2008

Web Search Algorithms and PageRank

Laleh Samarbakhsh
Wilfrid Laurier University

NOTE TO USERS

Your file Votre référence
ISBN: 978-0-494-38720-7
Our file Notre référence
ISBN: 978-0-494-38720-7

WEB SEARCH ALGORITHMS AND PAGERANK

by

Laleh Samarbakhsh

(BSc, Sharif University of Technology, 2005)

THESIS

Submitted to the Department of Mathematics

in partial fulfillment of the requirements for

Master of Science in Mathematics

Wilfrid Laurier University

2008

Abstract

The mathematical theory underlying the Google search engine is the PageRank algorithm, first introduced by Sergey Brin and Lawrence Page, the founders of Google. A ranking of web pages is made considering many criteria. PageRank exploits the graph structure of the web. The web's hyperlink structure forms a massive directed graph, where the web pages are presented as nodes and hyperlinks as edges. The PageRank equation finds a score by solving a recursive equation which calculates the PageRank vector. The PageRank vector is the stationary distribution of an ergodic Markov chain. The Perron-Frobenius theorem ensures that the primitive matrix produced by this massive Markov chain will converge to a unique stationary distribution. The PageRank vector existence is guaranteed since the so-called Google matrix is stochastic and has all entries positive.

In a recent work by Litvak, Scheinhardt and Volkovich [**14**], a mathematical model is presented that explains an interesting relation between PageRank values and in-degrees in power law graphs. They analytically prove that in power law graphs, the tail distributions of PageRank and in-degree differ only by a multiplicative factor.

We survey the mathematics of the PageRank algorithm, and study the work of Litvak et. al. We implement a PageRank calculator and expose different graphs to our calculator. For various power law graphs, we show that the ranking of the nodes by PageRank will be the same as the ranking given by in-degree. We give a counterexample for graphs which are not power law. For these graphs, the ranking derived from PageRank is different from the ranking derived from the in-degree values.

Keywords: graphs, directed graphs, PageRank, Google matrix, Markov chains, random walk, power law graph, binary tree

Acknowledgements

I would like to express my sincere thanks to my thesis supervisor, Dr. Anthony Bonato. He has been very helpful and provided me with constant supervision all throughout my Master's here at Laurier. I would especially like to thank him for providing financial support and encouraging me to take part in AARMS summer school in 2007, where I accomplished part of the requirements for my degree.

Special thanks to all my graduate course instructors Dr. Marc Kilgour, Dr. Joe Campolieti, Dr. Zilin Wang and Dr. Roderick Melnik from whom I learnt and strengthened my knowledge in different aspects of Mathematics. I would like to thank Dr. George Lai, Dr. Zilin Wang, and Dr. Dejan Delić for serving on my thesis committee. I am always grateful to Dr. Sydney Bulman-Fleming for his contribution to my growth as a graduate student.

Last but not least, I would like to thank both my family and my friends for their never-ending emotional support which is essential for my work.

Contents

List of Figures

CHAPTER 1

Introduction

1.1. Motivation

With the rapid growth of the world wide web, information retrieval presents increasing theoretical and practical challenges. With the massive amount of information entering the world wide web every moment, it becomes harder and harder to retrieve information from the web. That is why the presence of a search engine is as vital as the existence of the web itself. Since the birth of the web, it has been a central discussion in the web research community to design faster, more efficient, and more accurate search engines.

The most popular search engine currently is Google. The mathematical theory behind the Google search engine is the PageRank algorithm, which was introduced by Sergey Brin and Lawrence Page [**3**], the founders of Google. In 1998, Brin and Page were PhD students. They took a leave of absence from their Ph.D. to focus on developing their

Google prototype. Their pioneering paper described the PageRank algorithm, which is used to this day by Google to generate its rankings.

A search engine consists of key components: a crawler, and indexer, and a query engine [**2**]. The *crawler* collects and stores data from the web. Data is stored in an *indexer* which extracts information from the data collected from the crawler. The *query engine* responds to queries from users. As part of the query engine, a *ranking algorithm* ranks web pages in order of their relevance to the query. The ranking is achieved by the assignment of a score to each web page.

PageRank is a ranking algorithm of web pages and uses the link structure of the web. The web's link structure forms a directed graph where the web pages are represented as nodes and links as directed edges. A page is considered "important" if it is pointed to by other important pages. The following PageRank equation finds a score by solving the iterative equation:

$$\pi^T = \pi^T(\alpha \mathbf{S} + (1-\alpha)\mathbf{J}),$$

where $\mathbf{J}$ is the matrix of all 1's whose order equals the number of pages. The matrix $\mathbf{S}$ is the stochastic matrix associated to the directed adjacency matrix of the web graph. The parameter α is called the *teleportation factor*, a constant between 0 and 1 which is normally assumed to be around 0.85, and π is the PageRank vector [**13**]. PageRank will be discussed in detail in Chapter 2.

The PageRank vector consisting of the PageRank of each web page is the stationary value of a large ergodic Markov Chain [**3**]. The Perron-Frobenius theorem is used to ensure that the so-called *Google matrix* associated with this Markov Chain will converge to a stationary distribution [**15**]. The Perron-Frobenius theorem supplies a unique normalized positive dominant eigenvector, called the *Perron Vector*, which is the PageRank vector of the Google matrix.

In a recent work by Litvak, Scheinhardt, and Volkovich [**14**], a mathematical model is presented that derives an interesting relation between PageRank values and in-degrees of web pages. They investigate why the PageRank and in-degree of

web pages follow similar power laws in the web graph. Furthermore, they analytically prove that in power law graphs, the tail distributions of PageRank and in-degree differ only by a multiplicative factor [**14**].

The aim of my thesis is to first survey the mathematics of the PageRank algorithm, and then to investigate the recent work of [**14**]. In Chapter 2, I introduce PageRank and describe its key properties. I will implement a PageRank calculator and expose different graphs to my calculator. In Chapter 3, we summarize the work of [**14**], who proved that the ranking of the nodes by PageRank in power law graphs will be similar to their ranking via their in-degree values. For binary trees, we show in Chapter 4 that the ranking result from PageRank is different from the ranking of their in-degree values.

What follows in this chapter is the background and definitions needed throughout my thesis. As we will see, the mathematical study PageRank uses a blend of graph theory, probability, and linear algebra.

