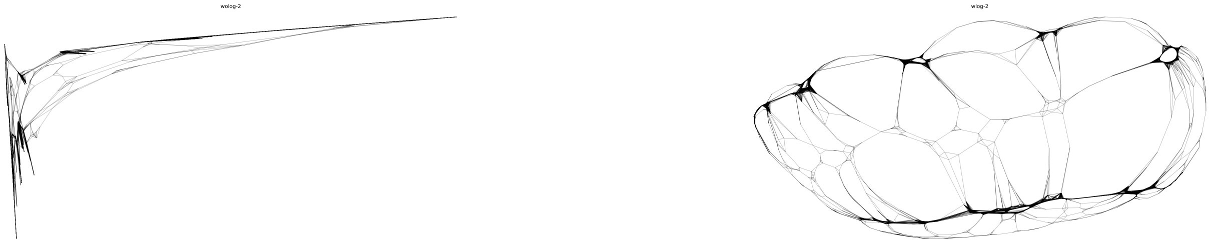
As a consequence of the above two observations, we look at the effect of increasing teleportation probability and the logarithm operation in the embeddings obtained from PageRank. The figures in (1.4a, 1.4b, 1.5) show the embeddings with and without log on PageRank for the original mesh graph with 5500 nodes at $\alpha = 0.9$. The next figures (1.6a,1.6b) show embeddings for the same graph generated with and without log of PageRank at $\alpha = 0.99$ while figures (1.7a,1.7b) show the same at $\alpha = 0.9999$. The next figures serve to exactly compare the vectors instead of the picture generated by the embeddings. Figures 1.8a-1.9 shows the effect on the scale between the embeddings as α grows without the element-wise log operation while 1.10a - 1.11 show the same with the log operation. Figures 1.12a - 1.13a show a case of the graph where the procedure was observed to fail. This graph was generated by a planted-partition model with in-block edge probability, $p = 0.001$ and out-block edge probability, $q = 0.005$ with 60 blocks of 50 nodes each. Yet, for a planted-partition model with 3 blocks each with 50 nodes and in-block edge probability of $p = 0.25$ and out-block



(a) Spectral embedding for the original graph

(b) PageRank embedding with log on PageRank at $\alpha = 0.9$

Figure 1.4. PageRank and spectral embeddings at lower α



(a) PageRank embedding without log at $\alpha = 0.99$

(b) PageRank embedding with log at $\alpha = 0.99$

Figure 1.6. PageRank embeddings at lower α

edge probability, $q = 0.001$, the method works again at a quite small value of $\alpha = 0.99$ as shown in figure 1.13b.

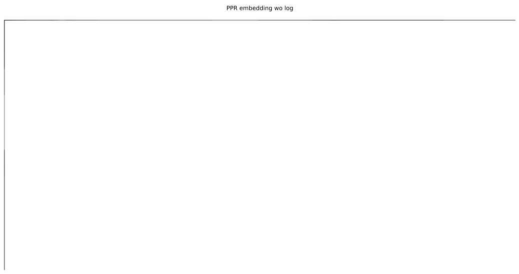
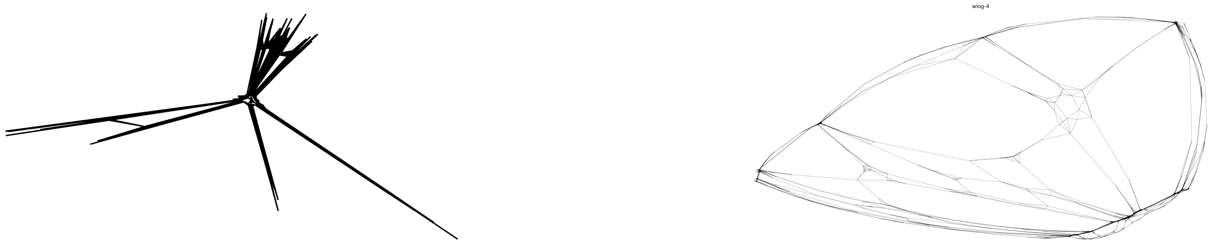


Figure 1.5. PageRank embedding without log on PageRank at $\alpha = 0.9$

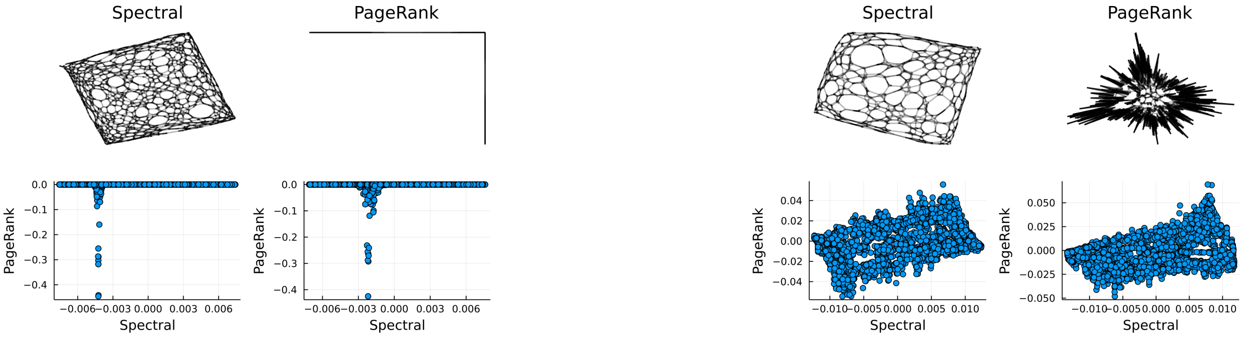
The next figure shows the embeddings generated by similar procedure as above but by varying the teleportation probability as $\alpha = 0.99$ and $\alpha = 0.9999$.



(a) PageRank embedding without log at $\alpha = 0.9999$

(b) PageRank embedding with log at $\alpha = 0.9999$

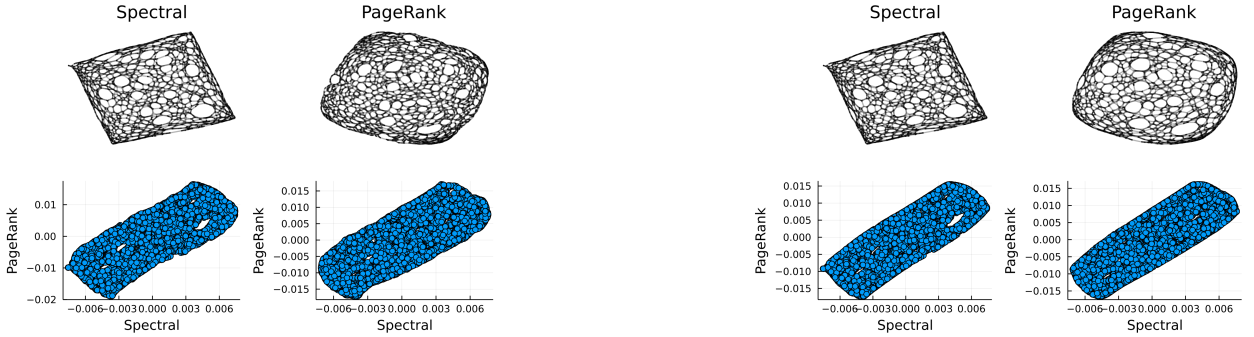
Figure 1.7. PageRank embeddings at higher α



(a) PageRank embeddings without log with $\alpha = 0.9$

(b) PageRank embeddings without log with $\alpha = 0.999$

Figure 1.8. Comparison of embedding vectors without log



(a) PageRank embeddings with log with $\alpha = 0.9$

(b) PageRank embeddings with log with $\alpha = 0.999$

Figure 1.10. Comparison of embedding vectors with log

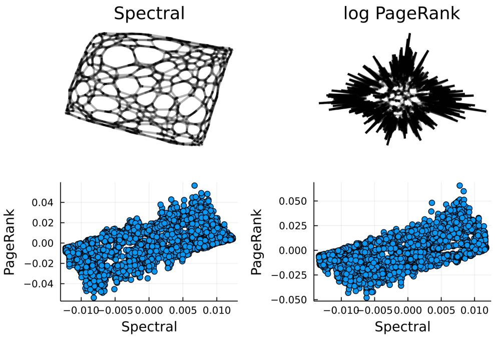


Figure 1.9. PageRank embeddings without log with $\alpha = 0.99999$

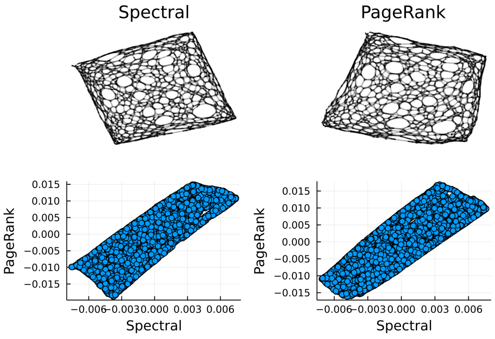
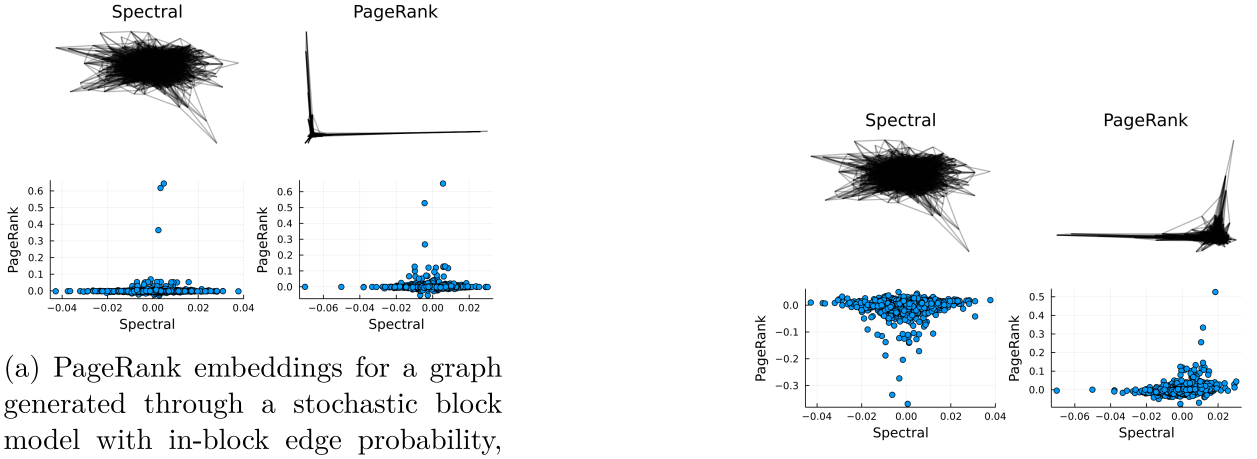


Figure 1.11. PageRank embeddings with log with $\alpha = 0.99999$

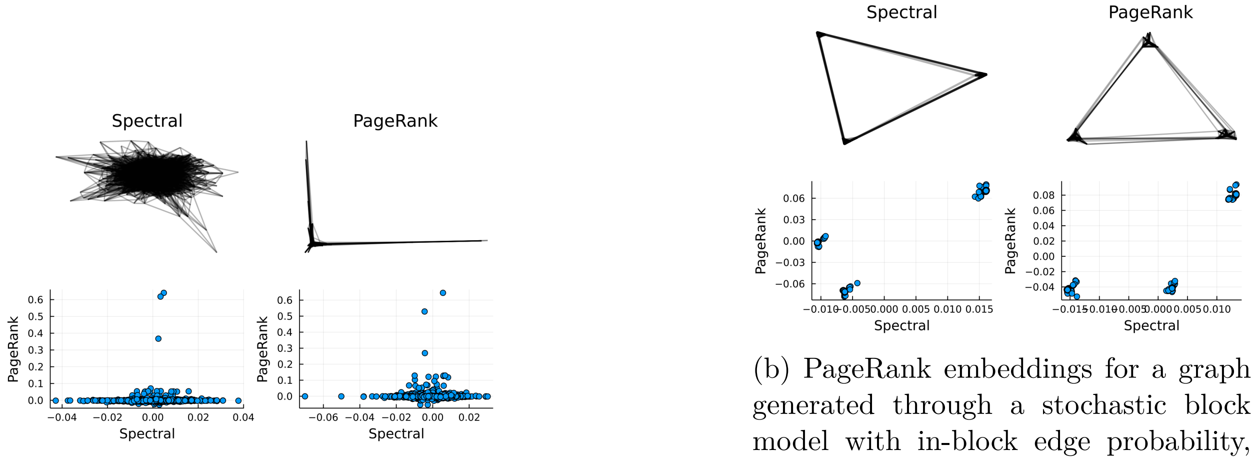
The logarithm operation on PageRank for certain graphs, apparently, makes its singular vectors closer to the eigenvectors of the laplacian and this similarity increases with the teleportation probability, when it no longer requires the element-wise logarithm operation. Further, above a certain minimum number of samples, the expected value of the operation



(a) PageRank embeddings for a graph generated through a stochastic block model with in-block edge probability, $p = 0.001$, and out-block edge probability, $q = 0.005$ with $\alpha = 0.99$

(b) PageRank embeddings for the above graph with log with $\alpha = 0.99$

Figure 1.12. Comparison of embeddings for planted partition model on low α



(a) PageRank embeddings for the same graph with with $\alpha = 0.99999$

(b) PageRank embeddings for a graph generated through a stochastic block model with in-block edge probability, $p = 0.25$, and out-block edge probability, $q = 0.001$ with $\alpha = 0.99$

Figure 1.13. Comparison of embeddings for planted partition model on high α

on a set of nodes approaches that on the entire set. The approximation seen here between SVD of the PageRank and the eigenvectors breaks for smaller number of samples, and has negligible improvement for excessive number of samples. This ambiguity with respect to the teleportation probability and graphs begs more investigation at the trajectory of PageRank as α tends to 1. Moreover, it is not apparent as to how would the logarithm operation, for smaller values of α, drive the PageRank to generate embeddings similar to spectral embeddings. These observations are the basis of the problem statement that this thesis tries to answer.

1.2 Problem Statement

Given an undirected and connected graph with adjacency matrix, $\mathbf{G}$, the singular vectors, $\mathbf{U}$, generated by the PageRank procedure personalized on individual seed nodes, resemble the eigenvectors, $\mathbf{Z}$, of its symmetric Laplacian, $\tilde{\mathbf{L}}$, under conditions controlled by the teleportation probability, α and $\mathbf{G}$. We quantify the observations under which this resemblance is found to be true and attempt to justify the following :-

1. For smaller graphs ($n \sim 10$), $\mathbf{U}$ is similar to $\mathbf{Z}$ for comparatively smaller values of the teleportation probability, $0.9 \leq \alpha \leq 0.999$.

2. For larger graphs ($n \sim 10^3$), $\mathbf{U}$ generated by the element-wise logarithm of the PageRank matrix resemble $\mathbf{Z}$ for $\alpha = 0.99$, but the logarithm operation is not needed for larger $\alpha = 0.9999$. That is, for the graph in this case, with $\alpha = 0.9999$, the PageRank embeddings are similar to spectral embeddings.

3. This resemblance did not stand true for all graphs, for example, the planted-partition stochastic block model with almost equal inter-block and intra-block edge probability.

Thus, this thesis formalizes why the teleportation probability seemingly needs to be increased with the number of nodes in the graph and what role does the element-wise logarithm play for larger graphs working with small α.

1.3 Results

In an attempt to answer the above questions, this thesis puts forward the following results.

1. The above resemblance occurs only for graphs where the second order difference at the Fiedler value is inversely proportional to the size of the graph, that is $\lambda_3 - 2\lambda_2 + \lambda_1 = O(1/n)$. This is always true for planar graphs because for planar graphs $\lambda_2 = O(1/n)$.

2. Embeddings generated by the PageRank, that is $\mathbf{U}$, resemble the spectral embeddings, $\mathbf{Z}$, when the teleportation probability, $\alpha \to 1 - \lambda_2$.

3. The element-wise logarithm operation is necessary because, besides maintaining the distribution of the PageRank values over the graph, for larger graphs, when $\alpha << 1 - \lambda_2$, the order of PageRank values are large negative numbers which become significant after using the logarithm.

4. We establish a closed form expression for Personalized PageRank for chain graph as a function of the teleportation probability, α, and the distance from the seed node. This justifies the low order of PageRank values for large graphs and low α.

5. We extend the relation of element-wise logarithm operation and the stationary distribution of a graph to the Personalized PageRank operation.

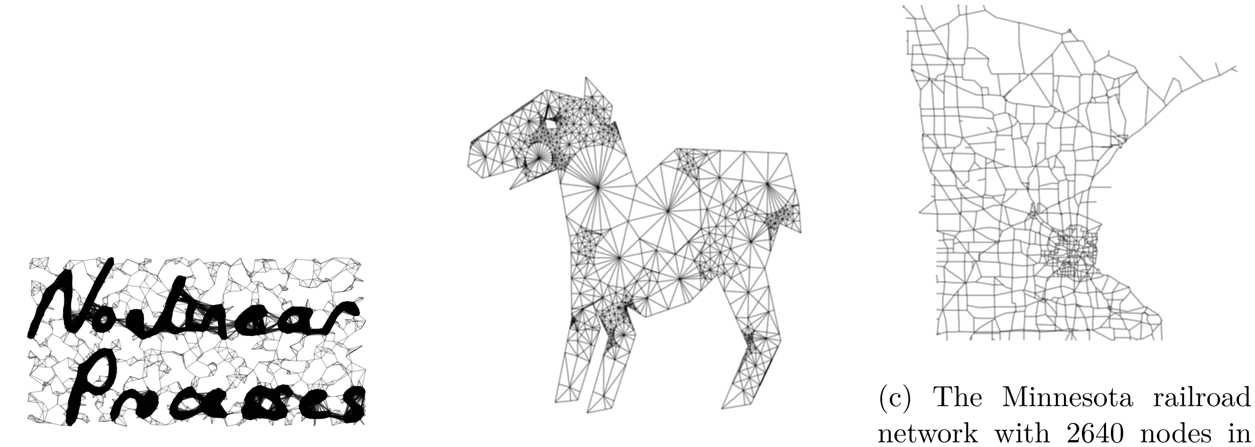
6. Most importantly, we establish that given the configuration of above-mentioned eigenvalues, and $\alpha \to 1$, log of PageRank behaves like matrix powers of $\mathbf{P}$, that is random walk on the graph and hence is equally reliable but more precise at creating low-dimensional representations of a graph.

1.4 Organization

We closely follow the order of results. Chapter 3 introduces the fundamentals of PageRank used to build the results in this thesis. Chapter 4 looks into the spectral properties of the graphs which do and do not agree with the resemblance. Chapter 5 looks at the Chain graph to justify the necessity of logarithm operation. Chapter 7 focuses on other techniques based on the transition probability matrix, $\mathbf{P}$, and the element-wise logarithm operation and compares it to PageRank. Chapter 6 looks at the limiting value of PageRank with respect to the teleportation probability. Finally, chapter 8 concludes with our experiments and observations with PageRank embeddings for Hypergraphs and other open directions.

2. NAMED GRAPHS

This chapter shows the named graphs and their details that have been frequently referred to in this document.



(a) Original mesh graph with 5500 nodes

(b) The Tapir graph with 1024 nodes

(c) The Minnesota railroad network with 2640 nodes in the largest connected component

Figure 2.1. Named Graphs

3. BASICS AND RELATED WORKS

This chapter outlines the fundamental definitions that are central to the concepts that this thesis works with.

3.1 Embeddings

Low dimensional embeddings are the basis of learning representations of any data. The driving cause of this work was to analyse a technique that produces low dimensional representations of a network that is also a reliable representation of its structure. In the context of this work, we draw our embeddings based on the quadratic energy minimizer defined in [14]. Throughout the document, we will maintain the following definition of embedding.

Definition 3.1.1 (Embedding)**.** *A $d-$dimensional embedding is an assignment of d coordinates given by d orthonormal vectors, $\langle \mathbf{u_1}, \mathbf{u_2}, \cdots, \mathbf{u_d} \rangle$ to each of the nodes such that the d-dimensional representation respects graph automorophism.*

In the course of this work, we ensure the above by assigning embeddings such that ith coordinate of the kth node is given by the kth entry in the ith vector, $\mathbf{u_i[k]}$. For example, the 2 dimensional spectral embeddings, according to the quadratic energy is given as $i : \{\mathbf{Q[i, 2]}, \mathbf{Q[i, 3]}\}$, where $\mathbf{Q}$ are the eigenvectors of the Laplacian, $\mathbf{\tilde{L}}$, of the graph, $\mathbf{G}$, defined as $\mathbf{\tilde{L}} = \mathbf{I} - \mathbf{D^{-1/2}GD^{-1/2}}$. Although for better visualization, we use the degree scaled version of these vectors, that is $\mathbf{D^{-1/2}Q}$, which are orthonormal with respect to the inner-product: $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u^T D v}$.

3.2 PageRank

The math involved in PageRank has been widely studied and documented since its inception ([1], [2],[15], [6]) in 1998 which would make discussing it here futile. Hence we would only be looking at the elements that are necessary to understand the arguments in the following chapter that attempt to explain the observations. The PageRank vector, starting with the transition probability matrix of a graph, P, can either be seen as an eigenvector to the following system,

$$(\alpha\mathbf{P} + (1-\alpha)\mathbf{v}\mathbf{e}^{\mathbf{T}})\mathbf{x} = \mathbf{x} \tag{3.1}$$

or as a solution to the linear system

$$(\mathbf{I} - \alpha\mathbf{P})\mathbf{x} = (1-\alpha)\mathbf{v} \tag{3.2}$$

where (3.2) follows from (3.1) by enforcing $\mathbf{e}^{\mathbf{T}}\mathbf{x} = 1$. Consequently, the PageRank vector, $\mathbf{x}$, and the transition probability matrix, $\mathbf{P}$, has the following properties that were of use in the work.

1. $x_i \geq 0$
2. $\mathbf{P} = \mathbf{G}\mathbf{D}^{\mathbf{-1}}$ is a column-stochastic matrix
3. Accordingly, $\mathbf{P}$ has $\mathbf{e}$ as its left eigenvector and the stationary distribution, $\boldsymbol{\pi}$, as its right eigenvector. We denote by $\mathbf{\Pi}$, the matrix where each column is given by $\boldsymbol{\pi}$.
4. $\mathbf{P}$ has eigenvalues between -1 and 1

3.3 PageRank calculation

The original work [1] started by computing the PageRank using the power method. The main idea behind power method looks as follows.

$$\mathbf{x}^{(k+1)} = \alpha\mathbf{P}\mathbf{x}^{(k)} + (1-\alpha)\mathbf{v} \tag{3.3}$$

An expansive range of algorithms that are variants of the power method have been discussed in [15]. Our initial experiments started with the same. Not much later, as the motivation section states, figure 1.5 hinted at effect increasing the value of α might have. But the more the value of α was increased, the more time the above procedure took to run. This makes sense if one looks at the convergence criteria of the power method. The vectors produced by the iterations in power method converge according to the second eigenvalue of the PageRank matrix in (3.1), λ_M, which is directly dependent on α.

$$\lambda_M = \alpha\epsilon_2 \tag{3.4}$$

For the graphs that we experiment with (more details in section 4.2), $\epsilon_1 = 1$ and $\epsilon_2 \sim 1$. This implies that the rate of convergence is directly proportional to α. For our experiments as $\alpha \to 1$, the number of iterations, hence the runtime, of the power iteration increases. Therefore, we resort to an alternative method where convergence does not rely on α. An advantage of using iterative methods such as the power method is these methods are matrix free, so the algorithm does not deal with matrix operations. Since our experiments are based on much smaller matrices ($\sim$ 10000) compared to Google's (where the order of the matrix was 8.1 billion), we can look beyond matrix-free methods.