1.2. Graph Theory

This section gives a concise introduction to the graph theory terminology used later in my thesis. For a general reference in graph theory, see [**6**]. A *graph* G consists of a nonempty *vertex set* $V(G)$, and an *edge set* $E(G)$ of 2-element subsets from $V(G)$. A graph is sometimes called *network*, especially with regards to real-world examples. More formally, we may consider $E(G)$ as a binary relation on $V(G)$ which is irreflexive and symmetric. We often write $G = (V(G), E(G))$, or if G is clear from context, then we write $G = (V, E)$. The set E may be empty. Elements of V are *vertices*, and elements of E are *edges*. Vertices are occasionally referred to as *nodes*, while edges are referred to as *lines* or *links*. We write uv for an edge $\{u, v\}$, and say that u and v are *joined* or *adjacent*; we may as well say that u and v are *incident* to the edge uv, and that u and v are *endpoints* of uv. The most common way to visualize a graph is by drawing a dot for each node and joining two of these dots by a line if the corresponding two nodes form an edge. By a *non-empty graph*, we mean a graph with at least one edge.

We allow graphs to have multiple edges, but no loops. A *simple graph* is a graph without multiple edges. The cardinality $|V(G)|$ is the *order* of G, while $|E(G)|$ is its *size*. For a node $v \in V(G)$, $deg_G(v)$ is the *degree* of v in G, namely the number of edges in G incident with v. A node of degree 0 is *isolated*.

For a node x in a graph G, define the *neighbourhood* of x, written $N_G(x)$, to the nodes joined to x. For $X \subseteq V(G)$, $N_G(X)$ is the union of the neighborhoods over nodes from X. If $X \subseteq V$, then define the *subgraph induced by* X, written $G \restriction X$ (or as either $\langle X \rangle_G$ or $G[X]$), to be the graph with nodes from X, with two nodes joined in $G \restriction X$ if and only if they are joined in G. A *subgraph* of G is a graph H such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. A graph G is called *bipartite* if $V(G)$ admits a partition into two classes such that every edge has its ends in different classes (hence, nodes in the same partition class must not be adjacent).

A graph may be directed or undirected. A *directed* graph or *digraph* is defined analogously as an undirected graph, except that now $E(G)$ need not be a symmetric binary relation on $V(G)$. The edges are written as ordered pairs,

and are called *directed edges*, (u, v), where u is the *head* and v is the *tail*. The directed edge (u, v) is then said to be *directed* from u to v. All the previously mentioned features and definitions can then be modified to directed graphs. The *in-degree* of u, written $\deg^-(u)$, is the number of nodes v such that (v, u) are directed edges; the *out-degree* $\deg^+(u)$ is defined dually. Moreover, a directed graph is called *strongly connected* if for each pair of nodes (v_i, v_j), there is a sequence of directed edges leading from v_i to v_j. The directed graph in Figure 1.1 is strongly connected. In

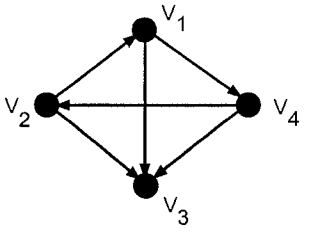

FIGURE 1.1. A strongly connected digraph.

all of the above definitions, we will not mention G if it is clear from the content.

One of the most important examples of a graph for us is the *web graph*. It is the graph where the nodes represent web pages, and the edges correspond to links between the

pages. We write W for this graph, which is a real-world evolving graph. We may consider W an undirected or directed graph, depending on the context.

A key property of the web graph is the presence of a power-law degree distributions. Given a graph G and a non-negative integer k, we define $N_{k,G}$ by

$$N_{k,G} = |\{x \in V(G) : \deg_G(x) = k\}|.$$

The parameter $N_{k,G}$ is *number of nodes of degree k in G.* The *degree distribution* of G is the sequence $(N_{k,G} : 0 \leq k \leq t)$. The degree distribution of G follows a *power law* if for each degree k,

$$\frac{N_{k,G}}{t} \sim k^{-\beta},$$

for a fixed real constant $\beta > 1$. We say that β is the *exponent of the power law.* A graph whose degree distribution follows a power law is often referred to as a *power law graph.* Power laws for the in-degree and out-degree distributions may be defined in a similar fashion. The in- and out-degree distributions of the web graph were observed to follow power law in the experiments conducted by Broder

et al. [4], which sampled 200 million web pages and their links. For additional reading on the web graph, the reader is directed to the books [**2, 5, 8**].

1.3. Linear Algebra

Matrices and vectors will be denoted in **bold**. Further, all vectors are column vectors unless otherwise stated. For a matrix $\mathbf{A}$, we use the notation a_{ij} for the i,j-entry of $\mathbf{A}$. An $m \times n$ matrix $\mathbf{A}$ is a *non-negative* matrix whenever each $a_{ij} \geq 0$, and this is denoted by writing $\mathbf{A} \geq 0$. The notation $\mathbf{A} \geq \mathbf{B}$ means that each $a_{ij} \geq b_{ij}$. A matrix $\mathbf{A}$ is *positive* when each $a_{ij} > 0$, and this is denoted by writing $\mathbf{A} > 0$. More generally, $\mathbf{A} > \mathbf{B}$ means that each $a_{ij} > b_{ij}$.

A convenient representation of a graph is via its adjacency matrix. The *adjacency matrix* $\mathbf{A}(G)$ of a digraph G is defined by

$$a_{ij} = \begin{cases} 1 & \text{if } v_i v_j \in \mathbf{E}(G), \\ 0 & \text{otherwise.} \end{cases}$$

If G is undirected of order n, then $\mathbf{A}(G)$ is an $n \times n$ symmetric (that is, $\mathbf{A}(G) = \mathbf{A}(G)^T$) matrix. Adjacency matrices are non-negative.

For an $n \times n$ matrix $\mathbf{A}$, a scalar λ for which

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0$$

is called an *eigenvalue* of $\mathbf{A}$. A nonzero $n \times 1$ vector $\mathbf{x}$ for which $\mathbf{A} = \lambda\mathbf{x}$ is the *eigenvector* of λ for $\mathbf{A}$. The pair $(\lambda, \mathbf{x})$ is called an *eigenpair* for $\mathbf{A}$. The set of all distinct eigenvalues, denoted by $\sigma(\mathbf{A})$, is called the *spectrum* of $\mathbf{A}$.