From (3.2),

$$\mathbf{x} = (1-\alpha)(\mathbf{I} - \alpha\mathbf{P})^{-1}\mathbf{v} \tag{3.5}$$

We know that the matrix inverse will exist because $\mathbf{I} - \alpha\mathbf{P}$ is non-singular. The properties of this matrix have been well discussed in Chapter 7 of [6], so we will not repeat it here. We will only note the fact that for small problems ($\sim$ 10000) as in our case, the above matrix-inverse can be computed using much less time and computation as compared to the power iteration method. Further, using LU decomposition of the concerned matrix, the matrix inverse need not be explicitly calculated, thus making it faster and independent of α. For example, for a 3000 node graph with 6 nearest neighbours, calculating one PageRank vector with $\alpha = 0.999$ with the power iteration method takes 4.5 seconds which scales to 45.17 second for $\alpha = 0.9999$, while that with the matrix-inversion method takes 0.12 seconds and stays the same even for larger α. For 200 instances of PageRank, the method based on power iterations would scale by a factor of 2000 which is impractical. Therefore, throughout the work reported in this thesis, matrix-inversion was used to calculate PageRank.

Another factor that arise from this discussion is the sensitivity of the PageRank vector according to α and the Fiedler value of the Laplacian. For embedding purposes, we need our PageRank vector to be sensitive to the structural changes, thus giving more weight to $\mathbf{P}$ in the PageRank matrix (and equivalently increasing the condition number of the linear system of the PageRank vector). Referring to the theory outlined in [6], we know that closer the Fiedler value is to 1, the more sensitive the PageRank vector becomes as α approaches 1, which implies, the embeddings generated by the PageRank vector, in that given range of α, would be more faithful to the structure of the network.

3.4 PageRank as power series

Physically, the personalized PageRank is a diffusion process that starts at the seed node. Hence, the similarities between PageRank and other diffusion processes such as the heat flow equation ([7],[4]) and the wave flow equation ([4],[8]) shed light on the role of the teleportation probability and significance of the PageRank values on log scale. The series representation of PageRank ([7])

$$x_{\alpha,\mathbf{v}} = (1-\alpha)\mathbf{v} + \alpha\mathbf{x}\mathbf{P} \rightarrow x_{\alpha,\mathbf{v}} = (1-\alpha)\sum_{k=0}^{\infty}\alpha^k\mathbf{P}^k\mathbf{v} \tag{3.6}$$

representation of PageRank facilitates the development of heat kernel PageRank in [7],

$$\frac{\partial}{\partial t}\rho_{t,\mathbf{v}} = -\rho_{t,\mathbf{v}}((I-P)) \rightarrow \rho_{t,\mathbf{v}} = e^{-t}\sum_{k=0}^{\infty}\frac{t^k}{k!}\mathbf{P}^k\mathbf{v} \tag{3.7}$$

which offers a localized algorithm for graph partitioning. The heat kernel representation is also used by [4] to compare graphs. A similar representation, but with second degree dependence, is derived with the wave equation in [16],

$$\frac{\partial^2 u}{\partial t^2} = c^2\delta u \rightarrow \mathbf{u}_i(t) = 2\mathbf{u}_i(t-1) + \mathbf{u}_i(t-2) - c^2\sum_{j\in neighbours(i)} L_{ij}u_j(t-1) \tag{3.8}$$

which helps by keeping intact the information in higher eigenvaectors, which is otherwise diminished in both PageRank (with higher α) and the heat equation (with higher t). Using this higher degree information of the eigenvectors and the above formulation, [16] gives a localized algorithm for eigenvector computation and hence, clustering. Along the same representation, [8] used spectral graph wavelet, for a node a

$$\mathbf{\Psi_a} = \mathbf{U} Diag(g_s(\lambda_1), \cdots, g_s(\lambda_N))\mathbf{U}^T \delta_a \tag{3.9}$$

whose diffusion based probability measure enables learning structural embeddings for nodes. This method again shares similarities with our technique in the sense that $\mathbf{\Psi}$ is $N \times N$ matrix with the ath column referring to wavelet originating at the ath node. Comparing this with our observation in terms of the personalized PageRank (PPR) vector, the definitions would change as,

$$\mathbf{X_v} = (\mathbf{\tilde{L}})^{-\mathbf{1}}\delta_\mathbf{v} = (\mathbf{D}^{-\mathbf{1/2}}\mathbf{U})\mathbf{\Lambda}^{-\mathbf{1}}(\mathbf{U^T D^{1/2}})\delta_\mathbf{v}$$

where δ_v is an indicator for the vertex v. Finally $X \in \mathcal{R}^{N \times N'}$ where N' is the number of seed nodes for which the PageRank vector is calculated and is randomly selected (to be 200 in our observations). The characteristic function here is meant to capture all moments of $\mathbf{\Psi}$. But instead of the calculating the characteristic function in terms of exponential, we take element-wise log of every element and then take the left singular values (instead of sampling) to get the embeddings.

All of these studies drawing comparisons between the geometric sum expression of PageRank and an exponential sum expression using the transition or the Laplacian matrix, hints towards the efficiency of logarithm of PageRank values as compared to the values without logarithm. Although, we tried to use other similar functions such as negative exponential of PageRank and the truncated Maclaurin series, but those could not maintain the distribution of the PageRank values of the graph.

3.5 Inspecting PageRank as optimization problem

The PageRank vector can also be calculated as the solution to the 2-norm optimization problem as described in [17] and [10]. Combined with the ideas in this work, it offers a way to develop embeddings for higher order structures such as hypergraphs. Although while working on the technique, we discovered that it needs the PageRank vector to not be sparse, as in [9], and hence the overall algorithm loses the advantage of being localized.

3.6 Other embedding techniques using the transition matrix, P

This section looks into some of the techniques that, like PageRank, uses the transition probability matrix to develop embeddings. While scrounging the literature for element-wise logarithm operation on the transition probability matrix, we discovered about the technique known as SkipGram ([18]) that can be expressed as an implicit matrix factorization([19]). Further, [20] shows the popular network embedding techniques ([21],[22],[23],[24]) to be matrix factorization of some function of the adjacency matrix. SkipGram, widely used for natural language understanding, uses the contextual similarity between words to develop a graph out of the given text. The adjacency matrix of the graph thus formed, [19] shows, can be factorized with element-wise logarithm to develop embeddings. Along with the popular embedding techniques, [20] explains the application of this procedure, using random walks on generic networks, that are not necessarily text based. The objective of SGNS is to maximize the similarity between the embeddings of each word-context pair and decrease that between the word and its negative samples. Mathematically, this is achieved by maximizing the log of sigmoid ($(1 + e^{-x})^{-1}$) of the dot product of the node (or word) embeddings. [19] showed that minimizing that objective function is equivalent to factorizing a matrix and using an information-theoretic lens, it can be related to the Pointwise Mutual Information (PMI) between any two words. This perspective drives the Chapter 6 of this thesis.

4. PAGERANK VS SPECTRAL EMBEDDING

The inception of this problem was the observation in [13] that the average of columns of the PageRank matrix, personalized on each seed separately, can be approximated by the first left singular vector. The secondary motivation behind this thesis is a tangential observation that at sufficiently high teleportation probability, the left singular vectors of the element-wise logarithm of PageRank, and sometimes only the PageRank, are similar to the eigenvectors of the symmetric Laplacian. This has been explained in detail in section 1.1. This section probes into this observation in terms of the spectral properties of the symmetric Laplacian, $\tilde{\mathbf{L}}$, and the transition probability matrix of the graph, $\mathbf{P}$.

As a primary step, we verified that the concerned equivalence did not stand out to be true for every graph. We saw a pattern among the eigenvalues. For example, for the graph in the motivation section, for which the equivalence worked, its symmetric Laplacian had the following lowest 5 eigenvalues (rounded to 4 significant figures).

$$\Lambda = \Big[-4.441e^{-16}, 0.005862, 0.02338, 0.05235, 0.09242\Big] \tag{4.1}$$

and for the planted-partition model for which it did not work, had the following eigenvalues.

$$\Lambda = \Big[0.0, 0.3969, 0.4009, 0.4032, 0.4041\Big] \tag{4.2}$$

Although the pictures provide a qualitative judgement, to quantify the similarity of the embeddings, we borrow ideas from the Rayleigh quotient, and define a similarity score as follows,

$$\text{sim\_error} = \frac{s-p}{s} \tag{4.3}$$

where

$$s = \frac{\mathbf{z_2'}\tilde{\mathbf{L}}\mathbf{z_2}}{\mathbf{z_2'}\mathbf{z_2}} \tag{4.4}$$

and

$$p = \frac{\mathbf{u_2'}\tilde{\mathbf{L}}\mathbf{u_2}}{\mathbf{u_2'}\mathbf{u_2}} \tag{4.5}$$

This error for different graphs have been tabulated below in Table 4.1. For the chain graph, the error started to go as low as 5.98% for $\alpha = 1 - 10^{-6}$ without the log operation because of the order of its Fiedler value. The corresponding reconstructions are shown in Figures

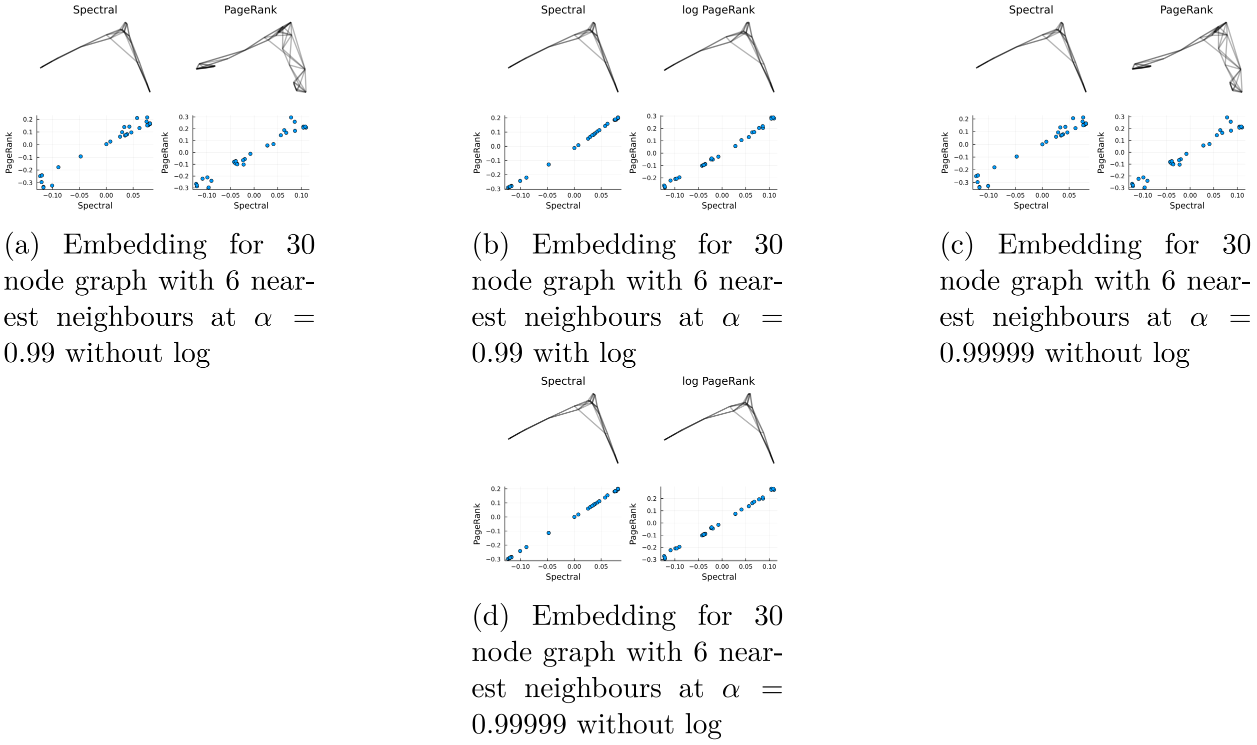
Table 4.1. Error between PageRank embedding and spectral embedding for different graphs at a low teleportation probability, $\alpha = 0.99$ and at a higher one $\alpha = 0.99999$

Graph	$\alpha = 0.99$	$\alpha = 0.99$ with log	$\alpha = 0.99999$	$\alpha = 0.99999$ with log
30-6 nearest neighbour	3.27%	0.06%	2.89%	0.05%
3000-6 nearest neighbour	47.6%	0.37%	5.06%	2.88%
10000-6 nearest neighbour	170.75%	2.13%	13.5%	1.76%
30 chain	26.88%	0.47%	28.42%	6.02%
3000 chain	2858.82%	1.06%	30.38%	0.75%
Minnesota(2640)	16.07%	1.97%	11.15%	0.44%
Tapir(1024)	10.17%	1.13%	15.41%	0.66%
Original(5500)	75.92%	4.11%	81.14%	0.33%
sbm(100,3,0.25,0.001)	8.18%	1.93%	8.24%	2.69%
sbm(50,60,0.001,0.005)	51.77%	15.22%	51.32%	67.25%
sbm(1000,3,0.001,0.005)	47.35%	16.93%	45.78%	89.39%
sbm(50,60,0.25,0.005)	17.88%	15.22%	90.13%	402.27%
sbm(1000,3,0.25,0.001)	53.7%	1.04%	16.21%	15.73%

4.1 Dependence of Rayleigh error on sampling columns

In this section, we investigate the dependence of the error in Table 4.1 with respect to the seed nodes, that is with respect to which columns of the PageRank matrix are sampled. For a theoretical understanding, we refer to the theory in [6] regarding sensitivity to $\mathbf{v}$. From equation 3.1 and 3.2, we know fundamentally that as $\alpha \to 1$, the sensitivity on $\mathbf{v}$ reduces, as the PageRank scores become more and more sensitive on the structure of the graph. More formally, as in [6],

$$\begin{aligned} \frac{d\mathbf{x}}{d\mathbf{v}} &= (1-\alpha)(\mathbf{I} - \alpha\mathbf{P})^{-1} \\ ||\frac{d\mathbf{x}}{d\mathbf{v}}||_1 &= 1 \end{aligned} \tag{4.6}$$



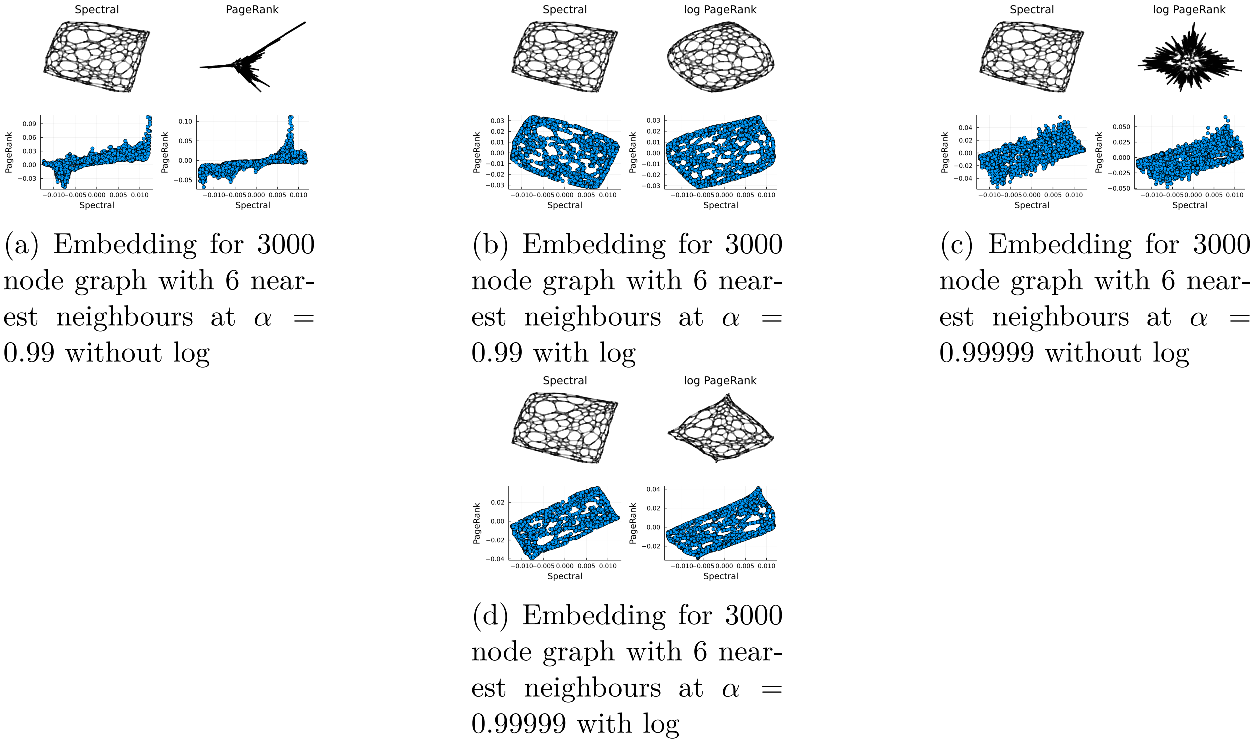
(a) Embedding for 30 node graph with 6 nearest neighbours at $\alpha = 0.99$ without log

(b) Embedding for 30 node graph with 6 nearest neighbours at $\alpha = 0.99$ with log

(c) Embedding for 30 node graph with 6 nearest neighbours at $\alpha = 0.99999$ without log

(d) Embedding for 30 node graph with 6 nearest neighbours at $\alpha = 0.99999$ without log

Figure 4.1. Embedding for 30 node graph with 6 nearest neighbours



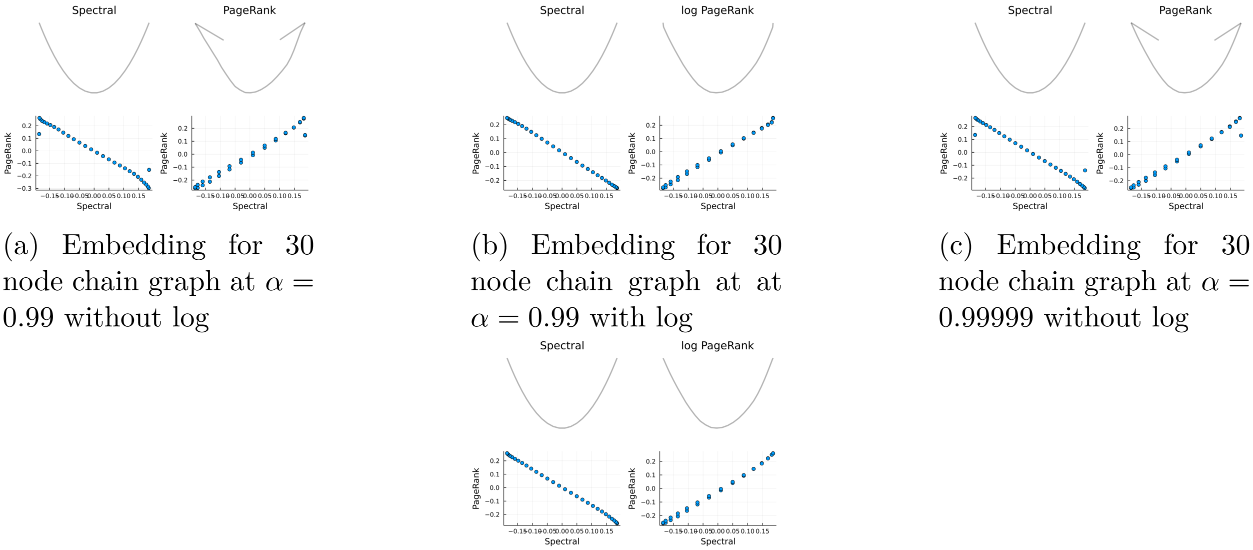
(a) Embedding for 3000 node graph with 6 nearest neighbours at $\alpha = 0.99$ without log

(b) Embedding for 3000 node graph with 6 nearest neighbours at $\alpha = 0.99$ with log

(c) Embedding for 3000 node graph with 6 nearest neighbours at $\alpha = 0.99999$ without log

(d) Embedding for 3000 node graph with 6 nearest neighbours at $\alpha = 0.99999$ with log

Figure 4.2. Embedding for 3000 node graph with 6 nearest neighbours



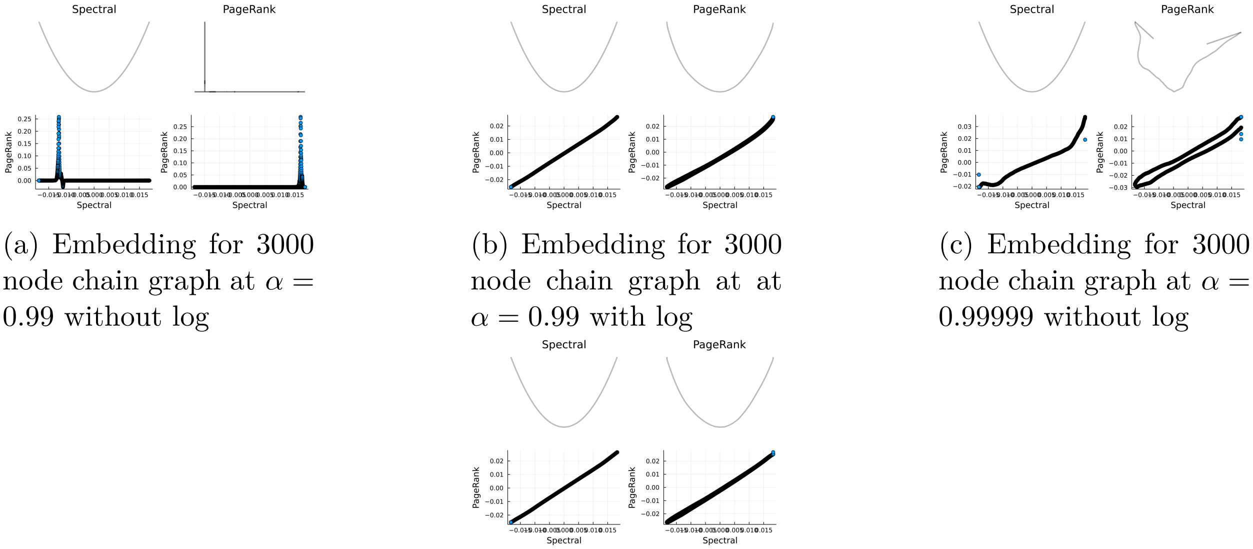
(a) Embedding for 30 node chain graph at $\alpha = 0.99$ without log