The eigenvalues and eigenvectors are fundamental topics in PageRank calculations. The adjacency matrix $\mathbf{A}(G)$ for an undirected graph G is a real and symmetric matrix, and hence, has n real eigenvalues $\lambda_1 > 0, \lambda_2, \ldots, \lambda_n$, which can be ordered by their absolute values:

$$\lambda_1 = |\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|.$$

(See, for example, [**2**].) The first (that is, largest in absolute value) eigenvalue λ_1 is the *radius* of the spectrum, denoted by $\rho(\mathbf{A})$. The real number λ_1 is also called the *dominant eigenvalue.*

We now state Perron-Frobenius theorem. A proof of this important result may be found in [**15**]. A non-negative matrix $\mathbf{A}$ is *primitive* if $\mathbf{A}^m > 0$ for some $m > 0$. The

1-*norm* (or *taxicab norm*) of $\mathbf{x}$ is defined as

$$||\mathbf{x}||_1 = \sum_{i=1}^{n} |x_i|.$$

THEOREM 1.3.1 (Perron-Frobenius). *If a matrix* $\mathbf{A} \geq 0$ *is primitive, then each of the following assertions holds.*

(1) $r = \rho(\mathbf{A}) > 0$.

(2) *There exists an eigenvector* $\mathbf{x} > 0$ *such that* $\mathbf{Ax} = r\mathbf{x}$.

(3) *The Perron vector* $\mathbf{p}$ *is the unique vector satisfying*

$$\mathbf{Ap} = r\mathbf{p}$$

and which is positive with 1*-norm equal to* 1. *There are no other non-negative eigenvectors for* $\mathbf{A}$ *regardless of the eigenvalue, except for the positive multiples of* $\mathbf{p}$.

We will sometimes use limits of matrices. If $(\mathbf{M}_t)$ is a sequence of $m \times n$ matrices, and $\mathbf{L}$ is an $m \times n$ matrix, then we write

$$\lim_{t \to \infty} \mathbf{M}_t = \mathbf{L}$$

if for all $1 \leq i \leq m$, $1 \leq j \leq n$,

$$\lim_{t \to \infty} (\mathbf{M}_t)_{i,j} = \mathbf{L}_{i,j}.$$

1.4. Markov Chains

Markov Chains provide a powerful framework for modelling certain random processes. Our approach to analyze the PageRank algorithm in Chapter 2 will use Markov chains. We give a brief discussion of Markov chains in this section. For a general reference in probability theory, see [**11**].

Fix n a positive integer. We denote $\mathbb{P}(A)$ the probability of an event A in a probability space. A (*discrete-time, time-homogeneous, finite-state*) *Markov chain* M consists of a discrete-time random process $(X_t : t \in \mathbb{N})$ each with codomain in the same finite set $S = \{a_0, \ldots, a_n\}$ with the property that for all $n \geq 1$ and $1 \leq t \leq n$,

$$\mathbb{P}(X_t = a_t | X_{t-1} = a_{t-1}, \ldots, X_0 = a_0) = \mathbb{P}(X_t = a_t | X_{t-1} = a_{t-1}).$$

This definition expresses that the state X_t depends on the previous state X_{t-1}, but is independent of *how* we actually arrived at X_{t-1}. In other words, the random process does not remember the way it reached the state X_{t-1}. This property is called *Markovian* or *memoryless* property for a random process. It is important to note that the Markov property does not imply that the state X_t does not depend on the random variables $X_0, X_1, \ldots, X_{t-2}$. However, what a Markovian property guarantees for X_t, is that any such dependency on the past will be captured and recorded in the value of X_{t-1}. In other words, only the present state gives any information of the future behaviour of the process. See [**16**] for more background on Markov chains.

The set of possible values S of M is called the *state space*, and without loss of generality we will always consider this to be $\{1, \ldots, n\}$, where n is an integer. The *transition probability*

$$P_{i,j} = \mathbb{P}(X_t = j | X_{t-1} = i)$$

is the probability that the process moves from state i to state j in one time-step. Using the Markovian property,

every Markov chain can be uniquely expressed by a *transition matrix* defined as

$$\mathbf{P} = \begin{pmatrix} P_{0,0} & P_{0,1} & \cdots & P_{0,j} & \cdots \\ P_{1,0} & P_{1,1} & \cdots & P_{1,j} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{i,0} & P_{i,1} & \cdots & P_{i,j} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}.$$

Hence, the i,j-entry in the matrix is the transition probability $P_{i,j}$. The representation of a Markov chain via its transition matrix makes it feasible to compute and predict the distribution of the future states of the process.

A useful representation of a Markov chain is via a directed weighted graph. The nodes correspond to the states, and the weight on each directed edge is the positive transition probability of getting from the head state to tail state. There is a directed edge (i, j) if and only if $P_{i,j} > 0$. A *stochastic* matrix is a non-negative matrix in which each row sum is equal to 1. Note that the transition matrix of every Markov chain is a stochastic matrix (which follows from the basic probability definitions). A *stationary distribution*

$\mathbf{s}$ of a Markov chain is a *probability distribution* (that is, a vector whose sum of all entries equals 1) with the property that

$$\mathbf{s}^T = \mathbf{s}^T\mathbf{P}.$$

We can also express this by saying that $\mathbf{s}$ is an eigenvector of $\mathbf{P}$ with eigenvalue 1, or $\mathbf{s}$ is a *fixed-point* of $\mathbf{P}$.

Stationary distributions exist and are unique if the Markov chain has a primitive transition matrix [**2**]. We refer to such Markov chains as *ergodic*. Hence, if we consider an ergodic Markov chain over a long period of time, the initial state becomes increasingly forgotten, and the probability that we are in state i approaches the ith component of $\mathbf{s}$. As we will see in Chapter 2, the PageRank vector corresponds to the stationary distribution of a certain Markov chain.

CHAPTER 2

The PageRank Algorithm

2.1. Introduction and Motivation

Information retrieval is the process of searching within a collection of documents for a particular item of information. The information you are looking for is normally called a *query*. To retrieve information from the world wide web, we need to first be able to model the web. The best way to model a massive network like the web is by representing it as a digraph. Each web page is a node of the graph and the links between two nodes are directed edges. To perform a search in this network, we should first be able to gather all of the information about its link structure in a database, and then classify and retrieve the query from this database.

A search engine consists of a *crawler*, *indexer* and a *query engine*; see [**13**]. See Figure 2.1 for a simplified model of a search engine.