(b) Embedding for 30 node chain graph at at $\alpha = 0.99$ with log

(c) Embedding for 30 node chain graph at $\alpha = 0.99999$ without log

(d) Embedding for 30 node chain graph at $\alpha = 0.99999$ with log

Figure 4.3. Embedding for 30 node chain graph



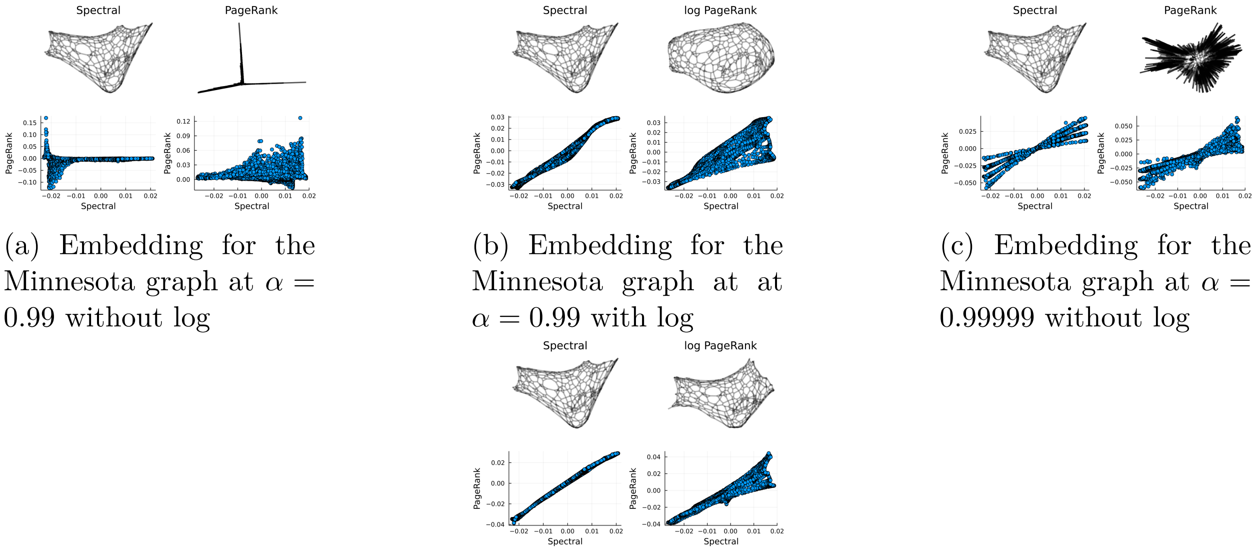
(a) Embedding for 3000 node chain graph at $\alpha = 0.99$ without log

(b) Embedding for 3000 node chain graph at at $\alpha = 0.99$ with log

(c) Embedding for 3000 node chain graph at $\alpha = 0.99999$ without log

(d) Embedding for 3000 node chain graph at $\alpha = 0.99999$ with log

Figure 4.4. Embedding for 3000 node chain graph



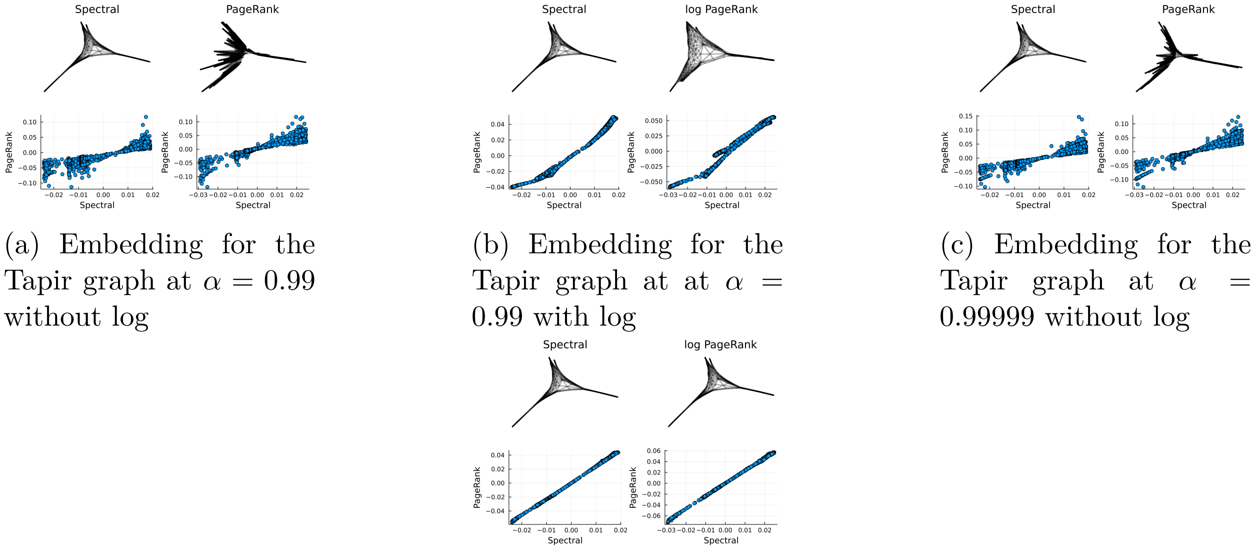
(a) Embedding for the Minnesota graph at $\alpha = 0.99$ without log

(b) Embedding for the Minnesota graph at at $\alpha = 0.99$ with log

(c) Embedding for the Minnesota graph at $\alpha = 0.99999$ without log

(d) Embedding for the Minnesota graph at $\alpha = 0.99999$ with log

Figure 4.5. Embedding for the Minnesota graph



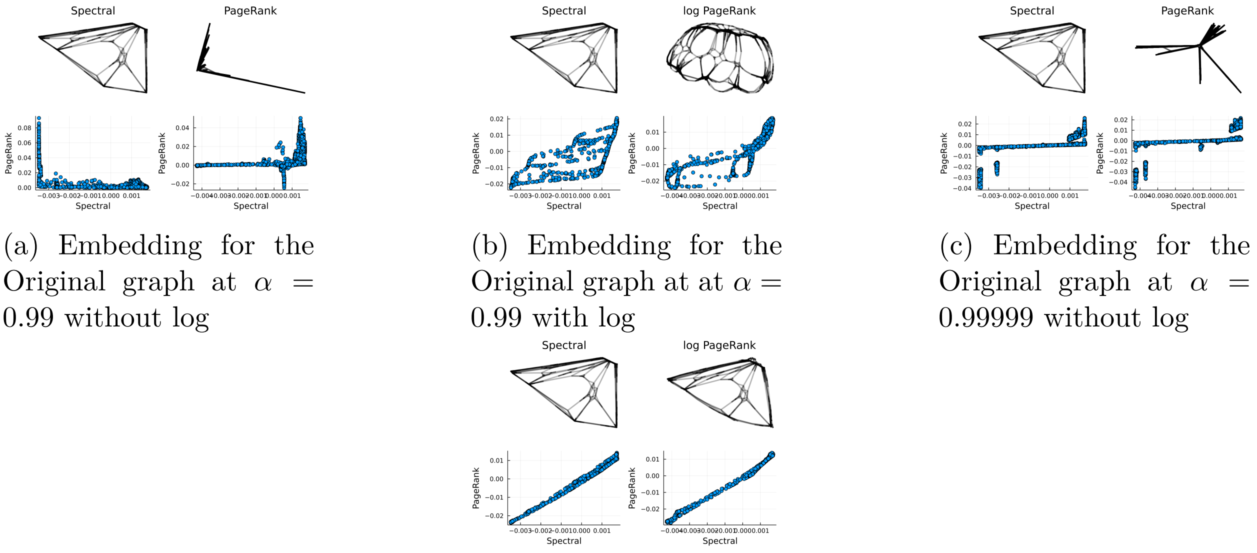
(a) Embedding for the Tapir graph at $\alpha = 0.99$ without log

(b) Embedding for the Tapir graph at at $\alpha = 0.99$ with log

(c) Embedding for the Tapir graph at $\alpha = 0.99999$ without log

(d) Embedding for the Tapir graph at $\alpha = 0.99999$ with log

Figure 4.6. Embedding for the Tapir graph



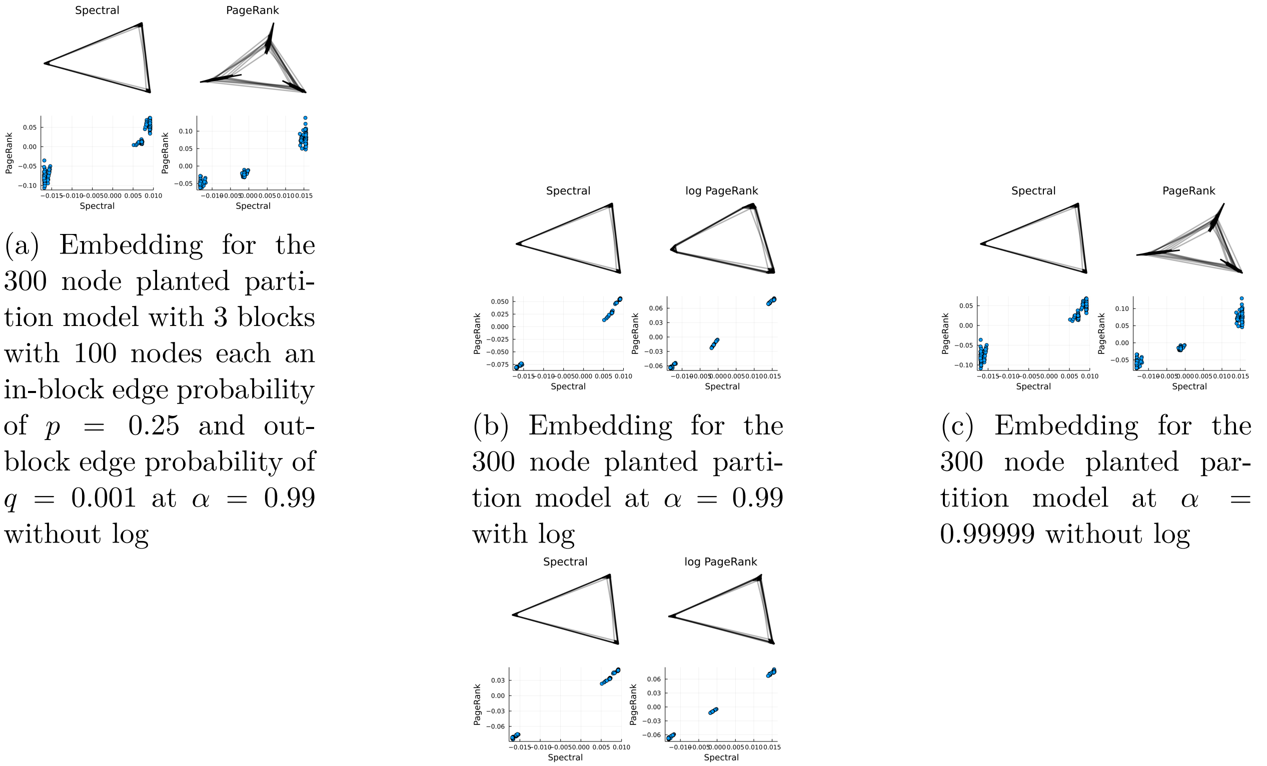
(a) Embedding for the Original graph at $\alpha = 0.99$ without log

(b) Embedding for the Original graph at at $\alpha = 0.99$ with log

(c) Embedding for the Original graph at $\alpha = 0.99999$ without log

(d) Embedding for the Original graph at $\alpha = 0.99999$ with log

Figure 4.7. Embedding for the Original graph



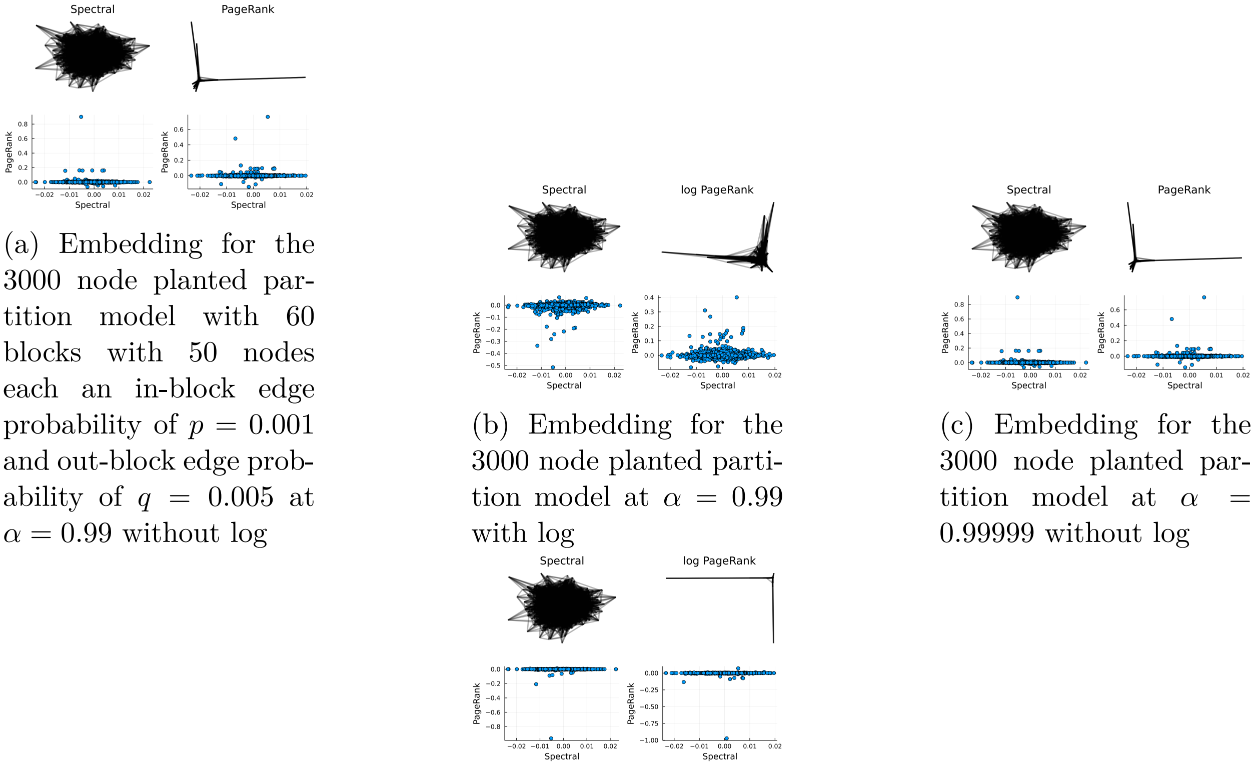
(a) Embedding for the 300 node planted partition model with 3 blocks with 100 nodes each an in-block edge probability of $p = 0.25$ and out-block edge probability of $q = 0.001$ at $\alpha = 0.99$ without log

(b) Embedding for the 300 node planted partition model at $\alpha = 0.99$ with log

(c) Embedding for the 300 node planted partition model at $\alpha = 0.99999$ without log

(d) Embedding for the 300 node planted partition model at $\alpha = 0.99999$ with log

Figure 4.8. Embedding for the 300 node planted partition model



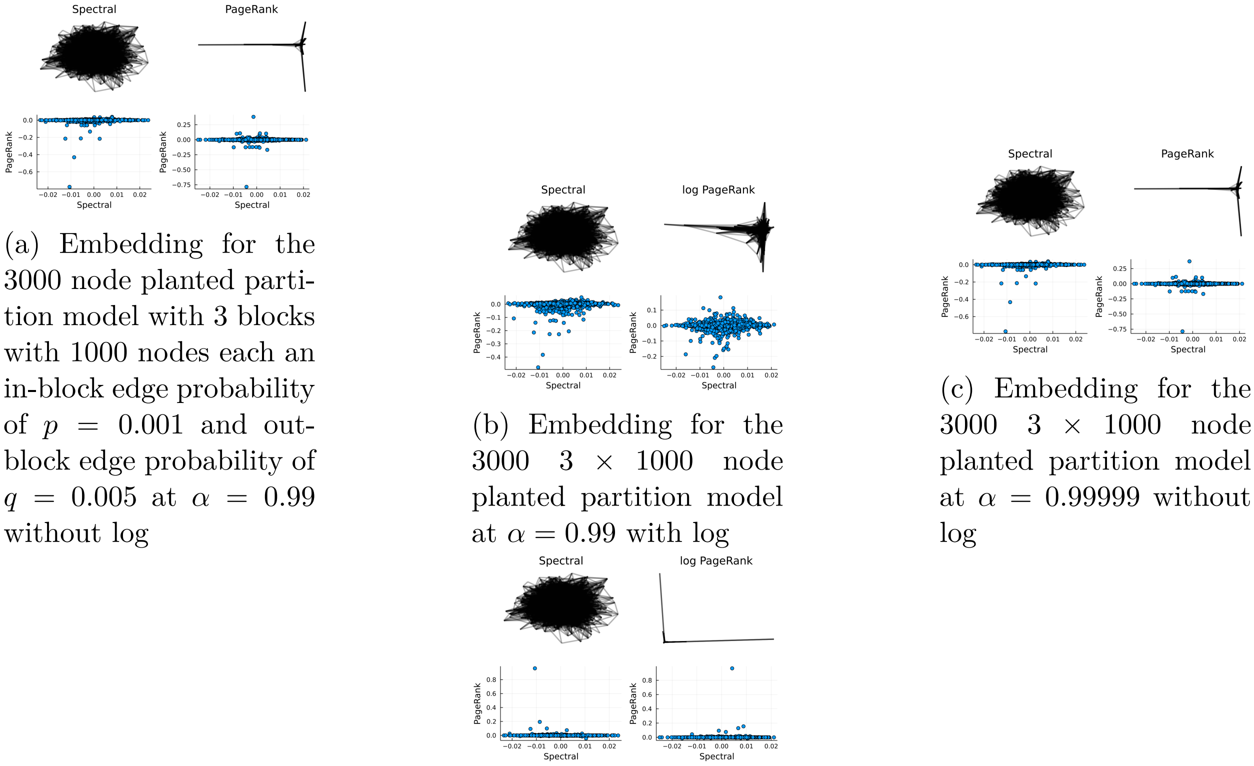
(a) Embedding for the 3000 node planted partition model with 60 blocks with 50 nodes each an in-block edge probability of $p = 0.001$ and out-block edge probability of $q = 0.005$ at $\alpha = 0.99$ without log

(b) Embedding for the 3000 node planted partition model at $\alpha = 0.99$ with log

(c) Embedding for the 3000 node planted partition model at $\alpha = 0.99999$ without log

(d) Embedding for the 3000 node planted partition model at $\alpha = 0.99999$ with log

Figure 4.9. Embedding for the 3000(60 × 50) node planted partition model



(a) Embedding for the 3000 node planted partition model with 3 blocks with 1000 nodes each an in-block edge probability of $p = 0.001$ and out-block edge probability of $q = 0.005$ at $\alpha = 0.99$ without log

(b) Embedding for the 3000 3×1000 node planted partition model at $\alpha = 0.99$ with log

(c) Embedding for the 3000 3×1000 node planted partition model at $\alpha = 0.99999$ without log

(d) Embedding for the 3000 3×1000 node planted partition model at $\alpha = 0.99999$ with log

Figure 4.10. Embedding for the 3000(3×1000) node planted partition model

which hints at a linear relation between the PageRank vector, $\mathbf{x}$, and the personalization vector $\mathbf{v}$.

The tables below (4.2 and 4.3) summarize the variance of the error between spectral embedding and PageRank embedding for the graphs discussed in Table 4.1 with respect to the columns samples. Since we know that the error has a tendency to go down for $\alpha = 0.99999$ or with element-wise log of PageRank at $\alpha = 0.99$, we will focus on those two cases for different columns of PageRank. From the tables we note the following points.

1. The reduction in error is not much as the number of sampled columns increases.
2. The variance seems to decrease with increase in size of the graph.
3. For the small graphs(with 30 nodes), the results are elusive.

Table 4.2. Error variation with column for $\alpha = 0.99999$. The percentage indicated in the column headings are the fraction of the nodes as seeds. Each entry is the variance, the maximum and the minimum for 50 trials.

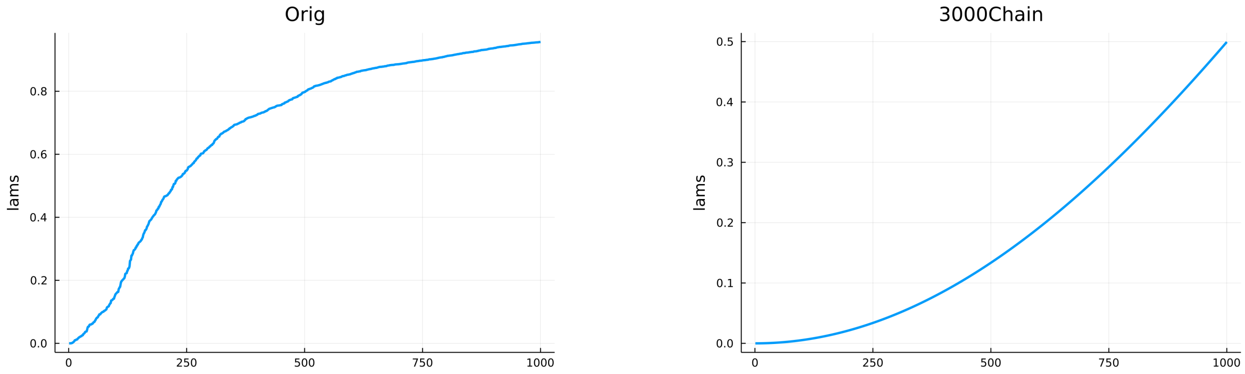
Graph	$\frac{k}{n} = 4\%$	7%	10%
30 - 6 nearest neighbour	23.68, 0.04, 23.5	23.68, 0.04, 23.5	23.68, 0.04, 23.5
3000 - 6 nearest neighbour	0.001,0.05,0.15	0.001,0.04,0.15	0.00096,0.05,0.14
10000 - 6 nearest neighbour	0.0001,0.09,0.13	2.4e-5,0.09,0.11	3.7e-5,0.09,0.12
30 chain	9.2,0.01,17.8	9.2,0.01,17.8	9.2,0.01,17.8
3000 chain	0.02,0.06,0.85	0.01,0.24,0.75	0.01,0.17,0.75
Minnesota(2046)	0.0001,0.09,0.13	6.3e-5,0.09,0.12	4.6e-5,0.09,0.12
Tapir(1024)	0.0006,0.07,0.18	0.0005,0.08,0.18	0.0002,0.11,0.18
Original(5500)	3.1e-7,0.8,0.8	1.5e-7,0.81,0.82	1.0e-7,0.8,0.8

4.2 Embeddings depend on the Fiedler value

This section reports our analysis of the eigenvalues of the symmetric Laplacian for the graphs and tries to identify their trends to underpin the graph characteristics where PageRank embeddings do not yield meaningful representations. The degree distributions did not seem to play a large or a consistent role in the deciding factor so we do not analyse them here. Hence, we will not focus on those here. The plots below show the trajectory of increasing

Table 4.3. Error variation with column for log of PageRank with $\alpha = 0.99$.The percentage indicated in the column headings are the fraction of the nodes as seeds. Each entry is the variance, the maximum and the minimum for 50 trials.