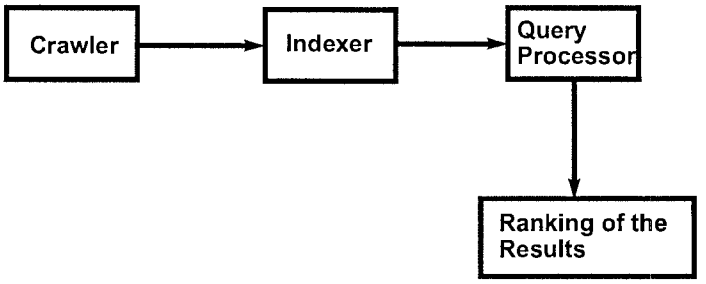

FIGURE 2.1. A basic search engine model.

The crawler performs frequent visits to the entire (or a large part of the) world wide web. The crawler travels from page to page to keep track of the existing links, and more importantly to update our database with new web pages and links. Imagine a backpacker who is walking through every link, and upon arriving at every new web page, writes down the address of the page and summarizes its content. There are certainly web pages that have no out-links. Hence, our backpacker will get stuck there. After recording such pages (called *dangling nodes*), the backpacker will step back as many steps needed to be able to find a way out.

The links are classified once they are entered into the indexer. Hence, now we can search for our query in the

indexer. The search is done, and some number of pages associated to the query are found.

After the search is done, let us say there are 200 pages found in the end. The question becomes: how to rank these pages? A useful way to display the 200 pages on the result screen is to rank them by popularity. We therefore need to find out how to rank web pages according to their popularity. This is the role of the query engine. As we will describe in Section 2.2, the PageRank algorithm is one effective way to accomplish this ranking.

2.2. Random Walks on Graphs

Before we define PageRank, we make a short digression to discuss random walks. A *random walk* on a connected graph G is a certain type of Markov chain defined by the sequence of moves (over discrete time-steps) of a particle between nodes of G. The location of the particle at a given time-step is its *state*. In the *uniform random walk*, the particle may move from its current state to any of its neighbouring nodes (with equal probability). A uniform random walk on a graph of size n may be represented by a transition

probability matrix $\mathbf{P}$ whose entries are p_{ij}, where

$$p_{ij} = \begin{cases} \frac{1}{deg(i)} & \text{if } j \in N(i), \\ 0 & \text{otherwise.} \end{cases}$$

Note that the transition probability matrix of a uniform random walk is *stochastic*; that is, the row sums are equal to 1. We define an *ergodic Markov chain* to be one whose transition matrix is primitive. An important theorem states that an ergodic Markov chain always has a stationary distribution [**19**]. Hence, the stationary probability distribution exists for the uniform random walk if its transition matrix is primitive, and is a probability vector $\mathbf{s^T}$ such that

$$\mathbf{s^T P} = \mathbf{s^T}.$$

The following theorem states sufficient conditions on G for the stationary distribution to exist.

THEOREM 2.2.1. [**2**] *Let G be a finite, connected, nonbipartite graph. A random walk on G converges to a stationary distribution $\mathbf{s^T} = (s_i)$, where*

$$s_i = \frac{deg(i)}{2|E(G)|}.$$

Analogous result holds for uniform random walks on directed graphs.

2.3. The Google Matrix

To define the Google matrix for an arbitrary graph G, we consider the transition probability matrix for the uniform random walk on G. Let $n = |V(G)|$ be the order of G and apply a fixed enumeration from 1 to n to the nodes of G. For the directed graph G, the matrix $\mathbf{P}_1$ is defined by

$$P_1(i,j) = \begin{cases} \frac{1}{deg^+(i)} & \text{if } (i,j) \in E(G), \\ 0 & \text{otherwise.} \end{cases}$$

The structure of the $\mathbf{P}_1$ matrix guarantees that at every node, the surfer will have equal probability to choose one of the out-neighbours. If there is no out-link from i to j, then this probability is 0. In the web there always exist web pages that do not link to any other web pages. These nodes are called *dangling* nodes. If we assume the only way to visit the web pages is by following the out-links, then the surfer gets stuck at such nodes. To overcome this problem, we manipulate the matrix $\mathbf{P}_1$ in a way to bypass the dangling nodes. Define $\mathbf{P}_2$ to be the matrix $\mathbf{P}_1$ such

that any zero rows are replaced with the vector with each entry equal to $\frac{1}{n}$. Define the *Google* matrix (or *PageRank* matrix) by

$$\mathbf{P} = \alpha \mathbf{P}_2 + \frac{1-\alpha}{n} \mathbf{J}_{n,n},$$

where α is a fixed real number in $(0, 1)$, and $\mathbf{J}_{n,n}$ is the $n \times n$ matrix of all 1's. (We do not use the notation $\mathbf{G}$ for the Google matrix, as G is reserved for graphs.) The constant α, called the *teleportation factor*, is a parameter measuring the frequency at which a surfer jumps to a new randomly chosen web page, rather than following the out-links. We now show why Google matrix is stochastic and primitive, and hence, has a stationary distribution.

LEMMA 2.3.1. *For a graph G with order n and $\mathbf{P} = \mathbf{P}(G)$ equalling its Google matrix, then the following assertions hold.*

(1) *The matrix $\mathbf{P}$ is stochastic.*

(2) *The matrix $\mathbf{P}$ is primitive.*

Proof. For item (1), to show that $\mathbf{P}$ is stochastic, we must show that the row sums in $\mathbf{P}$ are all equal to 1. For

a fixed $0 < i < n$, the row sum r_i equals:

$$\begin{aligned} r_i &= \sum_{1\le j\le n} \left(\alpha(\mathbf{P}_2)_{i,j} + \frac{1-\alpha}{n}(\mathbf{J}_{n,n})_{i,j} \right) \\ &= \alpha \sum_{1\le j\le n} (\mathbf{P}_2)_{i,j} + \frac{1-\alpha}{n} \sum_{1\le j\le n} 1 \\ &= \alpha \sum_{1\le j\le n} (\mathbf{P}_2)_{i,j} + (1-\alpha)\,. \end{aligned}$$

To find the value for $\sum_{1\le j\le n} (\mathbf{P}_2)_{i,j}$, we consider two cases.

Case 1. Node i is dangling node.

In this case,

$$(\mathbf{P}_2)_{i,j} = \frac{1}{n}.$$

Hence,

$$\begin{aligned} r_i &= \alpha \sum_{1\le j\le n} \frac{1}{n} + (1-\alpha) \\ &= \alpha + (1-\alpha) = 1. \end{aligned}$$

Case 2. Node i is not dangling.