Graph	$\frac{k}{n} = 4\%$	7%	10%
30 - 6 nearest neighbour	21.5,0.01,22.8	21.5,0.01,22.8	21.5,0.01,22.8
3000 - 6 nearest neighbour	0.0002,0.0003,0.05	0.0002,0.0006,0.05	0.0002,0.0001,0.05
10000 - 6 nearest neighbour	5.1e-5,0.0005,0.02	1.6e-5,0.0006,0.02	1.6e-5,0.001,0.02
30 chain	14.85,0.003,24.6	14.85,0.003,24.6	14.85,0.003,24.6
3000 chain	0.0003,0.0001,0.05	0.0001,0.0003,0.06	0.0001,0.0008,0.04
Minnesota(2046)	5.3e-5,0.004,0.03	2.8e-5,0.01,0.03	1.7e-5,0.01,0.03
Tapir(1024)	1.0e-5,0.0004,0.02	1.1e-5,0.002,0.01	5.0e-6,0.004,0.01
Original(5500)	1.3e-5,0.03,0.04	7.3e-6,0.03,0.04	5.5e-6,0.03,0.04

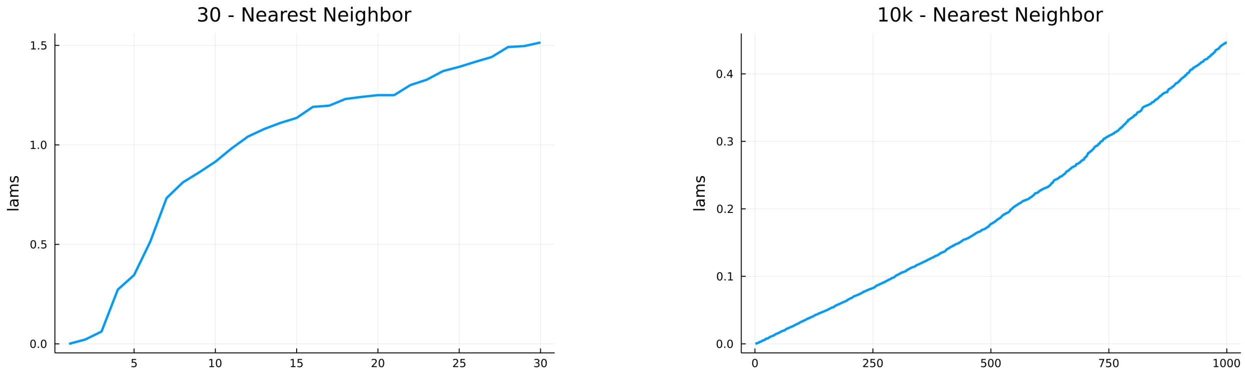


(a) 5500 node original graph

(b) 3000 node Chain graph

Figure 4.11. Eigenvalue trends for the 5500 node original graph and the 3000 node chain graph

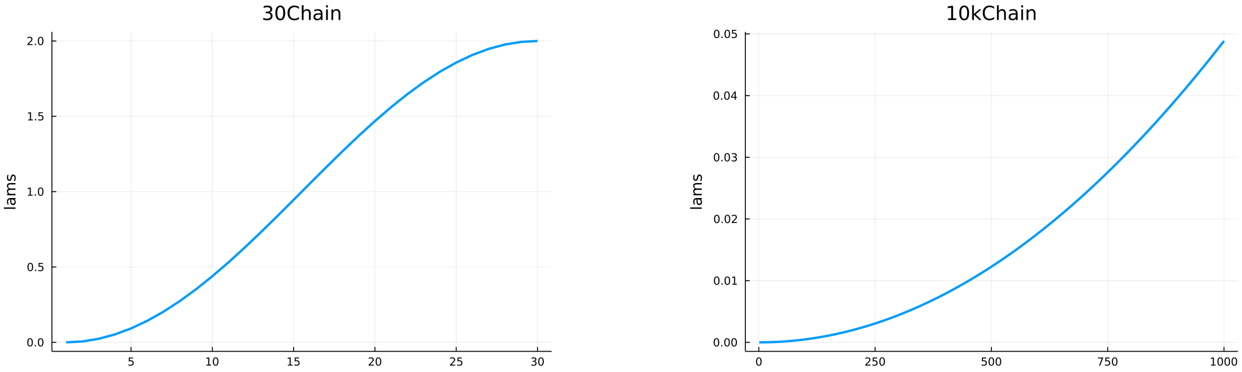
eigenvalues of the symmetric Laplacian for nearest neighbor graphs, some real-world graphs, chain graphs and random graphs.



(a) 30 node nearest neighbour graph

(b) 10000 node nearest neighbour graph

Figure 4.12. Eigenvalue trends for the 30 node nearest neighbour graph and the 10000 node nearest neighbour graph



(a) Eigenvalue trends for the 30 node chain graph

(b) Eigenvalue trends for the 10000 node chain graph

Figure 4.14. Eigenvalue trends for the 30 node chain graph and the 10000 node chain graph

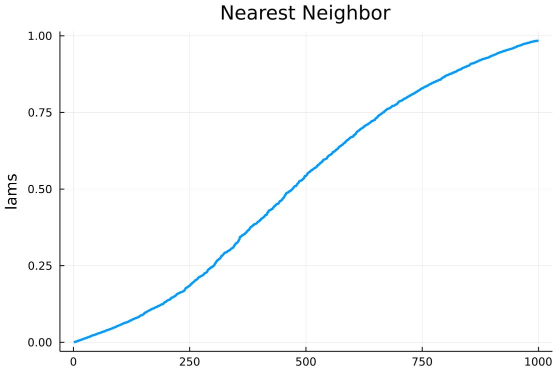
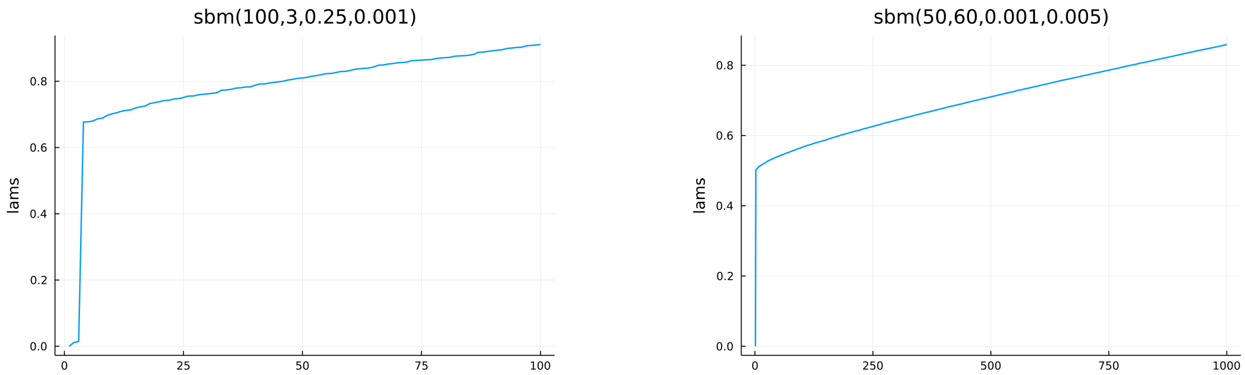


Figure 4.13. Eigenvalue trends for the 3000 node nearest neighbour graph with 6 nearest neighbours

Notice the gradual increase in eigenvalues for the planar graphs, as compared to the abrupt increase to the stochastic block model. We also experimented with some real-world graphs. Although they are not artificially generated like the nearest neighbour graphs, their eigenvalues followed the same trend as the above graphs.

The above plots tell us that the graphs for which the equivalence seemed to work should have a non-zero double differential, while the others should have a zero double differential because the plots are monotonically increasing. For the discrete eigen-values, a second order finite difference would be given as

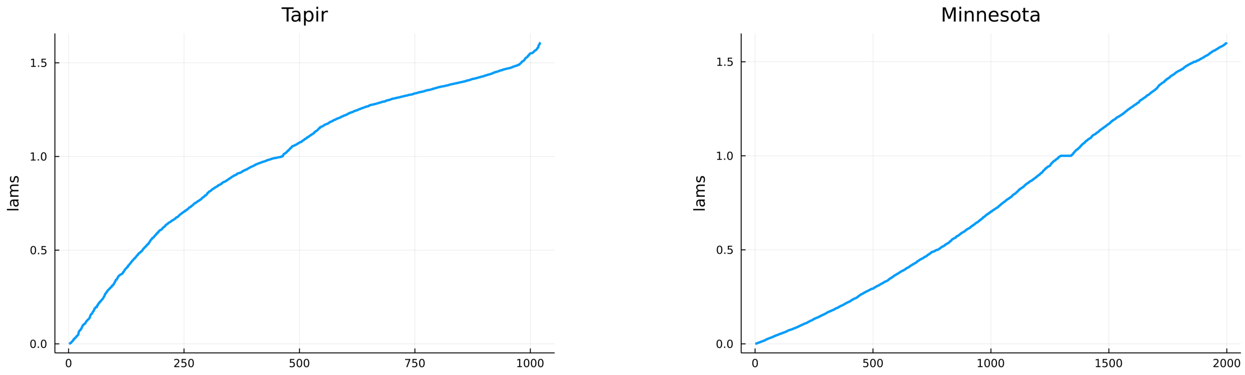
$$\lambda_s = \lambda_{i+2} - 2\lambda_{i+1} + \lambda_i \tag{4.7}$$



(a) 3 blocks each with 100 nodes and $p = 0.25$ and $q = 0.001$

(b) 60 blocks each with 50 nodes and $p = 0.001$ and $q = 0.005$

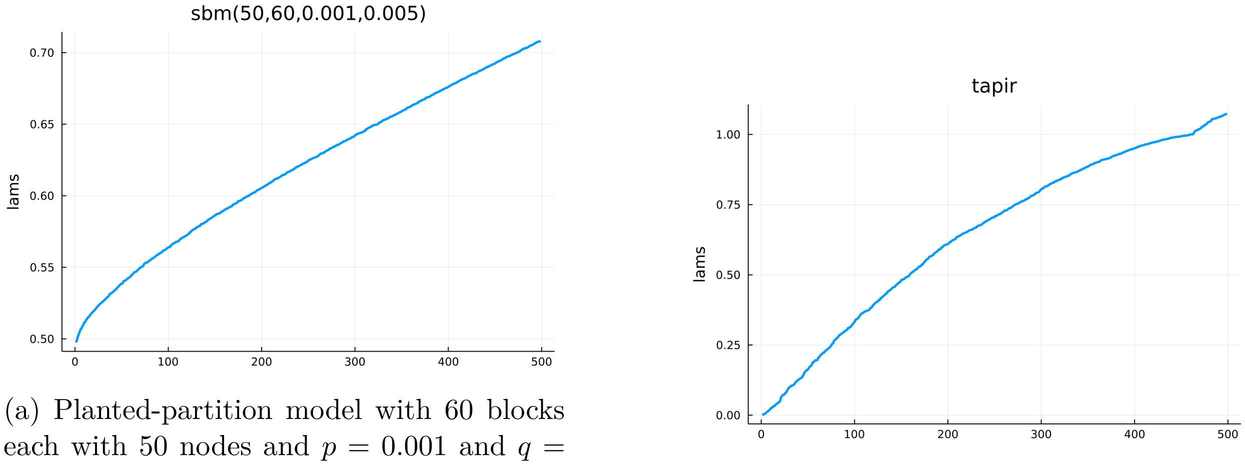
Figure 4.15. Eigenvalue trends for the planted-partition model with 3 blocks each with 100 nodes and $p = 0.25$ and $q = 0.001$ and with 60 blocks each with 50 nodes and $p = 0.001$ and $q = 0.005$



(a) Tapir network

(b) Minnesota network

Figure 4.16. Eigenvalue trends for the Tapir network and the Minnesota network



(a) Planted-partition model with 60 blocks each with 50 nodes and $p = 0.001$ and $q = 0.005$

(b) Tapir graph

Figure 4.17. Eigenvalue trends for the planted-partition model with 60 blocks each with 50 nodes and $p = 0.001$ and $q = 0.005$ and the Tapir graph without the trivial eigenvalue (4.15b)

```
AVERAGE SECOND ORDER DIFFERENCE DEFINITION
avg_second_slope = 0
for i=2:(size(lams,1)-2)
    avg_second_slope = lams[i+2]-2*lams[i+1]+lams[i]
end
avg_second_slope = avg_s_s/size(lams,1)
```

Somehow the average second order difference (4.2) for the first 500 eigenvalues could not differentiate between the two classes of graphs. The value returned by (4.2) for the Tapir graph and the 3000 node planted-partition model were similar. On a closer look of the eigenvalue plots without the first eigenvalue, (4.17a) and (4.17b) show a similar trend, as opposed to the graph in figure (4.18), while the embedding technique did not work for the first, but did for the last two.

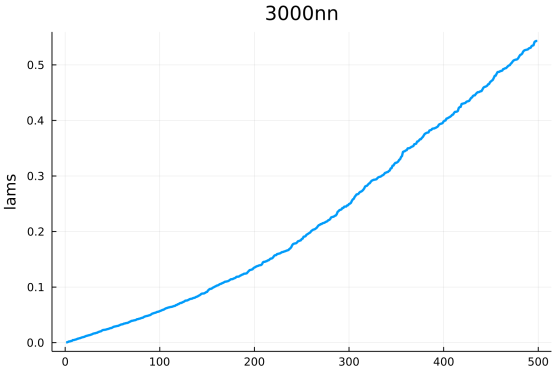


Figure 4.18. Eigenvalue trends for the 3000 nearest neighbour graph without the trivial eigenvalue (4.13)

From the plots in figures (4.11a - 4.16b), if not the second order slope of the graphs, the only thing that differentiates the two is the abrupt increment of the eigenvalues for 4.15b and 4.15a after the trivial eigenvalue. Hence, using their absolute values, we attempted to differentiate between the two classes of graph using the successive difference between the lowest 3 eigenvalues of the symmetric Laplacian of the graph through the following ratio,

$$\lambda_q = \frac{\lambda_3 - \lambda_2}{\lambda_2 - \lambda_1} \tag{4.8}$$

The value attained by different graphs for this quantity is given in Table 4.4.

Table 4.4. Eigenvalue Characterization

Graph	λ_q	λ_d
3000-9 nearest neighbour	0.025	0.0010
10000-9 nearest neighbour	0.042	0.00029
3000 chain	2.99	4.39e-6
Minnesota(2640)	1.5	0.00017
Tapir(1024)	0.5	0.00052
sbm(100,3,0.25,0.001)	0.46	0.0068
sbm(50,60,0.001,0.005)	0.0023	0.496
sbm(1000,10,0.001,0.005)	0.0018	0.71
sbm(100,100,0.001,0.005)	0.00096	0.72

Although λ_q seems to be an indicator of degree of the graph since it decreased as the degree increased, it is clearly not able to differentiate between the classes because although

λ_q for the 3000 node nearest neighbour graph and that for the 10000 node planted-partition model are almost similar, this technique does not work on the latter. But even so, the order of the eigenvalues were clearly different for the graphs on which this technique did and did not work. Hence we define a new quantity in terms of the eigenvalues

$$\lambda_d = |(\lambda_3 - \lambda_2) - (\lambda_2 - \lambda_1)| \tag{4.9}$$

The value of λ_d for different graphs have been noted in Table 4.4. The graphs for which the embedding technique did not work have a much higher value compared to those where it worked. From a second perspective, the formulation for λ_d, corresponds to the second order finite difference at the Fiedler value where the abrupt change occurs. For example, for the (50,60,0.001,0.005)-planted partition graph for which the eigenvalue trend is shown in figure 4.15b, the Fiedler value had an abrupt jump which is characterized well by the definition of λ_d. Additionally, the graphs for which it did work, λ_d, with some dependence on the degree, seems to be of the order $\frac{1}{n}$. This redirected us to [25] which gives an upper bound to the Fiedler value for planar graphs in terms of the size of the graph.

Theorem 4.2.1. *Theorem 3.3([25]): Let* $\mathbf{G}$ *be a planar graph on* n *nodes of degree at most* δ*. The the Fiedler value of* $\mathbf{G}$ *is at most* $\frac{8\delta}{n}$*.*

The theory in [25] along with our observations, prove that for planar graphs, spectral embedding is equivalent to element-wise logarithm of PageRank embedding for a high teleportation probability. Having established this crucial piece, we now zoom into the properties of the planar graphs that help validate (or in-validate) the general existence of the other observations.

4.2.1 Approximately equal eigenvalues

From Table 4.1 and figures (4.1a-4.8a) we understand that good reconstructions imply a constant scale between the spectral and the PageRank embedding as shown in the bottom subplots of the figures. However for the 3000 node nearest neighbour graph (figure 4.2a), even with the low error we see that the scale between the spectral and PageRank embedding is not constant as in the other examples (4.6a). The case represented by the Figure 4.2a hints at a more general situation where the eigenvalues corresponding to the eigenvectors are approximately equal. For the cases at hand, the corresponding eigenvalues of the 3000 node nearest neighbour graph (4.2a) were as follows

$$\lambda = [0.0, 0.00056, 0.00059, 0.0012, 0.0021] \tag{4.10}$$

and that of the Tapir graph (4.6a) were as follows,

$$\lambda = [0.0, 0.0012, 0.002, 0.004, 0.004] \tag{4.11}$$

Using the example of above two cases, we can say that for $\lambda_3 - \lambda_2 < \frac{1}{n}$, we can assume that the two eigenvalues are close, hence, the corresponding eigenvectors make an almost 2 dimensional invariant subspace. For $\alpha \to 1$, or for PageRank with log at $\alpha = 0.99$ when spectral embeddings are similar to PageRank embeddings, the PageRank vectors are sampled from the same 2 dimensional invariant subspace created by the eigenvectors corresponding to the equivalent eigenvalues. Therefore, as in the case of the 3000 node nearest neighbour graph, we have $\lambda_3 - \lambda_2 = 3.4e - 5 < 0.0003$, and therefore the PageRank embedding vectors are sampled from the same subspace as the spectral embedding vectors. In other words, the PageRank embedding vector is a rotated version of the spectral embedding vector and hence plotting them against each other, creates the rectangular figure as in figure 4.2a. While for the Tapir graph, $\lambda_3 - \lambda_2 = 0.0006 \sim 0.00097$, and hence the corresponding PageRank vectors are sampled from the 1-dimensional subspace created by the corresponding eigenvector which ensures the linear scale in the figure 4.6a. Figure 4.19 shows this rotation of the embeddings where the colours denote the node represented by embedding.

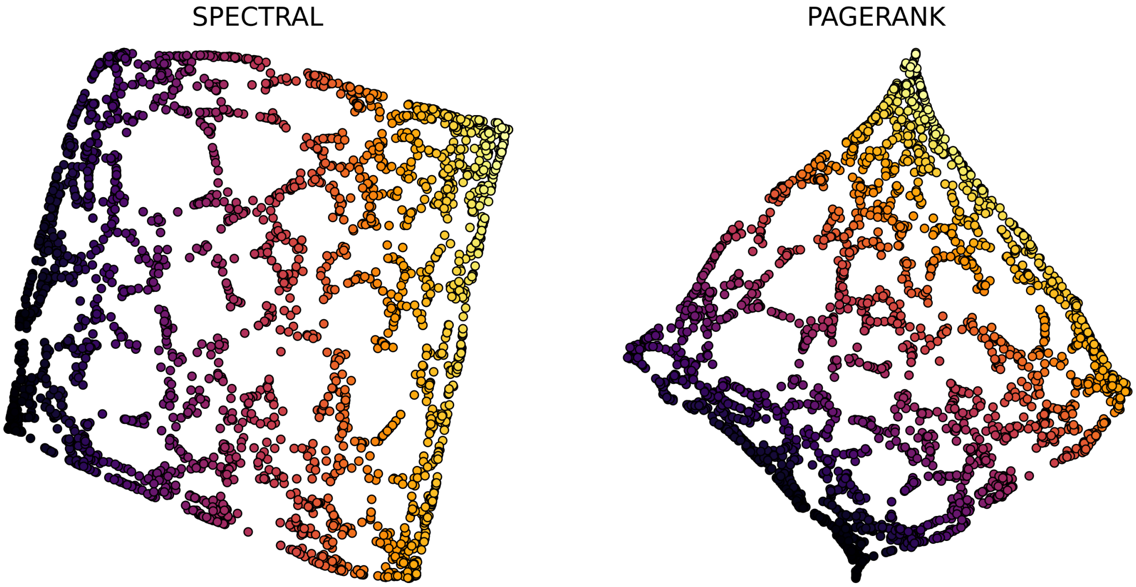


Figure 4.19. PageRank and the spectral embedding vectors being sampled from the 2 dimensional invariant subspace created by the eigenvectors corresponding to almost equal eigenvalues

4.3 Theoretical analysis of the spectrum

Following the notations described above, since the similarity is between the PageRank and the eigenvectors of symmetric Laplacian, we attempted to reason about the above observations as follows.

$$\begin{aligned}\tilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{G}\mathbf{D}^{-1/2} &= \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{\mathbf{T}} \\ \rightarrow \left(\mathbf{I} - \alpha\mathbf{G}\mathbf{D}^{-1}\right)^{-1} &= \mathbf{D}^{1/2}\mathbf{Q}(\mathbf{I} - \alpha\boldsymbol{\mathcal{E}})^{-1}\mathbf{Q}^{\mathbf{T}}\mathbf{D}^{-1/2} \\ \mathbf{X} &= (1-\alpha)\left(\mathbf{I} - \alpha\mathbf{G}\mathbf{D}^{-1}\right)^{-1} \\ &= (1-\alpha)\mathbf{D}\mathbf{Z}(\mathbf{I} - \alpha\boldsymbol{\mathcal{E}})^{-1}\mathbf{Z}^{\mathbf{T}}\end{aligned} \tag{4.12}$$

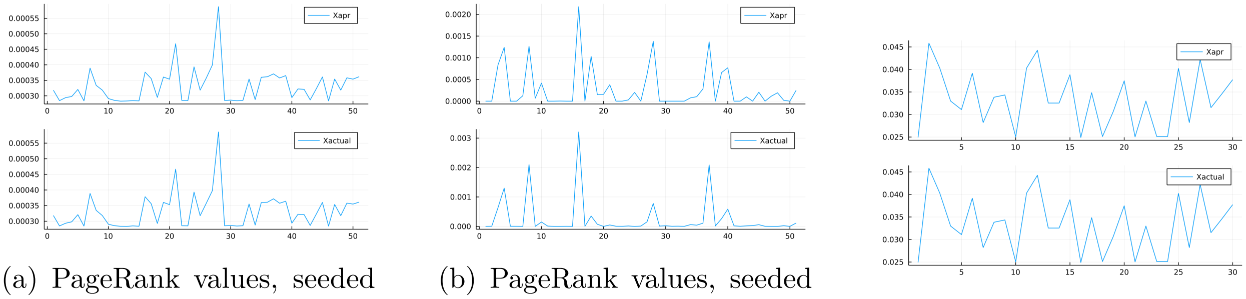
where $\mathbf{Z} = \mathbf{D}^{-1/2}\mathbf{Q}, \quad \boldsymbol{\mathcal{E}} = \mathbf{I} - \mathbf{\Lambda}$

The above formulation does not offer a conducive enough way to be treated with an element-wise logarithm operation as expanding the formulation of $\mathbf{X}$ would lead to a summation series for each element. Since the element-wise log seems to be playing a crucial role with small teleportation probability for large graphs, we expand the formulation element-wise and use the properties of the eigenvalues discussed in this section above.

$$\begin{aligned}
\mathbf{X} &= (1-\alpha)\mathbf{DZ}(\mathbf{I}-\alpha\boldsymbol{\mathcal{E}})^{\mathbf{-1}}\mathbf{Z^T} \\
\mathbf{X} &= \begin{bmatrix} \tfrac{1-\alpha}{1-\alpha\epsilon_1}\mathbf{dz_1} \cdots \tfrac{1-\alpha}{1-\alpha\epsilon_n}\mathbf{dz_n} \end{bmatrix} \begin{bmatrix} \mathbf{z_1}^T \\ \vdots \\ \mathbf{z_n}^T \end{bmatrix} \\
&\text{where } \mathbf{d} = \mathbf{De} \\
X_{i,j} = (1-\alpha)(I-\alpha GD^{-1})^{-1}_{i,j} &= d_i \left[\frac{1-\alpha}{1-\alpha\epsilon_1} z_{1i}z_{1j} + \frac{1-\alpha}{1-\alpha\epsilon_2} z_{2i}z_{2j} + \cdots + \frac{1-\alpha}{1-\alpha\epsilon_n} z_{ni}z_{nj} \right] \\
\rightarrow X_{i,j} &= \sum_{t=1}^{n} \frac{1-\alpha}{1-\alpha\epsilon_t} z_{ti}z_{tj}d_i \\
\log(X_{i,j}) &= \log(\sum_{t=1}^{n} \frac{1-\alpha}{1-\alpha\epsilon_t} z_{ti}z_{tj}d_i) \\
&= \log(1-\alpha) + \log(\sum_{t=1}^{n} \frac{z_{ti}z_{tj}}{1-\alpha\epsilon_t}) + \log(d_i)
\end{aligned} \tag{4.13}$$

Before using this formulation for justifying the need for large α and log operation on PageRank, a computational verification of the above two formulations (with and without log) generated the following results. We discuss these results here because, although with respect to this formulation, they only hint at the low rank reconstruction error of PageRank through the formulation, they also justify the resemblance between PageRank and spectral embeddings as well as the equivalence between the singular vectors of the PageRank matrix and the average of the PageRank columns according to the observations mentioned in (1.1).