In this case,

$$(\mathbf{P}_2)_{i,j} = (\mathbf{P}_1)_{i,j}.$$

Hence,

$$\begin{aligned} r_i &= \alpha \sum_{1 \leq j \leq n} (\mathbf{P}_1)_{i,j} + (1-\alpha) \\ &= \alpha + (1-\alpha) = 1. \end{aligned}$$

For item (2), since all entries of $\mathbf{P}$ are positive, $\mathbf{P}$ is primitive. □

Lemma 2.3.1 demonstrates that the Google matrix $\mathbf{P}$ is a transition probability matrix of an ergodic Markov chain. The Markov chain associated to this matrix is called the *PageRank Markov chain*, or the *PageRank random walk*. In this random walk, at any page, the surfer visits an out-neighbour of that node with probability α and visits any other node in G with probability $1 - \alpha$. In practice, the parameter α is normally assumed to be around 0.85; see [**13**].

We will now use the linear algebra preliminaries stated in Chapter 1 of the thesis to prove an important theorem about the PageRank random walk. The following theorem

guarantees that with the described structure of Google matrix, the PageRank random walk has a unique stationary distribution, called the *PageRank vector.*

THEOREM 2.3.2. *Fix a graph G. The PageRank Markov chain with transition probability matrix $\mathbf{P} = \mathbf{P}(G)$ converges to a unique stationary distribution* $\mathbf{s}$.

Proof. Since $\mathbf{P}$ is positive and primitive, the PageRank Markov chain is ergodic, and hence, converges to a stationary distribution $\mathbf{s}$. To show that $\mathbf{s}$ is unique, we will use the Perron-Frobenius theorem (See Theorem 1.3.1). By Theorem 1.3.1, $\mathbf{P}$ has a unique positive and dominant eigenvalue equal to 1. The corresponding eigenvector for this eigenvalue would be the vector $\mathbf{s}$, where

$$\mathbf{s}^T\mathbf{P} = \mathbf{s}^T. \qquad \square$$

The vector $\mathbf{s}$ is the PageRank vector for the Google matrix $\mathbf{P}$. The entries in the PageRank vector are the PageRank values for each node in the graph G (associated with the fixed enumeration of $V(G)$). To calculate the PageRank values, we need to find the stationary vector of the Google

matrix. In Section 2.4 we discuss a practical method used to calculate the PageRank, called the *power method.* In the next section, we explain another approach to calculate the PageRank vector first used by Brin and Page [**3**].

2.4. Another View of PageRank

The original formula for PageRank due to [**3**], is a summation formula which calculates the popularity of the pages by adding up the PageRank of all the pages pointing to this web page. Let $PR(P_i)$ denote the PageRank of the page P_i and let $In(P_i)$ denote the set of web pages that point to P_i. The PageRank is then

$$PR(P_i) = \sum_{P_j \in In(P_i)} \frac{PR(P_j)}{|P_j|}. \tag{2.1}$$

A problem is that the values for $PR(P_j)$ are unknown. To overcome this, we need to initialize all the web pages with an equal PageRank value, and then transform equation (2.1) into a recursive equation. Brin and Page assumed that at the beginning all the pages have a constant PageRank value of $\frac{1}{n}$, where n is the total number of pages in the web graph. The iterative procedure calculates PageRank

at $(k+1)$-th step as

$$PR_{k+1}(P_i) = \sum_{P_j \in In(P_i)} \frac{PR_k(P_j)}{|P_j|}. \tag{2.2}$$

The process initiates with setting $PR_0(P_i) = \frac{1}{n}$ for all pages P_i. As discussed in the previous section, since the PageRank Markov chain is ergodic, the eventual convergence of the PageRank scores is guaranteed.

2.5. The World's Largest Matrix Computation

Cleve Moler, the founder of the well-known mathematical software *matlab*, cited PageRank as "The World's Largest Matrix Computation" in [**17**]. At that time Google was applying the Power Method to a sparse matrix of order 2.7 billion. Now, it has at least 54 billion rows and columns! (See [**2**].)

To find the PageRank vector, we should solve for the eigenvector $\mathbf{s}$ such that

$$\mathbf{s}^T\mathbf{P} = \mathbf{s}^T.$$

Since $\mathbf{P}$ is a dense massive matrix, a direct approach to the calculations will not be feasible in general. To overcome

the computational problems, the *power method* is used to approximate the PageRank vector $\mathbf{s}$. The algorithm works as follows. Fix a directed graph G of order n.

(1) Initialize $\mathbf{z}_0$ to be the stochastic vector with every entry equal to $1/n$.

(2) Define

$$\mathbf{z}_{k+1}^T = \mathbf{z}_k^T \mathbf{P} = (\mathbf{z}_0^T)\mathbf{P}^k$$

The sequence $(\mathbf{z}_k : k \in \mathbb{N})$ consists of stochastic vectors, since at every time step, we have the result of the product of two stochastic matrices. It can be shown that

$$\lim_{k \to \infty} \mathbf{z}_{k+1}$$

is the dominant eigenvector of the Google matrix; see [**2**].

From the Power method, we can approximate the PageRank vector by taking powers of the Google matrix and multiplying it by $\mathbf{z}_0$. This amounts to simply summing up each column of $\mathbf{P}$ and multiplying the sum by $1/n$.

For completeness, we give an illustration of a PageRank computation. Figure 2.5 shows a sample graph G with six nodes. To find the PageRank vector of G, we first compute

the various matrices required in the definition of the Google matrix.

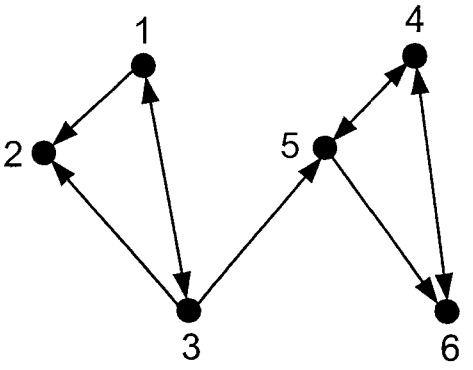

FIGURE 2.2. A directed graph G.