The notable implications of the images are that for smaller graphs($\sim$ 30 nodes), the approximations fit well for lower values of α and there is negligible improvement on increasing the value of α. However, the larger the graph became in terms of nodes, α needed to be increased to decrease the reconstruction error. Recall, we noticed a similar effect for the approximation of the average of PageRank columns by the left singular vector. Although, similar to the smaller graphs, for larger graphs, after a particular value of α, the improvement

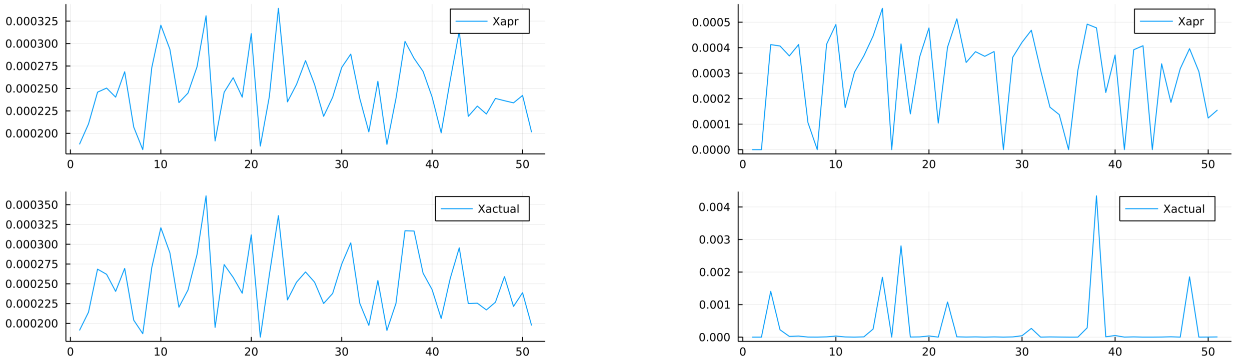


(a) PageRank values, seeded on node 2, for nodes 50 - 100 on the 3000 nodes graph with 9 nearest neighbours reconstructed with $\alpha = 0.99999$ had a reconstruction error of 0.44%

(b) PageRank values, seeded on node 2, for nodes 50 - 100 on the 3000 nodes graph with 9 nearest neighbours reconstructed with $\alpha = 0.99$ had a reconstruction error of 110.96%

(c) PageRank values, seeded on node 2, for nodes 1 - 30 on the 30 nodes graph with 9 nearest neighbours reconstructed with $\alpha = 0.99$ had a reconstruction error of e^{-13}%

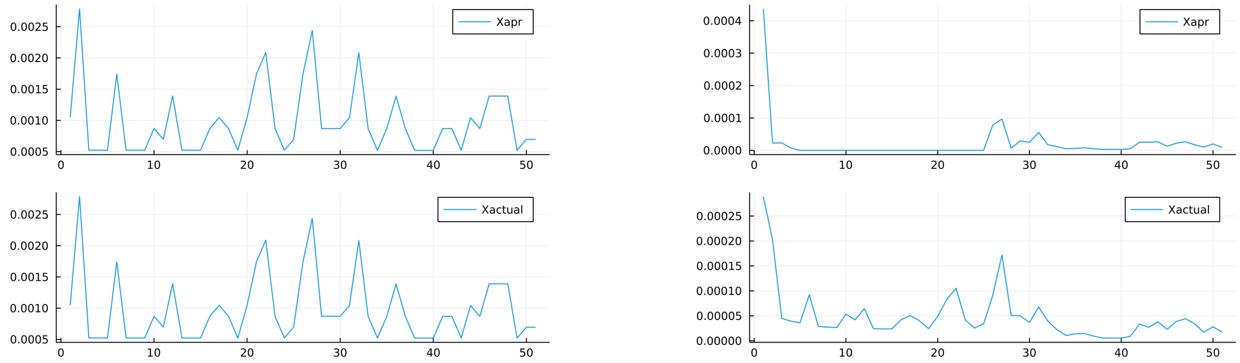
Figure 4.20. Approximate PageRank values for nearest neighbour graphs



(a) PageRank values, seeded on node 2, for nodes 50 - 100 on the original graph reconstructed with $\alpha = 0.99999$ had a reconstruction error of 3.7%

(b) PageRank values, seeded on node 2, for nodes 50 - 100 on the original graph reconstructed with $\alpha = 0.99$ had a reconstruction error of 157.8%

Figure 4.21. Approximate PageRank values for original graph with different α



(a) PageRank values, seeded on node 2, for nodes 50 - 100 on the Tapir graph reconstructed with $\alpha = 0.99999$ had a reconstruction error of 0.23%

(b) PageRank values, seeded on node 2, for nodes 50 - 100 on the Tapir graph reconstructed with $\alpha = 0.99$ had a reconstruction error of 102.8%

Figure 4.22. Approximate PageRank values for Tapir graph with different α

in reconstruction was negligible. This property seemed to hint at matching order between α and the size of the graph and hence, ϵ_t.

This formulation, itself does not work for larger graphs and smaller α, that is the conditions where log operation was necessary. The error for the same can be bounded as a low rank reconstruction error with α.

$$\mathbf{X} = (1-\alpha)\mathbf{DZ}(I-\alpha\boldsymbol{\mathcal{E}})^{-1}\mathbf{Z}^T = \sum_{t=1}^{n} \frac{1-\alpha}{1-\alpha\epsilon_t}\mathbf{z_t}\mathbf{z_t}^T \tag{4.14}$$

For reconstruction, in our experiments, we only use the top 5 vectors regardless of the graph size. With $\tilde{X}$ as the approximated matrix, the error can be written in terms of the Frobenius norm,

$$\begin{aligned} ||\mathbf{X}-\tilde{\mathbf{X}}||_F &= ||\sum_{t=1}^{n} \frac{1-\alpha}{1-\alpha\epsilon_t}\mathbf{z_t}\mathbf{z_t}^T - \sum_{t=1}^{5} \frac{1-\alpha}{1-\alpha\epsilon_t}\mathbf{z_t}\mathbf{z_t}^T||_F \\ &= ||\sum_{t=6}^{n} \frac{1-\alpha}{1-\alpha\epsilon_t}\mathbf{z_t}\mathbf{z_t}^T||_F \\ &= \sum_{t=6}^{n} \frac{1}{d_t}(\frac{1-\alpha}{1-\alpha\epsilon_t})^2 \end{aligned} \tag{4.15}$$

This explains the increasing error in a larger graph compared to a smaller one for the same α, as the missing number of eigenvectors are more for larger graphs. However the decrease in error with α, for large graphs with same number of vectors is not apparent. The behavior of the term $\frac{1-\alpha}{1-\alpha\epsilon_t}$ explains that. This term reduces for successive eigenvectors, that is creating a lower weight for the eigenvectors corresponding to the larger eigenvalues. While in a small graph with $\alpha = 0.99$, this reduction is significant, in large graphs the same α does not cause significant reduction in the eigenvectors corresponding to the larger eigenvalues. But with larger α, the term produces smaller values for larger eigenvectors, hence reducing the error. Besides reducing the error this term also has other effects on PageRank that we will not discuss here, but in section (5.2) where the context makes more sense.

Therefore, this formulation can not be used to explain the significance of element-wise log operation while working with large graphs and smaller α. And to note the obvious, none of the above experiments and formulation explain why none of this resemblance work for random graphs (or any graph with high second order difference). Having exhausted the apparent spectral properties, we attempt to dive in details by investigating one of the most simple graph there is - the chain graph.

5. CHAIN GRAPH

We derive a closed form expression of PageRank to understand the effect of element-wise log on it. Since the chain graph is the simplest planar graph there is, we develop the PageRank expression for chain graphs. This section records the derivation and further analysis of the spectral and PageRank embeddings with respect to the observations in (1.1).

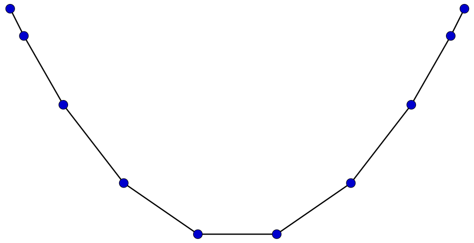


Figure 5.1. A Chain graph

We analysed the personalized PageRank equation for a chain graph in terms of the number of nodes, n, the teleportation probability, α, and the seed node, k. Given a graph with adjacency matrix, A and a diagonal degree matrix, D, the PageRank equations followed for this analysis is

$$
\begin{gathered}
(\alpha \mathbf{G}\mathbf{D}^{-1} + (1-\alpha)\mathbf{v_k}\mathbf{e^T})\mathbf{x} = \mathbf{x} \\
\text{Enforcing } \mathbf{e^T}\mathbf{x} = 1. \text{ Refer to [2] for more details} \\
(I - \alpha \mathbf{G}\mathbf{D}^{-1})x = (1-\alpha)\mathbf{v_k} \\
\rightarrow \alpha \mathbf{G}\mathbf{D}^{-1}\mathbf{x} = \mathbf{x} - (1-\alpha)\mathbf{v_k}
\end{gathered}
\tag{5.1}
$$

Here $\mathbf{v_k}$ is the indicator vector for seed k and $\mathbf{e}$ is the vector of all ones. The above relation gives n linear equation is $n+1$ variables (including α) as follows,

$$\begin{aligned}
\alpha/2x_2 &= x_1 \\
\alpha x_1 + \alpha/2x_3 &= x_2 \\
\alpha/2x_2 + \alpha/2x_4 &= x_3 \\
&\vdots \\
\alpha/2x_{k-2} + \alpha/2x_k &= x_{k-1} \\
\alpha/2x_{k-1} + \alpha/2x_{k+1} &= x_k - (1-\alpha) \\
\alpha/2x_k + \alpha/2x_{k+1} &= x_{k+1} \\
&\vdots \\
\alpha/2x_{n-3} + \alpha/2x_{n-1} &= x_{n-2} \\
\alpha/2x_{n-2} + \alpha x_n &= x_{n-1} \\
\alpha/2x_{n-1} &= x_n
\end{aligned} \tag{5.2}$$

With x_k as the seed node, we made an attempt to represent all nodes before x_k in terms of x_1 and all nodes after x_k in terms of x_n. Notice, from the formulation in (5.2), that the first $k-1$ linear equations, and the last $n-k$ linear equations form two systems of 2nd order recurrence. Solving them as such, we have the following expressions. For brevity, $(+) = \frac{1+\sqrt{1-\alpha^2}}{\alpha}$ and $(-) = \frac{1-\sqrt{1-\alpha^2}}{\alpha}$, and 2 sets of constants (C_1, D_1) and (C_n, D_n).

$$x_i = \left\{ \begin{array}{ll} C_1(\frac{1+\sqrt{1-\alpha^2}}{\alpha})^i + D_1(\frac{1-\sqrt{1-\alpha^2}}{\alpha})^i & \text{if } i = 2, \cdots, k-1 \\ \frac{\alpha}{2}x_{k-1} + \frac{\alpha}{2}x_{k+1} + (1-\alpha) & \text{if } i = k \\ C_n(\frac{1+\sqrt{1-\alpha^2}}{\alpha})^{n-i+1} + D_n(\frac{1-\sqrt{1-\alpha^2}}{\alpha})^{n-i+1} & \text{if } i = k+1, \cdots, n-1 \end{array} \right\} \tag{5.3}$$

Comparing the above expression for x_2 and x_3 with the first two equations in the system in (5.2), notice that we have an ambiguity for the expression for x_1, which gives us the following relation,

$$\frac{C_1}{D_1} = \frac{1-\sqrt{1-\alpha^2}}{1+\sqrt{1-\alpha^2}} \tag{5.4}$$

Substituting this ratio in the first equation in (5.2), we get expressions for C_1 and D_1 in terms of x_1, as follows,

$$C_1 = \frac{\alpha x_1}{1+\sqrt{1+\alpha^2}} \quad ; \quad D_1 = \frac{\alpha x_1}{1-\sqrt{1+\alpha^2}} \tag{5.5}$$

A similar procedure for C_n and D_n gives the following expression

$$C_n = \frac{\alpha x_n}{1+\sqrt{1+\alpha^2}} \quad ; \quad D_n = \frac{\alpha x_n}{1-\sqrt{1+\alpha^2}} \tag{5.6}$$

Substituting for the definition for the constants, we have the following definition of x_i,

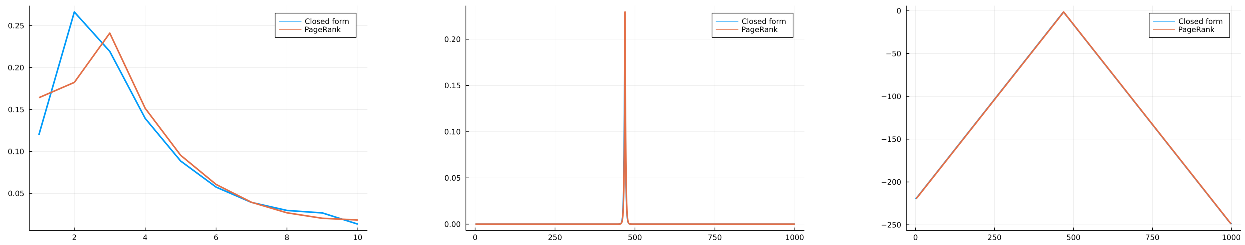
$$x_i = \left\{ \begin{array}{ll} x_1(+)^{i-1} + x_1(-)^{i-1} & \text{if } i = 2, \cdots, k-1 \\ \frac{\alpha}{2}[x_1(+)^{k-1} + x_1(-)^{k-1} + x_n(+)^{n-k-1} + x_n(-)^{n-k-1}] & \text{if } i = k \\ x_n(+)^{n-i} + x_n(-)^{n-i} & \text{if } i = k+1, \cdots, n-1 \end{array} \right\} \tag{5.7}$$

To find a value for x_1, x_n, we use the condition that $\sum_{i=1}^{n} x_i = 1$ which gives the following relation,

$$(x_1 + x_n)\left(1 + 2\frac{(+)^{k-2} - (+) - (+)^{n-k-1} - (+)}{(+)+1} + \frac{\alpha}{2}(+)^{k-2} + \frac{\alpha}{2}(+)^{n-k-1}\right) = \alpha \tag{5.8}$$

and that the value of x_k in terms of x_1 and x_n should be equal. That is,

$$\begin{aligned} &\frac{2}{\alpha}x_1((+)^{k-2} + (-)^{k-2}) - x_1((+)^{k-3} + (-)^{k-3}) \\ &= \frac{2}{\alpha}x_n((+)^{n-k-1} + (-)^{n-k-1}) - x_n((+)^{n-k-2} + (-)^{n-k-2}) \\ \rightarrow \frac{x_n}{x_1} &= \frac{(+)^{n-k} + (-)^{n-k}}{(+)^{k-1} + (-)^{k-1}} \end{aligned} \tag{5.9}$$



(a) Comparison of the PageRank values by the closed form expression on the 10 node chain graph

(b) Comparison of the PageRank values by the closed form expression on the 1000 node chain graph

(c) Comparison of the log of PageRank values by the closed form expression on the 1000 node chain graph

Figure 5.2. Closed form PageRank expression for Chain graph

The constraint in (5.8) simplifies to

$$x_1[(+)^{k-1} + (-)^{k-1}] + x_n[(+)^{n-k} + (-)^{n-k}] = \frac{2-2\alpha}{\sqrt{1-\alpha^2}} \tag{5.10}$$

Substituting the above in (5.9), we get the values of x_1 and x_n as follows,

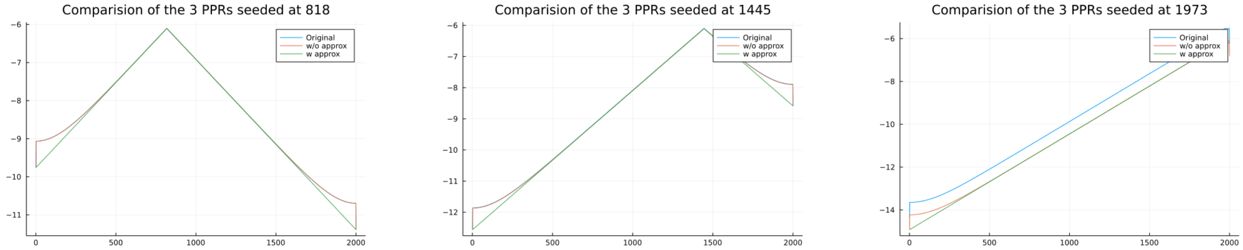
$$x_n = \frac{\sqrt{(1-\alpha)/(1+\alpha)}}{(+)^{n-k} + (-)^{n-k}} \quad ; \quad x_1 = \frac{\sqrt{(1-\alpha)/(1+\alpha)}}{(+)^{k-1} + (-)^{k-1}} \tag{5.11}$$

Thus, substituting this in (5.7), we have the following definition of PPR in terms of the seed node k, n, α

$$x_i = \left\{ \begin{array}{ll} \frac{\sqrt{(1-\alpha)/(1+\alpha)}}{(+)^{k-1}+(-)^{k-1}} \left((+)^{i-1} + (-)^{i-1}\right) & \text{if } i = 2, \cdots, k-1 \\ \frac{\alpha}{2}\sqrt{\frac{1-\alpha}{1+\alpha}} \left(\frac{(+)^{k-2}+(-)^{k-2}}{(+)^{k-1}+(-)^{k-1}} + \frac{(+)^{n-k-1}+(-)^{n-k-1}}{(+)^{n-k}+(-)^{n-k}}\right) & \text{if } i = k \\ \frac{\sqrt{(1-\alpha)/(1+\alpha)}}{(+)^{n-k}+(-)^{n-k}} \left((+)^{n-i} + (-)^{n-i}\right) & \text{if } i = k+1, \cdots, n-1 \end{array} \right\} \tag{5.12}$$

The following figures verify the accuracy of this closed form expression.

Notice the discontinuity in the expression for x_1, x_n and x_k. We found that, due to the low magnitude of $(-)$ and high values of the powers, considering only $(+)$ term, reduces the expression to have an intuitive format, in terms of distance from the seed node, k.



(a) PageRank for a 2000 node graph seeded at 818

(b) PageRank for a 2000 node graph seeded at 1445

(c) PageRank for a 2000 node graph seeded at 1973

Figure 5.3. Verification of approximate log of PageRank on chain graph

$$x_i = \sqrt{\frac{1-\alpha}{1+\alpha}}(+)^{-|i-k|} \quad ; \quad x_k = \sqrt{\frac{1-\alpha}{1+\alpha}}\frac{\alpha}{(+)} \tag{5.13}$$

for $i = \{1, \cdots n\} - k$. Then the logarithm of PageRank expression for x_i can be written as,

$$\log x_i = -|k-i|\log((+)) + \log(\sqrt{\frac{1-\alpha}{1+\alpha}}) \tag{5.14}$$

The following figures(5.3a) compare the PageRank value calculated for a chain graph, with 2000 nodes and teleportation probability, $\alpha = 0.99999$, using (i) the original method of calculating $(\mathbf{I} - \alpha\mathbf{G}\mathbf{D}^{-1})^{-1}$, (ii) the closed form expression (iii) the approximation of neglecting the factor $\frac{1-\sqrt{1-\alpha^2}}{\alpha}$. The figures after that (5.4) compare the embedding generated by the three procedures.