The $\mathbf{P_1}$ matrix for G is

$$\mathbf{P_1} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

The $\mathbf{P_2}$ matrix is

$$\mathbf{P_2} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

If $\alpha = 0.85$, then let $a = \alpha/2$, $b = (1-\alpha)/6$, $c = \alpha/6$, and $d = \alpha/3$. Then the Google matrix of G is then

$$\begin{pmatrix} b & a+b & a+b & b & b & b \\ b+c & b+c & b+c & b+c & b+c & b+c \\ b+d & b+d & b & b & b+d & b \\ b & b & b & b & a+b & a+b \\ b & b & b & a+b & b & a+b \\ b & b & b & \alpha+b & b & b \end{pmatrix}.$$

Using the power method, the approximate PageRank vector (using the natural ordering $\{1, 2, 3, 4, 5, 6\}$, and with two decimal places of accuracy) is

$$\begin{pmatrix} 0.05 & 0.07 & 0.06 & 0.35 & 0.2 & 0.27 \end{pmatrix}.$$

2.6. Implementation of a PageRank Calculator

In this section, a brief description of an implementation of PageRank is given on a graph with n nodes. The PageRank calculator implemented for the thesis uses the algorithm provided in [**13**].

(1) The vector `pi0` is the initial vector, which we normally set to $1/n$.
(2) `H` is the manipulated hyperlink matrix, P_2.
(3) `n` is the size of the matrix or the web.
(4) `alpha` is the teleportation factor.
(5) `epsilon` is the convergence tolerance; in the actual implementation, we set the total iteration steps equal to 20.
(6) The vector `a` is the dangling node vector in which an entry is 1 if its corresponding node is a dangling node, and 0, otherwise.

```
%Implementation of PageRank calculator
%using power method
function [pi,time,numiter]=
    PageRank(pi0,H,n,alpha,epsilon);
```

```
rowsumvector=ones(1,n)*H';
nonzerorows=find(rowsumvector);
zerorows=setdiff(1:n,nonzerorows);
l=length(zerorows);
a=sparse(zerorows,ones(1,1),ones(1,1),n,1);
k=0;
residual=l;
pi=pi0;
tic;
for ( i=0:20 )
%while(residual < epsilon)
  prevpi=pi;
  k=k+1;
  pi=alpha*pi*H + (alpha*(pi*a)+1-alpha)
    *((1/n)*ones(1,n));
  residual=norm(pi-prevpi,1);
end;
numiter=k;
time=toc;
%save pi;
```

CHAPTER 3

PageRank in Power Law Graphs

PageRank roughly measures the popularity of a web page based on its number of in-links. We discussed PageRank in detail in Chapter 2, and we proved that it is the stationary distribution of the PageRank random walk. We now present recent work by Litvak et al. [**14**], who proved that under certain assumptions PageRank and in-degree distributions of a power law digraph obey a power law with the same exponent. To prove this result, we model the relation between PageRank and in-degree via a stochastic equation. All the results described in this chapter come from [**14**].

Studying the potential similarity between PageRank and in-degree of the web pages is of particular importance because it provides ground for simpler, cheaper and less time-consuming calculations. The matrix calculations performed to estimate PageRank, are massive. However, it is straight forward to find the in-degree of a web page.

To study the behaviour of the PageRank tail, Laplace-Stieltjes transforms are used. The Tauberian Theorem and the theory of regularly varying variables are then applied to a certain stochastic equation to prove analytically that the tails of PageRank and in-degree distributions vary only in a multiplicative constant. Hence, the PageRank and in-degree distributions in power law graphs follow power laws with the same exponent.

We begin by recalling the PageRank equation in its summation form. (See Equation 2.1 from the previous chapter.)

$$PR(i) = \alpha \sum_{j \in N(i)} \frac{PR(j)}{d_j} + (1 - \alpha). \tag{3.1}$$

An interpretation of (3.1) is that the PageRank of node i depends on the in-degree of i and PageRank of its in-neighbours. However, it is important to note that while the linear algebraic methods often used in PageRank literature work well for most PageRank computations, they are not sufficient for analyzing the asymptotic properties of the PageRank distribution. The mathematical approach to PageRank analysis used in [**14**] stems more from applied

probability and stochastic operations research, than from linear algebra.

In Donato et al. [**7**], Fortunato et al. [**9**] and Becchetti and Castillo [**1**], experiments performed on the web graph confirm the similarity in tail behavior of PageRank and in-degree distributions. The exponent value β for the power laws of the PageRank and in-degree distributions were found in all cases to be around 1.1. Moreover, the cited experimental studies have shown that the PageRank of the top 10% of the nodes always follows a power law with the same exponent independent of the teleportation factor α.

In a power law distribution, there is a so-called 30-70 rule: the tail will cover 70 percent of the value of the distribution. We will therefore compare the tail distribution of PageRank and in-degree. In other words, we focus on *tail asymptotics* for PageRank and its relation with in-degree. Since we are only interested in the tail, we are looking into web pages with high popularity or PageRank value, which can be stated as

$$\mathbb{P}(PR > x), \tag{3.2}$$

for a suitably large x, and where $\mathbb{P}(A)$ is the probability of the event A in a probability space. Observe that (3.2) defines the fraction of pages having PageRank greater than x, where x is large. One way to analyze such a probability is to find an asymptotic expression $p(x)$ for which

$$\lim_{x\to\infty} \frac{\mathbb{P}(PR > x)}{p(x)} = 1.$$

If such a $p(x)$ is found, then $p(x)$ and $\mathbb{P}(PR > x)$ are asymptotically equal, and so we can approximate the tail of PageRank by $p(x)$.

3.1. Regularly Varying Random Variables

A real-valued function $RV(x)$ is said to be *regularly varying* of index $\beta \in \mathbb{R}$ if for every $t > 0$,

$$\lim_{x\to\infty} \frac{RV(tx)}{RV(x)} = t^{\beta}.$$

A real-valued function $SV(x)$ is said to be *slowly varying* if for every $t > 0$,

$$\lim_{x\to\infty} \frac{SV(tx)}{SV(x)} = 1.$$

A careful look at the above definitions leads us to a relation between RV and SV functions: every regularly varying function can be written as

$$RV(x) = x^{\beta} SV(x),$$

for some slowly varying function. We can also define the *regularly varying* property for random variables as well as functions. Recall that $a(x) \sim b(x)$ if

$$\lim_{x \to \infty} \frac{a(x)}{b(x)} = 1.$$

A random variable X is said to be *regularly varying with index* β if its distribution $F(x)$ can be written as

$$1 - F(x) \sim \ x^{-\beta} SV(x),$$

for some slowly varying function $SV(x)$. The *Laplace-Stieltjes transform* of X is

$$f(s) = \mathbb{E}[e^{-sX}],$$

where $s > 0$ and $\mathbb{E}(Y)$ is the expectation of the random variable Y. The *n-th moment* of X is written as

$$\xi_n = \int_0^\infty x^n dF(x).$$

By expanding f in a series at $s = 0$, the successive moments of F can be obtained. Moreover, the n-th moment of X is finite if and only if there exist coefficients $\xi_0, \ldots, \xi_n$ such that $f_n(s) = o(s^n)$, as $s \to 0$. The following lemma states this in a precise fashion.