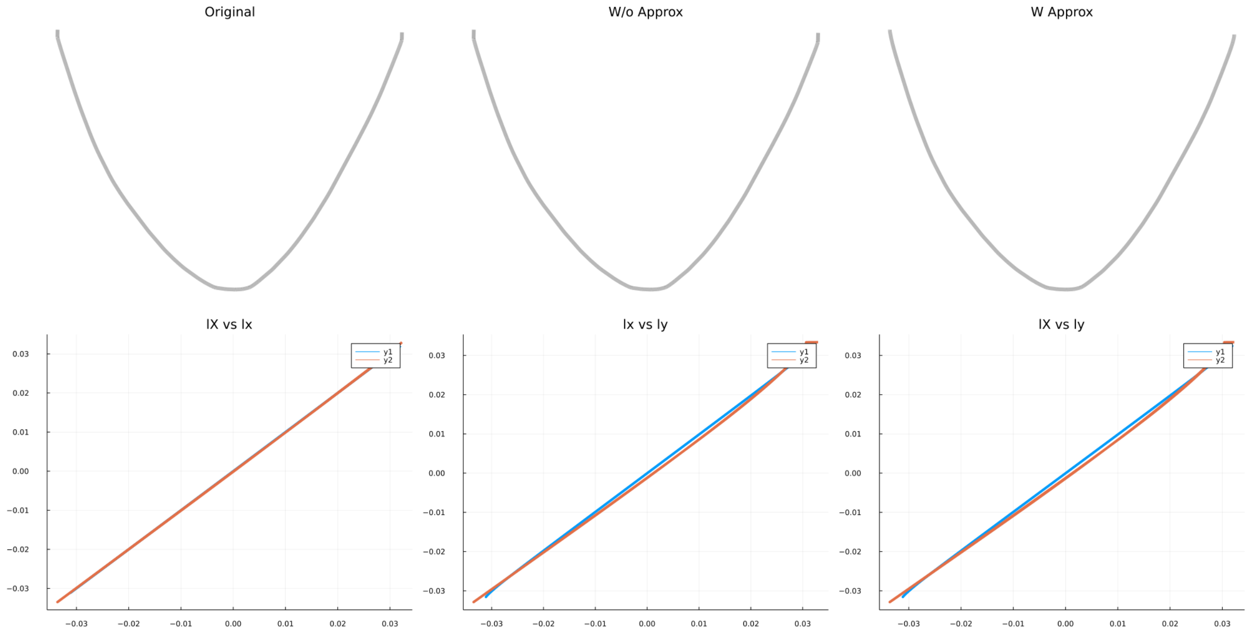


Figure 5.4. Embeddings generated by PageRank formulation and the closed form expression

To quantify the error, like 4.3, we use Rayleigh Coefficient with the second singular vector of the log PageRank matrix.

$$r = \mathbf{u^T G u / u^T u} \tag{5.15}$$

The error can then be defined as

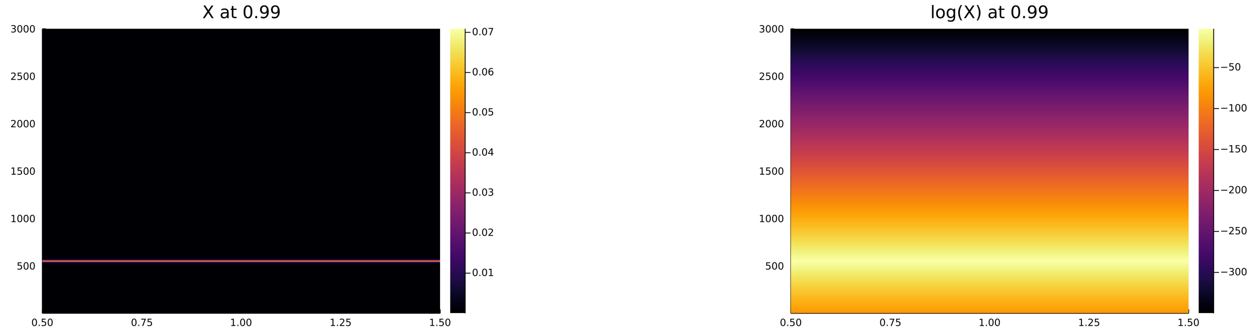
$$abs(r_1 - r_2)/r_2 \tag{5.16}$$

$r_i, i = 1, 2, 3$ refers to each of the 3 different procedures used to calculate the PPR value. The error between the original embedding and that developed by (5.12), that is r_{12}, is of the order -8 whereas the error between original embedding and (5.13), r_{13} and between (5.12) and (5.13), r_{23} is of the order -5.

5.1 Distance from the source matters

This section, using the formulations in the previous section, shows that as in any other diffusion process, distance from the source (the individual seed node in this case) matters.

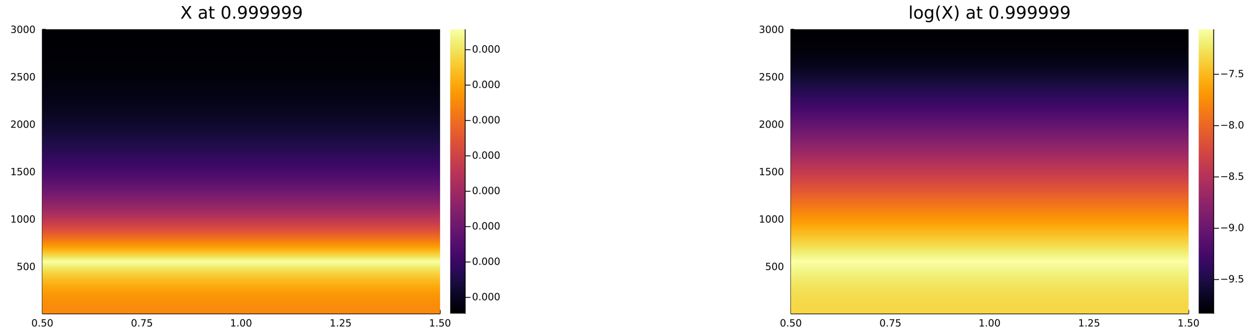
The formulation above was extremely helpful in validating the observation noted in section (1.1). The first notable fact is the decay of PageRank values with distance from the



(a) Heatmap of PageRank values at $\alpha =$ 0.99 for 3000 node chain graph

(b) Heatmap of log of PageRank values at $\alpha = 0.99$ for 3000 node chain graph

Figure 5.5. Distance effect in PageRank vs log of PageRank for chain graph for $\alpha = 0.99$



(a) Heatmap of PageRank values at $\alpha =$ 0.999999 for 3000 node chain graph

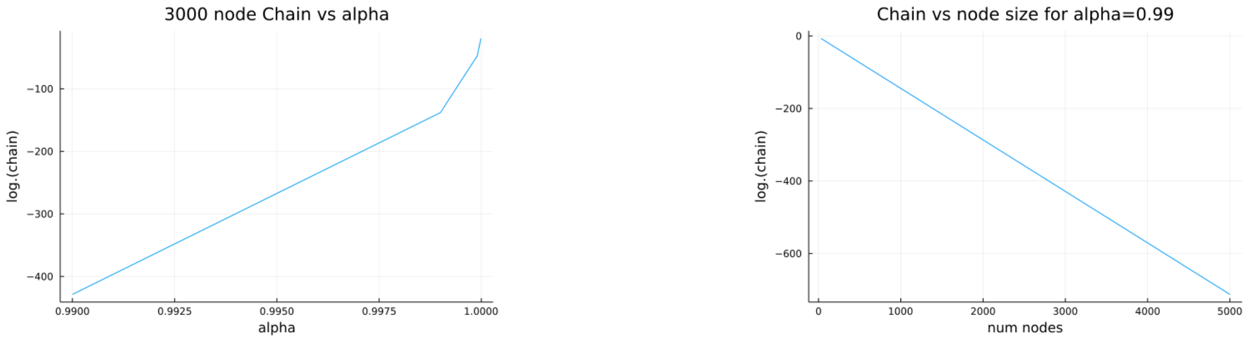
(b) Heatmap of log PageRank values at $\alpha = 0.999999$ for 3000 node chain graph

Figure 5.6. Distance effect in PageRank vs log of PageRank for chain graph for $\alpha = 0.999999$h

seed node which was also observed in figures 1.1a-1.2b. The same is justified by the approximation in (5.13) as the value is directly proportional to a constant raised to the negative power of $|i - k|$. The following heatmaps exemplify the same.

Numerically, we observed that for a 30 node chain graph, for an $\alpha = 0.99$, the smallest element in $\mathbf{X}$ had a power of -3, while for a 3000 node chain graph, for $\alpha = 0.99$, the lowest power in $\mathbf{X}$ was -146, which increases to -5 with $\alpha = 0.999999$.

This behavior is explained by the PageRank formulation for chain in (5.13) as for a larger chain the term $|i - k|$ would grow, therefore, for the same α, negative power in PageRank is larger in a large chain. But when α is increased, with the same power, the negative powers in PageRank reduces. The two plots below validate the same. The left one is PageRank



(a) The order of value on y-axis for $\alpha = 0.99$ was -187 as was observed in the PageRank as well

(b) The order of values on the y-axis for a size of 3000 was -187 which agrees with the values observed in PageRank

Figure 5.7. The numerical effect of α on the PageRank values for the Chain graph

element in the chain, $\mathbf{x}$ (from 5.13) vs α with constant power $-|i-k| = 3000$ and the right one is $\mathbf{x}$ vs the number of nodes (as an indication of how large $-|i-k|$ can get) for constant $\alpha = 0.99$. This also explains why using log becomes necessary for larger graphs with lower α.

We have the following inferences from our experiments on the chain graph.

1. *Larger α for larger graph size* - We observed that the term $|i-k|$ which refers to the distance of the current node, i, from the seed node, k, grows with the size of the chain. This in turn raises the factor $(+) = \frac{1+\sqrt{1-\alpha^2}}{\alpha}$ to a large negative power. This factor is smaller for larger α; hence, when raised to large negative power, the factor for larger α will be larger. Hence with increasing α, the PageRank values are more pronounced than the ones with smaller α.

2. *About* log - We observed that the element-wise log function was only necessary with (1) the lower values of α, and (2) large chain graph. In other words, for a small chain, smaller values of α at 0.99 are sufficient to generate spectral like embeddings; for larger chains, PageRank developed using small values of α need to use the element wise log operation, but, again, for larger graph, the PageRank generated using larger value of α do not need the element wise log operation. The reason for this is obvious after the previous observation. Large graphs imply large negative power with $-|i-k|$. In that

case small α would lead to smaller PageRank values, compared to bigger α. But using the element-wise logarithm operation, as can be seen from the second equation of the chain graph, the term $(+)$ is no longer raised to the power and hence becomes large enough to develop pictures.

5.2 Extension to planar graphs

The crucial takeaway from the formulation in (5.13) and the above discussion is that the element-wise logarithm is needed to boost the PageRank values when the distance between the seed node and the individual nodes are large, as is in large graphs, with lower α. However, that does not imply anything about its importance for other planar graphs. The next question that needs answered is how does the significance of the teleportation probability and the logarithm operation translate to the generic planar graphs that have been studied other than the chain graphs? Looking at a simple case like chain graph, we know that the PageRank formulation is given by

$$\begin{aligned} x_i &= \sqrt{\frac{1-\alpha}{1+\alpha}}(+)^{-|i-k|} \\ \log x_i &= -|i-k|\log((+)) + \log(\sqrt{\frac{1-\alpha}{1+\alpha}}) \end{aligned} \tag{5.17}$$

where $(+) = \frac{1+\sqrt{1-\alpha^2}}{\alpha}$, and hence the large distance from the seed node accounts for the large negative power which further necessitates using element-wise logarithm on PageRank values. However, we don't have such formulation for a general planar graph which justifies the above observation.

Interestingly, for other planar graphs as well, we noticed the increase in negative powers of PageRank with graph size for low α. For a 30 nodes nearest neighbour graph, with a teleportation probability of $\alpha = 0.99$, the lowest power in $\mathbf{X}$ was -2, but when the same $\alpha = 0.99$ was used on a 10000 node nearest neighbour graph, the lowest power in $\mathbf{X}$ became -18. In all cases, that is, for the chain graph, the nearest neighbour graphs and other planar graphs tried during the experiments, for α far lower compared to the order of the second eigen value, the powers in $\mathbf{X}$ were very low; it increased only when α was close to the order of $1 - \lambda_2$.

The element-wise expansion of PageRank in terms of the eigenvector of the Laplacian, $\mathbf{Z}$, (4.13), along with the matrix powers of the transition probability matrix, $\mathbf{P}$, offer a hint. Recall (4.13) and (4.12),

$$\begin{gathered}\log(X_{i,j}) = \log\left(\sum_{t=1}^{n} \frac{1-\alpha}{1-\alpha\epsilon_t} z_{ti} z_{tj} d_i\right) \\ \mathbf{X} = \mathbf{DZMZ^T}\end{gathered} \tag{5.18}$$

where $\mathbf{M}_t = \frac{1-\alpha}{1-\alpha\epsilon_t}$ and $\epsilon_t = 1 - \lambda_t$

Although we don't have a chain graph like formulation, as we can see from (4.13), these powers must be accounted for by the terms in $\mathbf{M}$. We begin by verifying the increase in powers caused by the denominator in $\mathbf{M}_t = \frac{1}{1-\alpha\epsilon_t}$. The following plots show the variation in $\frac{1}{1-\alpha\epsilon_t}$ vs α for different ϵ_2. A higher ϵ_2 implies a larger graph. The plot shows that for a large graph, as $\alpha \to 1$, the value $\frac{1}{1-\alpha\epsilon_t}$ increases but it does not sufficiently account for the increment in power.

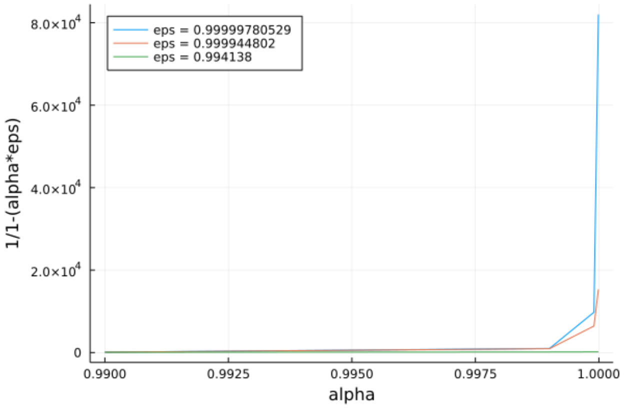
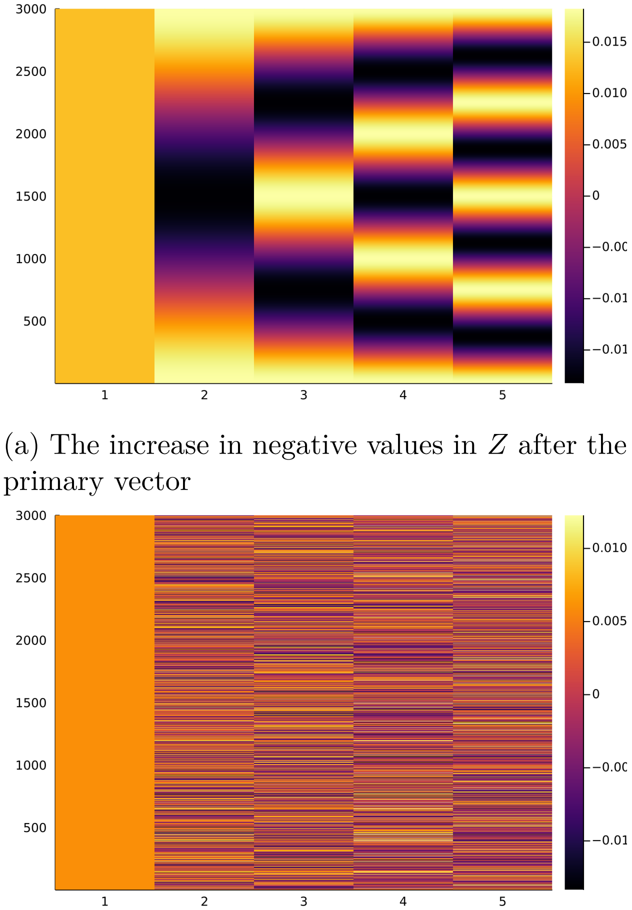


Figure 5.8. The increase in powers of PageRank because of $m = \frac{1}{1-\alpha\epsilon_t}$ where higher ϵ means larger graph

In the above plot, we did not consider the numerator in the terms in $\mathbf{M}$. The terms in $\mathbf{M}$ can be expressed as

$$M_{i,i} = \frac{1-\alpha}{1-\alpha\epsilon_i} \tag{5.19}$$



(a) The increase in negative values in Z after the primary vector

(b) The negative values in Z for 3000 node graphs with 9 nearest neighbours

Figure 5.9. Cumulative effect of the negative values in the eigenvector

Considering the total expression of $\mathbf{M}_i$, elements in the matrix $\mathbf{M}$ for the 3000 node chain graph with a teleportation probability of $\alpha = 0.99$ and $\alpha = 0.999999$ change as follows.

$$\mathbf{M}(\alpha = 0.99) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.9998 & 0 & 0 & 0 \\ 0 & 0 & 0.9991 & 0 & 0 \\ 0 & 0 & 0 & 0.998 & 0 \\ 0 & 0 & 0 & 0 & 0.996 \end{bmatrix} \tag{5.20}$$

$$\mathbf{M}(\alpha = 0.999999) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.313 & 0 & 0 & 0 \\ 0 & 0 & 0.102 & 0 & 0 \\ 0 & 0 & 0 & 0.048 & 0 \\ 0 & 0 & 0 & 0 & 0.027 \end{bmatrix} \tag{5.21}$$

Notice the negative values in all but the primary eigenvector in figures 5.9a and 5.9b. For $\alpha = 0.99$, all these vectors add up in almost unit ratio and hence causing large negative powers in PageRank; whereas, for $\alpha = 0.999999$, the eigenvectors corresponding to the larger eigenvalues combine in very small proportion which increases the value of the sum, and hence increasing the value of each PageRank element. The structure of $\mathbf{M}$ can be formalized by the following lemma.

Lemma 5.2.1. *As $\alpha \to 1$, the terms in $\mathbf{M}$ begin to concentrate on the first element as $\mathbf{M}_i \to 0$ for $i \neq 1$.*

This reasoning applies to the planar graphs as well because of a similar trend in the values of $\mathbf{M}$ and the eigenvectors $\mathbf{Z}$ shown in figure 5.9b. For the 3000 node nearest neighbour graph with 9 nearest neighbors,

$$\mathbf{M}(\alpha = 0.99999) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.0095 & 0 & 0 & 0 \\ 0 & 0 & 0.0093 & 0 & 0 \\ 0 & 0 & 0 & 0.004 & 0 \\ 0 & 0 & 0 & 0 & 0.002 \end{bmatrix} \tag{5.22}$$

$$\mathbf{M}(\alpha = 0.9) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.9907 & 0 & 0 & 0 \\ 0 & 0 & 0.9905 & 0 & 0 \\ 0 & 0 & 0 & 0.98 & 0 \\ 0 & 0 & 0 & 0 & 0.96 \end{bmatrix} \tag{5.23}$$

This was further validated by approximating PageRank using the formulation $\mathbf{X} = \mathbf{DZMZ^T}$. Since, we discussed that, this formulation works only for large α, we compared the values of $\mathbf{X}$ generated by it for the 3000 node chain graph. The minimum value in the approximated PageRank for $\alpha = 0.99999$ was of the order e^{-10} while for $\alpha = 0.999999$, it was e^{-5}. Along the same reasons, for the 3000 node nearest neighbour graph, the minimum value in the approximated PageRank for $\alpha = 0.99999$ was of the order e^{-5} while for $\alpha = 0.999$, it was e^{-10}.

This observation justifies the equivalence at higher teleportation probabilities and the need for element-wise logarithms at lower teleportation probabilities. We strengthen this argument through another perspective - the power series representation of the PageRank that involves matrix powers of the probability transition matrix, $\mathbf{GD^{-1}}$ along with their contribution in the PageRank value determined by powers of the teleportation probability, α. Recall the progression of the values in symmetric version of the transition matrix in figures 1.2a and 1.2b, combined with the powers of α in PageRank, that is, $\mathbf{X} = (1 - \alpha)\sum_{r=1}^{\infty}(\alpha)^r(\mathbf{GD}^{-1})^r$ hint at similar powers observed for the chain graph. The table shows an example of the order of the values from $(\mathbf{GD}^{-1})^r$ for a node in the 3000 node chain graph and that in the 30 node chain graph, both seeded at node 5 to facilitate comparison for $r = 100$ and $r = 1000$. We compare the order of the value for the node that has smallest non-zero value at $r = 100$.

Table 5.1. Matrix powers of $\mathbf{GD}^{-1}$

Graph	r	node	$\mathbf{GD}^{-1}$
30-chain	100	13	e^{-2}
3000-chain	100	106	e^{-30}
3000-chain	1000	106	e^{-3}
30-6 nearest neighbour	100	13	e^{-1}
3000-6 nearest neighbour	100	1792	e^{-20}
3000-6 nearest neighbour	1000	1792	e^{-4}

The powers in Table 5.1 combined with the respective powers of α, that is, $(\alpha = 0.9)^{1000} = 1.74e - 46$ and $(\alpha = 0.99999)^{1000} = 0.999$ are responsible for the powers in the PageRank matrix. Both the 30 node graphs (representative of small graphs) attain an order of e^{-2} for $r = 100$ as compared to the 3000 node graphs that attain that order at $r = 1000$. When these matrix powers add up in the PageRank with $\alpha = 0.9$, the higher matrix powers in larger graphs, where the values are significant, are made negligible for powers of $\alpha = 0.9$ but not for powers for $\alpha = 0.99999$. A larger graph needs more powers of the transition matrix $\mathbf{P}$ to contribute to PageRank to be able to create a low dimensional representation of the structure, and the contribution of these higher powers become significant in PageRank only for higher values of α. With the element-wise logarithm operation, although we do not have a closed form expression for matrix powers, we understand that teleportation probabilities stay significant for higher powers even with low α, that is $(\alpha = 0.99)^{1000} = 1.74e - 46$ but $\log(\alpha = 0.99)^{1000} = 1000 \log(0.99) = -10.05$. Therefore, the PageRank embedding technique with lower teleportation probabilities still work for larger graphs with the element-wise logarithm operation. On a graph, these matrix powers translate to steps of random walk. Larger negative powers in elements for increasing matrix powers imply that the values reduce as one goes farther from the seed node. Accordingly, since for nodes too far from the seed node, this value would be minuscule, the element-wise logarithm makes those values pronounced enough so that the nodes far from the seed node do not become non-trivial. According to the random surfer model, a larger teleportation probability would mean that the surfer continues with their random walk with a larger probability than getting bored and jumping to the nodes chosen in $\mathbf{v}_k$. Both of the above interpretations of PageRank, in terms of eigenvectors of Laplacian and power series of transition probability matrix, account for the reduction in negative powers with increasing α and the effect of element-wise log on PageRank.

The following questions still remain to answer.

1. The element-wise logarithm and normalization by square root of degree, seem to drive the PageRank embeddings towards spectral embeddings at low α for larger graphs. A similar effect was observed for embedding with the eigenvectors of the Laplacian, $\mathbf{Q}$ (this is also the reason why we instead use $\mathbf{Z} = \mathbf{D}^{-1/2}\mathbf{Q}$). With our previous explanations, it is not apparent as to why the element-wise logarithm operation should behave similar to degree normalization.

2. Why does this resemblance only work for a certain configuration of eigenvalues? In other words, why does this embedding technique not work on the planted partition models studies earlier?

6. PAGERANK APPROACHES STATIONARY DISTRIBUTION

One of the important question that this work tries to address is the importance of the element-wise logarithm operation. In previous sections, we concluded that the element-wise logarithm was necessary for $\alpha << \lambda_2$ in large graphs, because of the order of PageRank values which showed an inverse power dependence on the distance from seed node. In the course of our experiments with PageRank, we looked into the role that the log function had to play for larger graphs with smaller teleportation probabilities. We investigated this through the power series representation.

There are two power series involved in this formulation - the power series representation of log, and the Neumann series representation of PageRank. While our experiments with the first one did not lead to any significant observation, we found that the second perspective is being exercised by the perceptron learning models that work by minimizing the log of error. This section discusses the same.

6.1 The information theoretic perspective

SkipGram is a popular embedding technique that the analysis in [19], [20] and [26] inspired us to investigate that works with element-wise logarithm of PageRank from the perspective of the Markov matrix, $\mathbf{P} = \mathbf{G}\mathbf{D}^{\mathbf{-1}}$.