LEMMA 3.1.1 ([14]). *The n-th moment of X is finite if and only if there exist real numbers $\xi_0 = 1$ and $\xi_1, \ldots, \xi_n$, such that*

$$\lim_{s \to 0} f(s) = \sum_{i=0}^{n} \frac{\xi_i}{i!}(-s)^i + o(s^n).$$

The above coefficients ξ_i may be uniquely found. If $\xi_n < \infty$, then we write:

$$f_n(s) = (-1)^{n+1} \left(f(s) - \sum_{i=0}^{n} \frac{\xi_i}{i!}(-s^i) \right).$$

Later, we will use $f_n(s)$ further to discuss the tail properties of the distribution. There exist an important relation

between asymptotic behaviour of a regularly varying distribution and its Laplace-Stieltjes transform. The following theorem, used throughout this chapter, makes this relation precise.

THEOREM 3.1.2 (Tauberian Theorem; [**14**]). *For $n \in \mathbb{N}$ and if $\xi_n < \infty$, $\beta = n+\eta$, and $\eta \in (0,1)$, then the following are equivalent*

(1) $f_n(s) \sim (-1)^\beta \Gamma(1-\beta) s^\beta SV(\frac{1}{s})$, *as* $s \to 0$

(2) $1 - F(x) \sim x^{-\beta} SV(x)$, *as* $x \to \infty$.

The proofs of the above lemma and theorem may be found in [**14**]. Theorem 3.1.2 plays an important role in finding the relation between asymptotic distributions of PageRank and in-degree.

3.2. The Relationship between In-degree and PageRank

We now describe the relationship between PageRank and in-degree. We consider equation (3.1), but make some important simplifying assumptions. These assumptions will enable us to model this relation by focusing on the influence of in-degree without considering other factors. Naturally,

these assumptions are not realistic, but in further discussions we try to reduce them by generalizing the model obtained. Rewrite equation (3.1) in the following form of a distributional identity with the random variable R:

$$R \stackrel{d}{=} \alpha \sum_{j=1}^{M} \frac{1}{d} R_j + (1-\alpha), \tag{3.3}$$

where $\stackrel{d}{=}$ represents a distributional identity and M is the in-degree of the considered random page. The assumptions we make are as follows.

(1) Let R represent the PageRank of a randomly chosen page. One of our goals in this chapter is to determine the distribution of the random variable R.
(2) Fix $d \geq 1$ and assume that it is the number of outgoing links for *all* pages. Hence, out-degree is equal for all nodes.
(3) The *dangling node* effect is neglected. That is, we do not consider the effect of pages without outgoing links.
(4) The random variables R and M are independent. That is, the in-degree distribution and PageRank

distribution of a random page have independent distributions (which is not the case, in general).

(5) All R_j's are independent and have the same distribution as R and hence, $R \equiv 1$ constitutes the unique solution of the equation (3.3).

The equation (3.3) has the same form as the original PageRank formula as in equation (3.1).

We will now find the in-degree distribution for a randomly chosen web page. Although it is well-known that the in-degree distribution of the web graph follows power law (see for example, [**2**]), we need to be able to formally describe this random variable for our analysis. We use the theory of regularly varying random variables. The in-degree of a randomly chosen page is modeled by a non-negative integer-valued, regularly varying random variable which is distributed as $N(X)$. In particular, the random variable X is regularly varying with index β and $N(x)$ is *the number of Poisson arrivals* on the time interval $[0, x]$. For more details on Poisson processes and their application in Markov chains, see Sections 8.2 and 8.3 of [**12**].

The variable $N(x)$ is a "discretization" of the random variable X. In this way, we guarantee that while in-degree has a power law distribution, it only takes integer values and hence, we do not have to put any restrictions on X. In Theorem 3.2.1 below, we will prove that $N(X)$ is also regularly varying with the same index as X, and so follows a power law with the same exponent. First, let F_X and $F_N(X)$ be the distribution functions of X and $N(X)$, respectively. Let f and ϕ be their corresponding Laplace-Stieltjes transforms.

THEOREM 3.2.1 ([**14**]). *The following are equivalent.*

(1) $1 - F_X(x) \sim x^{-\beta} SV(x)$, *as* $x \to \infty$

(2) $1 - F_{N(X)} \sim x^{-\beta} SV(x)$, *as* $x \to \infty$

We give a brief sketch of the theorem. We first need a technical lemma.

LEMMA 3.2.2 ([**14**]). *Let* $f_n(s)$ *and* $\phi_n(s)$ *be the Laplace-Stieltjes transforms of* X *and* $N(X)$, *respectively. Then*

$$f_n(s) = o(s^n) \text{ if and only if } \phi_n(s) = o(s^n).$$

While we omit the proof of the lemma, here is an informal sketch of its proof. One shows that the corresponding moments of X and $N(X)$ always exist. It may be shown that since we fixed the out-degree of all pages to be equal to d, then the average in-degree would also equal d. That is, $\mathbb{E}[X] = d$ and similarly, $\mathbb{E}[N(X)] = d$. The final step in the proof of Lemma 3.2.2 is to consider the generating function of $N(X)$ and derive its Laplace-Stieltjes transform in terms of the Laplace-Stieltjes transform of the random variable X.

We now sketch a proof of Theorem 3.2.1.

Proof of Theorem 3.2.1. We only prove that (1) implies (2). From Theorem 3.1.2 in the previous section, we have that

$$1 - F_X(x) \sim x^{-\beta} SV(x), \text{ as } x \to \infty$$

implies that

$$f_n(t) \sim (-1)^{\beta} \Gamma(1-\beta) t^{\beta} SV\left(\frac{1}{t}\right), \text{ as } t \to 0$$

where n is the largest integer smaller than β, and Γ is the gamma function. Since $\phi(s) = f(t)$, by Lemma 3.2.2 we

have that $f_n(s) \sim o(s^n)$ where $t(s) = 1 - e^{-s} \sim s$. By Lemma 3.1.1 and Theorem 3.1.2, we have that

$$1 - F_{N(X)}(x) \sim x^{-\beta} SV(x) \text{ as } x \to \infty. \qquad \square$$

The model of Litvak et al. for the number of incoming links of a randomly chosen web page works well, since it describes an in-degree distribution which follows a power law with finite expectation and a non-integer exponent $\beta > 1$. Having obtained the distribution for in-degree and PageRank, we will now proceed to retrieve the main stochastic equation for the relation between PageRank and in-degree and compare their tail distributions in the next section.