The minimal essential knowledge required to understand SkipGram is that it treats each word in the given test as a node, $\mathbf{w}$, and the other words that are semantically associated with it, as context, $\mathbf{c}$, and creates a dictionary, $\boldsymbol{\mathcal{D}}$, by sampling words from the given text. Conventionally, SkipGram proceeds by minimizing the following,

$$\log \sigma(\mathbf{wc}) \tag{6.1}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ and $\mathbf{w}$ and $\mathbf{c}$ are vector representations of those words. It was pointed out in [19] that according to this definition of the objective function, the matrix that is being factorized by the technique can be written in a closed form as,

$$\begin{aligned} SG_{i,j} &= \log(\frac{\#(w,c)|D|}{\#(w),\#(c)}) \\ &= \log(\frac{vol(G)}{T}(\sum_{r=1}^{T} P^r D^{-1})) \end{aligned} \tag{6.2}$$

where $\#(w,c)$ represents the number of times the two words appear together in $\boldsymbol{\mathcal{D}}$ and $\#w$ and $\#c$ is the number of times each of those words appear in the dictionary. The procedure calls T as the window length but we will not go into the details of language processing techniques here, and will just stick to calling it matrix powers. Hence, SkipGram is essentially factorizing a matrix resulting from the power series of the probability transition matrix on the graph developed from $\boldsymbol{\mathcal{D}}$. As is well knows, PageRank can also be represented as,

$$\begin{aligned} \mathbf{X} &= (1-\alpha)(I-\alpha\mathbf{P})^{-1} \\ &= (1-\alpha)\sum_{r=0}^{\infty} \alpha^r \mathbf{P}^r \\ &= (1-\alpha)\sum_{r=0}^{\infty} \alpha^r (\mathbf{G}\mathbf{D}^{-1})^r \end{aligned} \tag{6.3}$$

The similarity in the above two formulations enables us to derive an expression of the personalized PageRank matrix in the limit of $\alpha \to 1$. As before, we drop the $\mathbf{v}$ from further analysis assuming all the nodes are sampled once.

Recall here that according to our formulation, $\mathbf{P}$ is column stochastic and $\mathbf{P}\boldsymbol{\pi} = \boldsymbol{\pi}$. Expanding $\mathbf{P}$ in terms of its eigenvectors

$$\begin{aligned}
\mathbf{P} &= \sum_{i=1}^{n} \lambda_i \mathbf{u_i} \mathbf{v_i}^T \\
&= \lambda_1 \mathbf{u_1} \mathbf{v_1}^T + \sum_{i=2}^{n} \lambda_i \mathbf{u_i} \mathbf{v_i}^T \\
&= \pi \mathbf{e^T} + \sum_{i=2}^{n} \lambda_i \mathbf{u_i} \mathbf{v_i}^T \\
\rightarrow \mathbf{P}^r &= \sum_{i=1}^{n} \lambda_i^r \mathbf{u_i} \mathbf{v_i}^T \\
&= \pi \mathbf{e}^T + \sum_{i=2}^{n} \lambda_i^r \mathbf{u_i} \mathbf{v_i}^T
\end{aligned} \tag{6.4}$$

Using the Neumann series expansion of PageRank and the above formulation of $\mathbf{P}^r$,

$$\begin{aligned}
\mathbf{X}(\alpha) &= (1-\alpha) \sum_{r=0}^{\infty} (\alpha \mathbf{P})^r \\
&= (1-\alpha) \sum_{r=0}^{\infty} \alpha^r \left(\boldsymbol{\pi} e^T + \sum_{i=2}^{n} \lambda_i^r \mathbf{u_i} \mathbf{v_i}^T \right) \\
&= (1-\alpha) \left(\boldsymbol{\pi} e^T \sum_{r=0}^{\infty} \alpha^r + \sum_{i=2}^{n} \sum_{r=0}^{\infty} (\alpha \lambda_i)^r \mathbf{u_i} \mathbf{v_i}^T \right) \\
&= (1-\alpha) \left(\frac{\boldsymbol{\pi} \mathbf{e}^T}{1-\alpha} + \sum_{i=2}^{n} \frac{1}{1-\alpha\lambda_i} \mathbf{u_i} \mathbf{v_i}^T \right) \\
&= \boldsymbol{\pi} e^T + (1-\alpha) \sum_{i=2}^{n} \frac{1}{1-\alpha\lambda_i} \mathbf{u_i} \mathbf{v_i}^T \\
\mathbf{D^{-1}} \mathbf{X}(\alpha) &= \mathbf{D}^{-1} \left(\boldsymbol{\pi} \mathbf{e}^T + (1-\alpha) \sum_{i=2}^{n} \frac{1}{1-\alpha\lambda_i} \mathbf{u_i} \mathbf{v_i}^T \right)
\end{aligned} \tag{6.5}$$

The LHS of the above equation will have the rows scaled by the corresponding degree. An element-wise logarithm on the matrix will expand as follows.

$$\log.(\mathbf{D^{-1}X}(\alpha)) = \log. \begin{bmatrix} \frac{x_{11}}{d_1} & \frac{x_{21}}{d_1} & \cdots \frac{x_{n1}}{d_1} \\ \frac{x_{12}}{d_2} & \frac{x_{22}}{d_2} & \cdots \frac{x_{n2}}{d_2} \\ \vdots & & \\ \frac{x_{1n}}{d_n} & \frac{x_{2n}}{d_2} & \cdots \frac{x_{nn}}{d_n} \end{bmatrix} = \begin{bmatrix} \log x_{11} \cdots \log x_{n1} \\ \vdots \\ \log x_{1n} \cdots \log x_{nn} \end{bmatrix} - \begin{bmatrix} \log d_1 \cdots \log d_1 \\ \vdots \\ \log d_n \cdots \log d_n \end{bmatrix}$$

$$\log.(\mathbf{D^{-1}X}(\alpha)) = \log \mathbf{X}(\alpha) - \log(\mathbf{\Pi}) \tag{6.6}$$

For the RHS of the equation, we use the approximation of logarithm that

$$\lim_{x \to 0} \log(1+x) \to x$$

. The formulation can finally be written as

$$\begin{aligned} \log \mathbf{X}(\alpha) - \log(\mathbf{\Pi}) &= \log\left(\mathbf{D}^{-1}\boldsymbol{\pi}\mathbf{e}^T + \mathbf{D}^{-1}\sum_{i=2}^{n} \frac{1-\alpha}{1-\alpha\lambda_i}\mathbf{u_i}\mathbf{v_i}^T\right) \\ &= \log\left(\mathbf{J} + (1-\alpha)\mathbf{D}^{-1}\sum_{i=2}^{n} \frac{1}{1-\alpha\lambda_i}\mathbf{u_i}\mathbf{v_i}^T\right) \end{aligned} \tag{6.7}$$

Using the fact that this technique only works for planar graphs where the eigenvalues of the symmetric Laplacian follow $\lambda_2 < O(1/n)$, we argue that for the first few eigenvalues, we can say that $\epsilon \to 1$, where $\epsilon = 1 - \lambda$. Since the trajectory followed by the teleportation probability is similar, that is $\alpha \to 1$, we further say that, $\alpha \to \epsilon$. In a limit we denote this as $\lim_{\alpha,\epsilon \to 1}$. Thus, the RHS in the above formulation can be calculated in the limit for planar graphs, as follows,

$$\begin{aligned} \lim_{\alpha \to 1}(1-\alpha)\sum_{i=2}^{n} \frac{1}{1-\alpha(1-\lambda_i)} &= \lim_{\alpha,\epsilon \to 1}\left((1-\alpha)\sum_{i=2}^{5} \frac{1}{1-\alpha(1-\lambda_i)}\right) + \lim_{\alpha \to 1}(1-\alpha)\sum_{i=6}^{n} \frac{1}{1-\alpha\epsilon_i} \\ &= \frac{1}{2} + 0 \end{aligned} \tag{6.8}$$

Therefore, using the above LHS and RHS approximations, the logarithm of PageRank can be written as,

$$\begin{aligned}\lim_{\alpha\to 1}\log \mathbf{X}(\alpha) - \log(\mathbf{\Pi}) &= \lim_{\alpha\to 1}\log\left(\mathbf{J} + (1-\alpha)\mathbf{D}^{-1}\sum_{i=2}^{n}\frac{1}{1-\alpha\lambda_i}\mathbf{u_i}\mathbf{v_i}^T\right)\\ \log \mathbf{X} &= \log\left(\mathbf{J} + \mathbf{D}^{-1}\left(\lim_{\alpha,(1-\lambda)\to\epsilon}\sum_{i=2}^{6}\frac{1-\alpha}{1-\alpha\lambda_i} + \sum_{i=7}^{n}\lim_{\alpha\to 1}\frac{1-\alpha}{1-\alpha\lambda_i}\right)\mathbf{u_i}\mathbf{v_i}^T\right) + \log(\mathbf{\Pi})\\ \log \mathbf{X} &= \log(\mathbf{J} + \frac{1}{2}\mathbf{D}^{-1}\sum_{i=2}^{6}\mathbf{u_i}\mathbf{v_i}^T) + \log(\mathbf{\Pi})\end{aligned} \tag{6.9}$$

According to 4.12, we have $\mathbf{P} = \mathbf{D^{1/2}Q\mathcal{E}Q^TD^{-1/2}}$. We denote $\mathbf{Z} = \mathbf{D^{-1/2}Q}$, primarily because we only have access to $\mathbf{Z}$ for experimentation. So $\mathbf{U} = \mathbf{DZ}$ and $\mathbf{V} = \mathbf{Z}$, which implies $\mathbf{V} = \mathbf{DU} \to \mathbf{v_i^T} = \mathbf{u_i^T D}$. So,

$$\begin{aligned}\log \mathbf{X} &= \log(\mathbf{J} + \frac{1}{2}\mathbf{D}^{-1}\sum_{i=2}^{6}\mathbf{u_i u_i^T D}) + \log(\mathbf{\Pi})\\ \log \mathbf{X} &\approx \frac{1}{2}\mathbf{D^{-1}UU^TD} + \log(\mathbf{\Pi})\\ \log \mathbf{X} &\approx \frac{1}{2}\mathbf{D^{-3/2}QQ^TD^{1/2}} + \log(\mathbf{\Pi})\end{aligned} \tag{6.10}$$

The above expression shows that for small values of $\mathbf{Q}$, that is for large graphs, we can approximate the element-wise logarithm of PageRank as,

$$\log \mathbf{X} = \frac{1}{2}\mathbf{D}^{-1}\sum_{i=2}^{6}\mathbf{u_i}\mathbf{v_i}^T \tag{6.11}$$

A different representation of the limiting PageRank in terms of the teleportation probability was discussed by [15]. Starting with a Jordan canonical representation, $\mathbf{P} = \mathbf{XJX}^{-1}$, where

$$\mathbf{J} = \begin{bmatrix}\mathbf{I} & \\ & \mathbf{J_1}\end{bmatrix}$$

where $\mathbf{J_1}$ is the matrix of Jordan blocks for all $|\lambda_i| < 1$. A curious reader might notice we don't consider $\mathbf{D_1}$ as in [15] because the geometric and algebraic multiplicity of $\lambda_1 = 1$ is always 1 in our case. Thus, the personalized PageRank equation in terms of the Jordan representation can be written as

$$\begin{aligned}
\mathbf{X}(\mathbf{I} - \alpha\mathbf{P})\mathbf{X^{-1}} &= (1-\alpha)\mathbf{v} \\
(\mathbf{I} - \alpha\mathbf{P})\mathbf{X}^{-1}\mathbf{x} &= (1-\alpha)\mathbf{X^{-1}}\mathbf{v} \\
\left(\mathbf{I} - \begin{bmatrix} \mathbf{I} & \\ & \mathbf{J_1} \end{bmatrix} \alpha \right)\mathbf{z} &= (1-\alpha)\mathbf{u} \qquad (6.12) \\
\rightarrow 1 - \alpha z_1 &= (1-\alpha)u_1 \\
\rightarrow (\mathbf{I} - \alpha\mathbf{J_2})\mathbf{z_2} &= (1-\alpha)\mathbf{u_2}
\end{aligned}$$

All values in $\mathbf{J_2}$ will be less than 1, so for the last equation to hold with $\alpha \rightarrow 1$, $\mathbf{z_2} \rightarrow 0$. So, we are left with $\mathbf{z_1} = \mathbf{u_1}$. Separating $\mathbf{X} = [\mathbf{X_1}, \mathbf{X_2}]$ and $\mathbf{X^{-1}} = [\mathbf{Y_1}, \mathbf{Y_2}]^{\mathbf{T}}$, according to the values in $\mathbf{J}$, we can write

$$\begin{aligned}
z_0 &= u_0 \\
\mathbf{x}(\alpha) &= \mathbf{X_0}\mathbf{Y_0}\mathbf{v} \qquad (6.13) \\
\lim_{\alpha\rightarrow 1} \mathbf{X}(\alpha) &= \mathbf{X_0}\mathbf{Y_0}
\end{aligned}$$

where the last equality is because we seed on individual node and assume each seed acts as a node exactly once. Therefore, we can conclude that

$$\lim_{\alpha\rightarrow 1} \mathbf{X}(\alpha) = \mathbf{D}\mathbf{z_1}\mathbf{z_1^T} \qquad (6.14)$$

where we use the eigen decomposition of $\mathbf{P}$ in terms of $\mathbf{Z}$. At the same time, using the properties of Markov matrices as in [15], we know that, by our definition of $\mathbf{P} = \mathbf{G}\mathbf{D}^{-1}$, the first right eigenvector will be the stationary distribution, $\boldsymbol{\pi}$, and the first left eigenvector would be], both with eigenvalues of 1. Hence, using 6.13, we have $\mathbf{X_0} = \boldsymbol{\pi} = \frac{\mathbf{Ge}}{\mathbf{e^T Ge}}$ and $\mathbf{Y_0} = \mathbf{e^T}$. Therefore, the limiting PageRank matrix can be expressed as,

$$\begin{aligned}\lim_{\alpha\to 1} X(\alpha) &= \frac{\mathbf{Gee}^T}{\mathbf{e^T Ge}} \\ &= \frac{\mathbf{GJ}}{vol_G} \\ \to \lim_{\alpha\to 1} \log(\mathbf{X}(\alpha)) &= \log(\boldsymbol{\pi} e^T)\end{aligned} \tag{6.15}$$

where log(.) is the element-wise logarithm operation and vol_G is the sum of degrees of all vertices in $\mathbf{G}$.

Comparing the representation in 6.15 with 6.10, and if we don't use the condition of planar graph on the eigenvalues, both formulation agree that,

$$\lim_{\alpha\to 1} \log \mathbf{X}(\alpha) = \log(\boldsymbol{\pi}\mathbf{e}^T) \tag{6.16}$$

We can compile the above observations in the following lemma.

Lemma 6.1.1. *The PageRank matrix, where each column denotes the personalized PageRank vector seeded on each individual node in the graph, approaches the stationary distribution matrix as $\alpha \to 1$. That is*

$$\lim_{\alpha\to 1} \log \mathbf{X}(\alpha) = \mathbf{D z_1 z_1^T} = \log(\boldsymbol{\pi}\mathbf{e}^T) \tag{6.17}$$

However, using the planar graph conditions allows us to make further simplifications leasing up to the formulation in (6.10).

7. THEORETICAL IMPLICATIONS

7.1 Approaching the eigenvectors

This section looks at the singular vectors of personalized PageRank and tries to answer how the singular vectors become similar to $\mathbf{Z}$ as $\alpha \to 1$.

In the previous section we concluded that

$$\mathbf{X} = \mathbf{DZMZ^T} = \mathbf{D}\sum_{t=1}^{n} \frac{1-\alpha}{1-\alpha\epsilon_t}\mathbf{z_t}\mathbf{z_t}^T \tag{7.1}$$

where $\mathbf{M} = (1-\alpha)(\mathbf{I} - \alpha\boldsymbol{\mathcal{E}})^{-1}$ and that for large graphs with $\alpha \to 1$, higher powers of $\mathbf{GD}^{-1}$ are included and the weights of the eigenvectors in $\mathbf{z}$ corresponding to larger eigenvalues of $\tilde{\mathbf{L}}$ are diminished. The same perspective also answers the question we are addressing in this section.

Using Lemma 5.2.1 and the above formulation of $\mathbf{X}$,

$$\lim_{\alpha \to 1} \mathbf{X} \to \mathbf{D}\mathbf{z_1}\mathbf{z_1}^T \tag{7.2}$$

As a corollary of Lemma 6.1.1, we have that

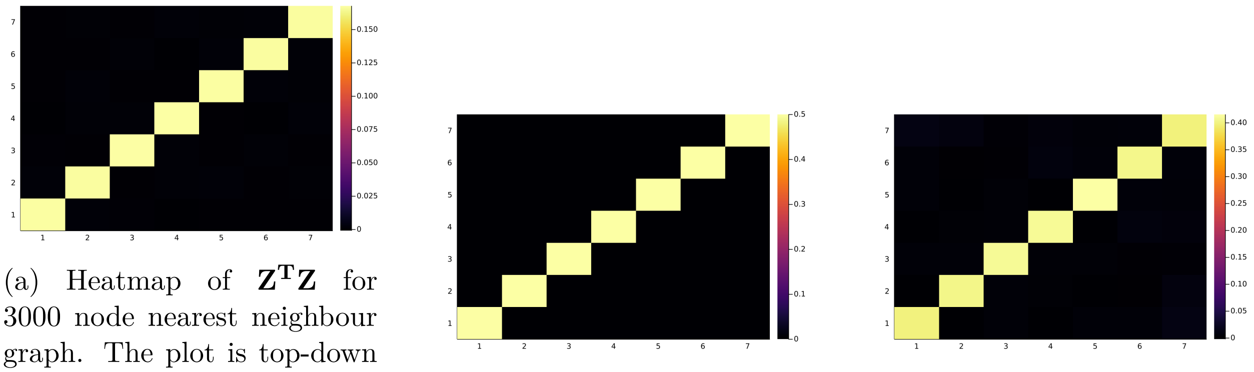
$$\mathbf{z_1} = \frac{1}{\sqrt{vol_G}}\mathbf{e} \tag{7.3}$$

This further implies

$$\begin{aligned} \lim_{\alpha \to 1} \mathbf{X} &= \frac{1}{vol_G}\mathbf{Dee}^T \\ &= \frac{1}{vol_G}\mathbf{de}^T \\ &= \mathbf{\Pi} \end{aligned} \tag{7.4}$$

where $\mathbf{\Pi}$ is the matrix with the stationary distribution of the graph $\mathbf{G}$ as its columns.

During experiments, we only sample a few columns of the above matrix depending on the seed node which renders a rectangular matrix, which necessitates the singular value decomposition operation. It is crucial to note that the power of this technique lies in the



(a) Heatmap of $\mathbf{Z^TZ}$ for 3000 node nearest neighbour graph. The plot is top-down inverted which means the bottom left is the $(0, 0)$ index.

(b) Heatmap of $\mathbf{Z^TZ}$ for 3000 node chain graph.

(c) Heatmap of $\mathbf{Z^TZ}$ for the Minnesota graph.

Figure 7.1. The pattern in $\mathbf{Z^TZ}$

fact that only a few columns of the PageRank matrix needs to sampled to create a spectral-embedding like representation of the graph localized around the seed nodes. We calculate PageRank with respect to each individual seed node, whose population is much less than the total graph. Had it been otherwise and one had to sample PageRank personalized on each node of the graph, the subsequent SVD operation would become expensive and the technique would lose its edge.

Having noticed the trend followed by the eigenvalues of the graphs that agree with the equivalence (4.2) and the elements in $\mathbf{M}$, we can write the expression for singular vectors used for embedding using the approximation of PageRank from ((4.12)) as,

$$\begin{aligned} \mathbf{X} &= \mathbf{DZMZ}^T \\ \rightarrow \mathbf{XX^T} &= \mathbf{DZMZ^TZMZ^TD} \end{aligned} \tag{7.5}$$

The above equation is critical because it shows how the singular values and the left singular vector of the PageRank might be related to α. Recall that for the 3000 node chain graph, for $\alpha = 0.99$, $\mathbf{M}$ is almost close to $\mathbf{I}$ (see 5.23) for the first few eigenvectors. Additionally, we observed that for $\mathbf{Z^TZ}$, the ratio of sum of diagonal elements to non-diagonal elements was more than 99%, as in figures(7.1a-7.1c). For analysis purpose, we treat $\mathbf{Z^TZ}$ as identity matrix, $\mathbf{I}$. No helpful structure was observed for $\mathbf{ZZ^T}$.

Hence, effectively, at $\alpha = 0.99$, for larger graphs ($n \sim 10^3$),

$$\mathbf{XX^T} = \mathbf{DZZ^TD} \tag{7.6}$$

However, for higher values of the teleportation probability, in large graphs the elements in $\mathbf{M}$ begin to concentrate on the first element of the diagonal of the matrix. For the 3000 node nearest neighbour graph with $\alpha = 0.999999$, the values are as follows.

$$\mathbf{M} = \begin{bmatrix} 1.0 & & & & \\ & 0.00096 & & & \\ & & 0.00094 & & \\ & & & 0.000449627 & \\ & & & & 0.000254531 \end{bmatrix} \tag{7.7}$$

Thus, for larger graphs and higher values of α, the information in the PageRank matrix begins to concentrate on the dominant eigenvector of $\tilde{\mathbf{L}}$ or equivalently the first right eigenvector of $\mathbf{P}$. Recall that during the process we do not have access to all the eigenvectors as we are only sampling a small fraction of PageRank columns. While approximating PageRank at high values of α, using small number (compared to the size of the graph) of eigenvectors is sufficient because the information is concentrated on the eigenvectors corresponding to the highest values of $\mathbf{M}$ and hence the lowest eigenvalues of the symmetric Laplacian.

So for $\alpha \to 1$, for larger graphs ($n \sim 10^3$),

$$\mathbf{XX^T} = \mathbf{DZMZ^TZMZ^TD} \approx \mathbf{DZe_1e_1^TZ^TD} \tag{7.8}$$

This further implies that when the PageRank matrix is developed using only a few columns of personalized PageRank vectors, the singular vectors will contain more information for larger values of α because most of the information will be concentrated in the first few eigenvectors. The observations agree with this theory as the error between actual PageRank matrix and the PageRank approximated using that formulation are decrease with increasing α for the same number of eigenvectors used. The exact errors are given in Table(7.1) for different graphs with different α. These reconstructions use only the top 5 to 7 eigenvectors (eigenvectors corresponding to the lowest 5 or 7 eigenvalues of the symmetric Laplacian) and the error has been measured in terms of 1-norm. This error is also theoretically justified by (4.15). To clarify, in this section we focus on the concentration of singular vectors of PageRank on $\mathbf{Z}$ and not on their low reconstruction error. The above formulation delineates the expression for the embedding vectors, that is, the singular vectors of PageRank.

This formulation, along with the trend of values in $\mathbf{M}$ with respect to the graph size and order of α (see 5.23 and 5.22) also explains the insight shared in [13] regarding the equivalence of the PageRank on the entire seedset being equal to the first left singular vector of the matrix generated by PageRank on individual seed. We noted in (1.1) that is equivalence was only true when α matched the order of Fiedler value of the respective graph. In the limit of $\alpha \to 1$, it can be justified as follows. We know from chapter 6 that $\lim_{\alpha\to 1} \mathbf{X} = \mathbf{\Pi}$ where $\mathbf{\Pi}$ has k columns of π. So in the limit that $\alpha \to 1$,

$$\begin{aligned} \lim_{\alpha\to 1}\left[\left((1-\alpha)\mathbf{v_1}\mathbf{e^T} + \alpha\mathbf{P}\right)\mathbf{x_1}\cdots\left((1-\alpha)\mathbf{v_k}\mathbf{e^T} + \alpha\mathbf{P}\right)\mathbf{x_k}\right] &= \lim_{\alpha\to 1}\left[\mathbf{x_1}\cdots\mathbf{x_k}\right] \\ \left[\mathbf{P}\pi\cdots\mathbf{P}\pi\right] &= \left[\boldsymbol{\pi}\cdots\boldsymbol{\pi}\right] \\ \text{Taking column average on both sides} \\ \left[\mathbf{P}\pi\cdots\mathbf{P}\pi\right]\mathbf{e} &= \left[\boldsymbol{\pi}\cdots\boldsymbol{\pi}\right]\mathbf{e} \end{aligned} \tag{7.9}$$

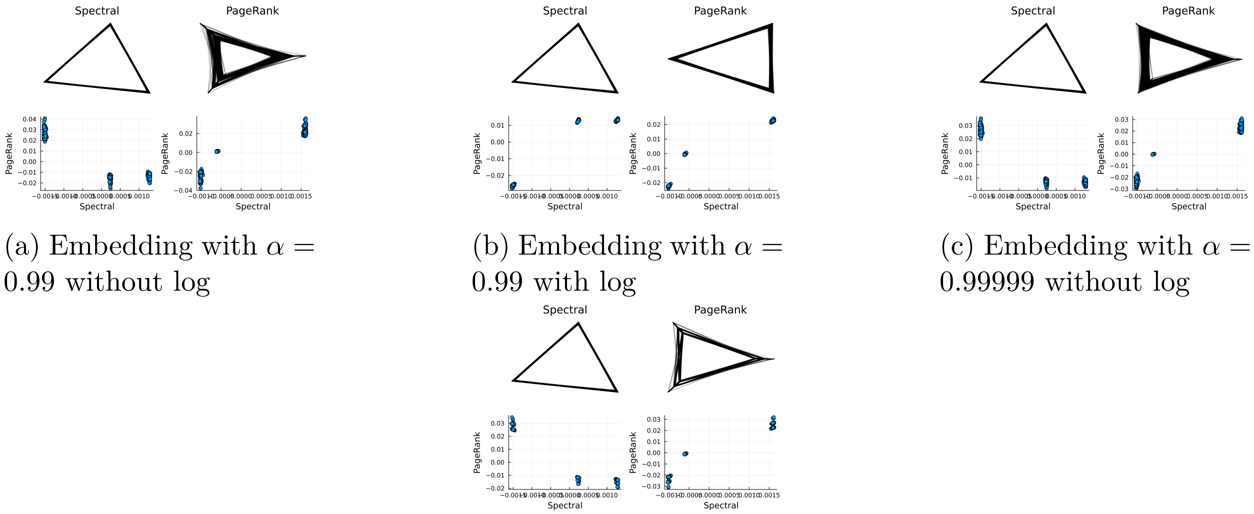
Now we use the fact that $\mathbf{z_1} = c\mathbf{e}$, where the constant c depends on the graph and hence, we can write $\boldsymbol{\pi} = \mathbf{Dz_1}$. Further, using $\mathbf{XX}^T = \mathbf{DZe_1e_1^TZ^TD}$ as an insight, we computationally verified that $\mathbf{u_1} = \mathbf{Dz_1}$.

$$\begin{aligned} \left[\mathbf{PDz_1}\cdots\mathbf{PDz_1}\right]\mathbf{e} &= \left[\mathbf{Dz_1}\cdots\mathbf{Dz_1}\right]\mathbf{e} \\ \left[\mathbf{Pu_1}\cdots\mathbf{Pu_1}\right]\mathbf{e} &= \left[\mathbf{u_1}\cdots\mathbf{u_1}\right]\mathbf{e} \end{aligned} \tag{7.10}$$

which proves that $\mathbf{u_1}$ can approximate the PageRank on the seedset.

Table 7.1. Error in PageRank approximation

Graph	α	error(%)
3000chain	0.9	192.82
3000chain	0.999999	5.8
30chain	0.99	12.64
3000 NN	0.9	169.21
3000 NN	0.99999	11.4
30 NN	0.99	14.5
Tapir-1024	0.9	158.7
Tapir-1024	0.9999	1.56
Original-5500	0.9	172.89
Original-5500	0.9999	8.1
Minnesota-2640	0.9	186.67
Minnesota-2640	0.9999	7.65



(a) Embedding with $\alpha =$ 0.99 without log

(b) Embedding with $\alpha =$ 0.99 with log

(c) Embedding with $\alpha =$ 0.99999 without log

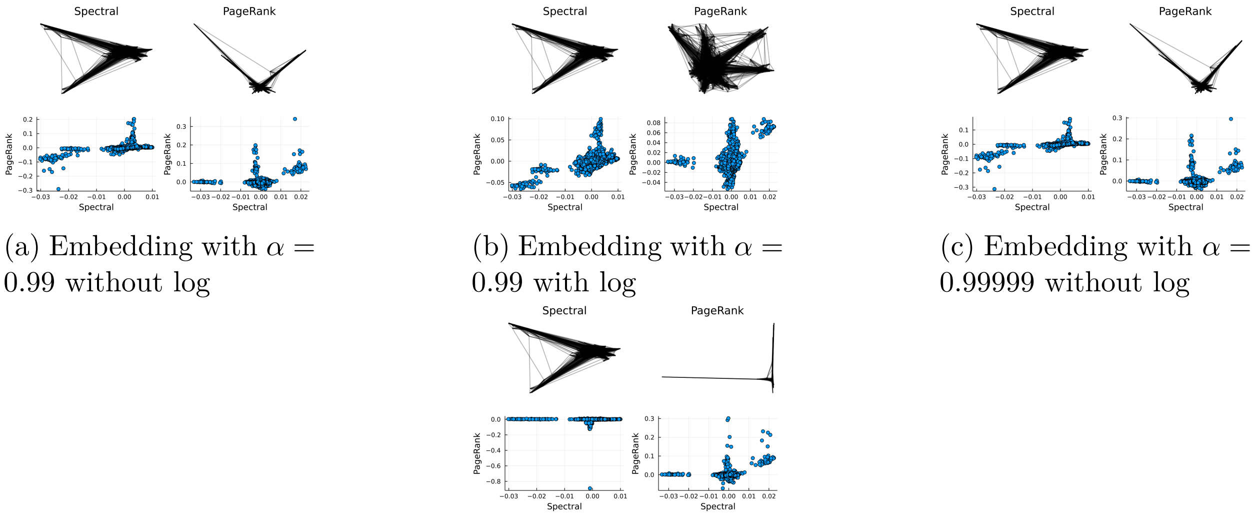
(d) Embedding with $\alpha =$ 0.99999 with log

Figure 7.2. PageRank embeddings for planted partition model with 3 blocks of 10000 nodes each with in block edge probability of 0.25 and out-block edge probability of 0.001

7.2 Discussion on Random Graphs

Through this section, we try to investigate the planted partition model and its characteristics that promote the resemblance between PageRank embeddings and eigenvector embeddings. In section 1.1, we saw that in Figures 1.8a, 1.8b and 1.9 the technique did not work for planted partition models. However, through more experiments, we noticed that it did work for models with $p >> q$, where p is in-block edge probability and q is out-block edge probability as can be seen from the Rayleigh quotient error in Table 4.1. We present the corresponding pictures here.

The similarity score for these graphs have been reported in Table 4.1. As expected from the figures, the overall error was much lower for planted partition model with 3 blocks of 1000 nodes each as compared to the one for 60 blocks of 50 nodes each. Further, the error for the former was also much lower than the planted partition model with the same node distribution but where the in-block edge probability was almost equal to the out-block edge probability, that is $p \sim q$.



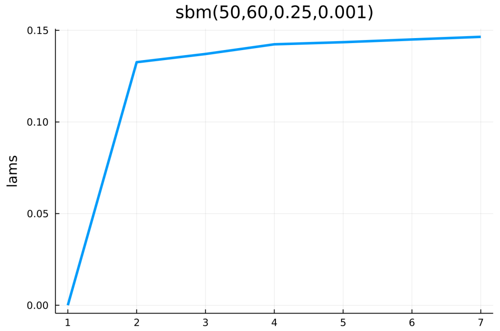
(a) Embedding with $\alpha =$ 0.99 without log

(b) Embedding with $\alpha =$ 0.99 with log

(c) Embedding with $\alpha =$ 0.99999 without log

(d) Embedding with $\alpha =$ 0.99999 with log

Figure 7.3. Embedding for planted partition model with 60 blocks of 50 nodes each with in block edge probability of 0.25 and out-block edge probability of 0.001



(a) Eigenvalues of the planted partition model with 60 blocks of 50 nodes each with in-block edge probability of 0.25 and out-block edge probability of 0.001

(b) Eigenvalues of the planted partition model with 3 blocks of 1000 nodes each with in-block edge probability of 0.25 and out-block edge probability of 0.001

Figure 7.4. Eigenvalue patterns in planted partition model

Recall from the earlier discussion on the significance of eigenvalues that for graphs where this embedding technique works, λ_d is of the order $\frac{1}{n}$. We also find the same trend of eigenvalues for these graphs as well as was observed in 4.2 for other graphs that show the resemblance. The value of $\lambda_d = |(\lambda_3 - \lambda_2) - (\lambda_2 - \lambda_1)|$ for these graphs are 0.1281 for $sbm(50, 60, 0.25, 0.001)$ and 0.011 for $sbm(1000, 3, 0.25, 0.001)$. As expected the graph for which the embedding technique did not work has a mugh higher 2nd order slope at λ_2.

8. HYPERGRAPHS

Hypergraphs are an higher-order extensions of graphs that are naturally better at modelling real-world data allowing for multi-way interactions between nodes. PageRank and other diffusion based techniques have been used for hypergraph clustering ([27],[28],[9]) and embedding ([12],[29],[11]) for learning on hypergraphs. In this section we attempt to extend the PageRank embedding technique discussed above to hypergraphs.

8.1 A different perspective

PageRank as a solution to an optimization problem was introduced in [17] and [10] developed a localized algorithm to compute such a solution. Although we were not successful in our current efforts, we identify problems for future work.

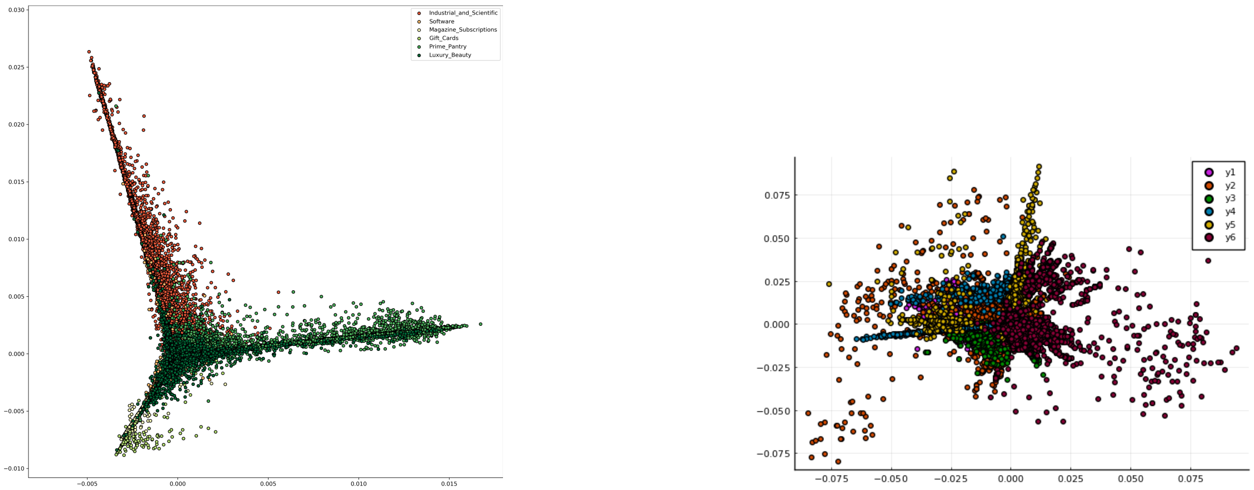
Consider a hypergraph $\mathcal{H} = (V, \mathcal{E})$ and a given set of seeds $R \subseteq V$. The linear equation formulation of the PageRank problem can also be described as a solution to the following optimization problem.

$$\min_{\mathbf{x}} \sum_{(i,j)\in\mathcal{E}} \mathbf{C}_{(i,j)}(x_i - x_j)^2 \tag{8.1}$$

where $\mathbf{C}$ is the edge weight matrix developed according to the $\mathcal{H}$. Individuals curious about how this relates to the PageRank formulation are redirected to [17]. Instead we describe the data we worked with and the identified parts of improvement.

8.2 Results

We attempt to develop embeddings for the amazon dataset([30]) with given seedsets of 6 different product categories. We sample 4% of nodes from each of the product categories as seeds and run the PageRank procedure with each of those nodes acting as the personalization vector. While factorizing the PageRank matrix thus formed, we augmented all the above PageRank vectors together, across all categories, and performed SVD. The pictures below depict the embeddings for the new categories, that is, the clusters obtained on performing sweep-cut of the PageRank vectors. These clusters consist of the unique nodes obtained from the PageRank vectors seeded on different nodes of the same product type.



(a) The amazon product graph with $\alpha = 0.99$ and without log

(b) The amazon product graph with $\alpha = 0.9$ and with log

Figure 8.1. Performance on Hypergraph

One of the obvious reasons we failed to generate good embeddings in this case is because the PageRank vector produced for each instance was sparse which is expected given the localized guarantees of the algorithm. As we saw the case for graph, without the element-wise log operation, the embeddings produced were not meaningful. While after the element-wise log operation, the $-\infty$ elements are replaced by the minimum elements in the matrix, which results in most of the elements being replaced by the same element and thus the resulting singular vectors do not produce good pictures. This is our primary goal for future work.

8.3 Future directions

Other than sharing the insight about the relation between the singular vectors and the average of PageRank, [13] and [31] show the weakness of spectral embedding, which we saw a hint of in figure (1.3), where distinct nodes have overlapping embeddings. The PageRank embedding technique discussed in this thesis was shown to be better than spectral embeddings in figure(19) of [13]. This underscores the efficiency of the former even when it does not appear to be similar to the latter. Further, [9] and [13] share the leaking effect of PageRank and suggest ways to counteract it. We also intend to address it in our future work.

REFERENCES

[1] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1, pp. 107–117, 1998, Proceedings of the Seventh International World Wide Web Conference, ISSN: 0169-7552. DOI: https://doi.org/10.1016/S0169-7552(98)00110-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016975529800110X.

[2] D. F. Gleich, "Pagerank beyond the web," *SIAM Rev.*, vol. 57, no. 3, pp. 321–363, Jan. 2015, ISSN: 0036-1445. DOI: 10.1137/140976649. [Online]. Available: https://doi.org/10.1137/140976649.

[3] I. M. Kloumann, J. Ugander, and J. Kleinberg, "Block models and personalized pagerank," *Proceedings of the National Academy of Sciences*, vol. 114, no. 1, pp. 33–38, 2017, ISSN: 0027-8424. DOI: 10.1073/pnas.1611275114. eprint: https://www.pnas.org/content/114/1/33.full.pdf. [Online]. Available: https://www.pnas.org/content/114/1/33.

[4] A. Tsitsulin, D. Mottin, P. Karras, A. Bronstein, and E. Müller, "Netlsd: Hearing the shape of a graph," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Jul. 2018. DOI: 10.1145/3219819.3219991. [Online]. Available: http://dx.doi.org/10.1145/3219819.3219991.

[5] R. Yang, J. Shi, X. Xiao, Y. Yang, and S. S. Bhowmick, "Homogeneous network embedding for massive graphs via reweighted personalized pagerank," *Proc. VLDB Endow.*, vol. 13, no. 5, pp. 670–683, Jan. 2020, ISSN: 2150-8097. DOI: 10.14778/3377369.3377376. [Online]. Available: https://doi.org/10.14778/3377369.3377376.

[6] A. N. Langville and C. D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings*. USA: Princeton University Press, 2006, ISBN: 0691122024.

[7] F. Chung, "The heat kernel as the pagerank of a graph," *Proceedings of the National Academy of Sciences*, vol. 104, no. 50, pp. 19 735–19 740, 2007, ISSN: 0027-8424. DOI: 10.1073/pnas.0708838104. eprint: https://www.pnas.org/content/104/50/19735.full.pdf. [Online]. Available: https://www.pnas.org/content/104/50/19735.

[8] C. Donnat, M. Zitnik, D. Hallac, and J. Leskovec, "Learning structural node embeddings via diffusion wavelets," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD'18, London, United Kingdom: Association for Computing Machinery, 2018, pp. 1320–1329, ISBN: 9781450355520. DOI: 10.1145/3219819.3220025. [Online]. Available: https://doi.org/10.1145/3219819.3220025.

[9] Y. Takai, A. Miyauchi, M. Ikeda, and Y. Yoshida, "Hypergraph clustering based on pagerank," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1970–1978, ISBN: 9781450379984. [Online]. Available: https://doi.org/10.1145/3394486.3403248.

[10] M. Liu, N. Veldt, H. Song, P. Li, and D. F. Gleich, *Strongly local hypergraph diffusions for clustering and semi-supervised learning*, 2020. arXiv: 2011.07752 [cs.SI].

[11] F. Tudisco, K. Prokopchik, and A. R. Benson, *A nonlinear diffusion method for semi-supervised learning on hypergraphs*, 2021. arXiv: 2103.14867 [cs.LG].

[12] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, and P. Talukdar, *Hypergcn: A new method of training graph convolutional networks on hypergraphs*, 2019. arXiv: 1809.02589 [cs.LG].

[13] K. Fountoulakis, M. Liu, D. F. Gleich, and M. W. Mahoney, *Flow-based algorithms for improving clusters: A unifying framework, software, and performance*, 2020. arXiv: 2004.09608 [cs.LG].

[14] D. F. Gleich, M. D. Rasmussen, K. J. Lang, and L. Zhukov, "Tthe world of music: User ratings; spectral and spherical embeddings; map projections," 2005. [Online]. Available: https://www.cs.purdue.edu/homes/dgleich/publications/Gleich%5C%202006%5C%20-%5C%20wom.pdf.

[15] D. F. Gleich, "Models and algorithms for PageRank sensitivity," Ph.D. dissertation, Stanford University, Sep. 2009. [Online]. Available: http://www.stanford.edu/group/SOL/dissertations/pagerank-sensitivity-thesis-online.pdf.

[16] T. Sahai, A. Speranzon, and A. Banaszuk, *Hearing the clusters in a graph: A distributed algorithm*, 2011. arXiv: 0911.4729 [cs.DM].

[17] D. Gleich and M. Mahoney, "Anti-differentiating approximation algorithms:a case study with min-cuts, spectral, and flow," in *Proceedings of the 31st International Conference on Machine Learning*, E. P. Xing and T. Jebara, Eds., ser. Proceedings of Machine Learning Research, vol. 32, Bejing, China: PMLR, 22–24 Jun 2014, pp. 1018–1025. [Online]. Available: http://proceedings.mlr.press/v32/gleich14.html.

[18] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," ser. NIPS'13, Lake Tahoe, Nevada: Curran Associates Inc., 2013, pp. 3111–3119.

[19] O. Levy and Y. Goldberg, “Neural word embedding as implicit matrix factorization,” in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27, Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/feab05aa91085b7a8012516bc3533958-Paper.pdf.

[20] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, “Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec,” in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, ser. WSDM ’18, Marina Del Rey, CA, USA: Association for Computing Machinery, 2018, pp. 459–467, ISBN: 9781450355810. DOI: 10.1145/3159652.3159706. [Online]. Available: https://doi.org/10.1145/3159652.3159706.

[21] A. Grover and J. Leskovec, “Node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16, San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 855–864, ISBN: 9781450342322. DOI: 10.1145/2939672.2939754. [Online]. Available: https://doi.org/10.1145/2939672.2939754.

[22] J. Tang, M. Qu, and Q. Mei, “Pte: Predictive text embedding through large-scale heterogeneous text networks,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’15, Sydney, NSW, Australia: Association for Computing Machinery, 2015, pp. 1165–1174, ISBN: 9781450336642. DOI: 10.1145/2783258.2783307. [Online]. Available: https://doi.org/10.1145/2783258.2783307.

[23] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “Line: Large-scale information network embedding,” in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW ’15, Florence, Italy: International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077, ISBN: 9781450334693. DOI: 10.1145/2736277.2741093. [Online]. Available: https://doi.org/10.1145/2736277.2741093.

[24] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’14, New York, New York, USA: Association for Computing Machinery, 2014, pp. 701–710, ISBN: 9781450329569. DOI: 10.1145/2623330.2623732. [Online]. Available: https://doi.org/10.1145/2623330.2623732.

[25] D. A. Spielman and S.-H. Teng, “Spectral partitioning works: Planar graphs and finite element meshes,” *Linear Algebra and its Applications*, vol. 421, no. 2, pp. 284–305, 2007, Special Issue in honor of Miroslav Fiedler, ISSN: 0024-3795. DOI: https://doi.org/10.1016/j.laa.2006.07.020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0024379506003454.

[26] S. Chanpuriya and C. Musco, “Infinitewalk: Deep network embeddings as laplacian embeddings with a nonlinearity,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1325–1333, ISBN: 9781450379984. [Online]. Available: https://doi.org/10.1145/3394486.3403185.

[27] R. Ibrahim and D. F. Gleich, “Local hypergraph clustering using capacity releasing diffusion,” *PLOS ONE*, vol. 15, no. 12, I. Sendiña-Nadal, Ed., e0243485, Dec. 2020, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0243485. [Online]. Available: http://dx.doi.org/10.1371/journal.pone.0243485.

[28] P. Li and O. Milenkovic, “Submodular hypergraphs: P-laplacians, Cheeger inequalities and spectral clustering,” in *Proceedings of the 35th International Conference on Machine Learning*, J. Dy and A. Krause, Eds., ser. Proceedings of Machine Learning Research, vol. 80, Stockholmsmässan, Stockholm Sweden: PMLR, Oct. 2018, pp. 3014–3023. [Online]. Available: http://proceedings.mlr.press/v80/li18e.html.

[29] M. Hein, S. Setzer, L. Jost, and S. S. Rangapuram, “The total variation on hypergraphs - learning on hypergraphs revisited,” in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS’13, Lake Tahoe, Nevada: Curran Associates Inc., 2013, pp. 2427–2435.

[30] J. Ni, J. Li, and J. McAuley, “Justifying recommendations using distantly-labeled reviews and fine-grained aspects,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 188–197.

[31] K. Lang, “Fixing two weaknesses of the spectral method,” in *Advances in Neural Information Processing Systems*, Y. Weiss, B. Schölkopf, and J. Platt, Eds., vol. 18, MIT Press, 2005. [Online]. Available: https://proceedings.neurips.cc/paper/2005/file/045cf83ab0722e782cf72d14e44adf98-Paper.pdf.