3.3. Stochastic Equations

Using the discussion in the previous two sections, we can now reformulate the equation (3.3) as follows:

$$R \stackrel{d}{=} \alpha \sum_{j=1}^{N(X)} \frac{1}{d} R_j + (1 - \alpha), \tag{3.4}$$

where $\alpha \in (0, 1)$ is the teleportation factor, $d \geq 1$ is the fixed out-degree of each page, and $N(X)$ describes the in-degree of a randomly chosen page in terms of the Poisson

arrivals on a regularly varying random variable X which represents time. The stochastic equation (3.4) adequately represents the PageRank distribution and its relation with the in-degree distribution. We can now apply analytical methods to study the tail behaviour.

The main idea of the analysis is to apply the Laplace-Stieltjes transforms of X and R. By using the Tauberian Theorem, we may prove that R is regularly varying with the same index as X. By the Theorem 3.2.1, this then guarantees similarity in the tail behaviour of the PageRank R and the in-degree $N(X)$.

The first step is to write the Laplace-Stieltjes transform of the PageRank distribution R in terms of the probability generating function of $N(X)$. Let $\mathbb{G}_{N(X)}$ be the generating function of $N(X)$. By applying Laplace-Stieltjes transform

to the definition of R in (3.4) we have that

$$\begin{aligned} r(s) &= \mathbb{E}[e^{-sR}] = \mathbb{E}[e^{-s(1-\alpha)}]\mathbb{E}\left[\exp\left(-s\frac{\alpha}{d}\sum_{i=1}^{N(X)} R_i\right)\right] \\ &= e^{-s(1-\alpha)}\sum_{k=1}^{\infty}\mathbb{E}\left[\exp\left(-s\frac{\alpha}{d}\sum_{i=1}^{k} R_i\right)\right]\mathbb{P}(N(X)=k) \\ &= e^{-s(1-\alpha)}\sum_{k=1}^{\infty}\left(r\left(s\frac{\alpha}{d}\right)\right)^k \mathbb{P}(N(X)=k) \\ &= e^{-s(1-\alpha)}\mathbb{G}_{N(X)}\left(r\left(s\frac{\alpha}{d}\right)\right). \end{aligned}$$

Note that for all i, we have $R_i \stackrel{d}{=} R$, and that is how the second equality in the above set of equations is obtained. In the following corollary, we prove that $\mathbb{G}_{N(X)}$ can be expressed in terms of the Laplace-Stieltjes transform of X.

COROLLARY 3.3.1 ([**14**]). *If $\mathbb{G}_{N(X)}$ is the generating function of $N(X)$ and f is the Laplace-Stieltjes transform of X, then we have that*

$$\mathbb{G}_{N(X)}(s) = f(1-s).$$

Proof. By definition of the generating function,

$$\begin{aligned}
\mathbb{G}_{N(X)}(s) &= \mathbb{E}[s^{N(X)}] \\
&= \int_0^\infty \mathbb{E}[s^{N(t)}] dF_X(t) \\
&= \int_0^\infty e^{-t(1-s)} dF_X(t) \\
&= f(1-s). \qquad \square
\end{aligned}$$

The above corollary leads us to an important conclusion for the Laplace-Stieltjes transform of R. By Corollary 3.3.1, as $\mathbb{G}_{N(X)}(s) = f(1-s)$ we obtain that

$$r(s) = f(1 - r(\frac{\alpha}{d}s))e^{-s(1-\alpha)}. \tag{3.5}$$

As in the previous section where the distribution for in-degree was calculated, here we perform the analysis by showing the correspondence between the existence of the n-th moments of X and R. The independence of $N(X)$ and the R_j's is heavily used. For example, using this independence, we can take the expectation from both sides

of (3.4). Similar to the result of Lemma 3.1.1, we can reformulate it and show that

$$f_n(s) = o(s^n) \text{ if and only if } r_n(s) = o(s^n). \tag{3.6}$$

Now we can present the final theorem in this chapter in which the observed correlation between in-degree and PageRank distributions is explained in power law graphs. The proof of the theorem can be found in [**14**].

THEOREM 3.3.2. *The following are equivalent.*

(1) $1 - F_{N(X)}(x) \sim x^{-\beta} SV(x)$, *as* $x \to \infty$ *In particular,* $N(X)$ *is a regularly varying random variable with index* β.

(2) $1 - F_R(x) \sim \frac{\alpha^\beta}{d^\beta - \alpha^\beta d} x^{-\beta} SV(x)$, *as* $x \to \infty$. *In particular,* R *is a regularly varying random variable with the same index* β.

Theorem 3.3.2 shows that the asymptotic behaviour of PageRank and in-degree are similar in power law graphs, since they both follow a power law with equal exponents (they differ only by the multiplicative factor $\frac{\alpha^\beta}{d^\beta - \alpha^\beta d}$).

3.4. Numerical Experiments and Conclusion

How do we verify a power law behaviour in practice? It is not always simple to plot, measure, or numerically identify power law distributions. A well-known technique is to plot the so-called log-log graph of the distribution. More precisely, we plot the degree distribution in logarithmic scale and expect to obtain a straight line. Experiments conducted by Newman in [**18**] suggests that since we are focusing on tail distributions, we should plot the fraction of quantities which are not less than a certain value. In particular, we should plot the complementary cumulative function instead; that is,

$$1 - F(x) = \mathbb{P}(X > x),$$

rather than to plot the histogram. In this way, we will have a more concentrated plot.

Another issue is that if a distribution X follows power law with exponent β such that $1-F(x) \sim Cx^{-\beta}$, where C is a constant, then the corresponding histogram has exponent

$\beta + 1$. Thus, the plot of $1 - F(x)$ on logarithmic scale will have a smaller slope than the original plot of the histogram.

Computation of the correct slope from real-world data is also an important part of the numerical analysis. Goldstein et al. in [**10**] suggest that using an MLM (Maximum Likelihood Method) is advantageous over the standard least square fit method, since the former provides us with a more robust estimation of the power law exponent. The calculations based on MLM yield a slope of -1.1 which confirms that both in-degree and PageRank have power laws with the same exponent $\beta = 1.1$.

In the results retrieved from the experiments using web data, Litvak et al. focus on the right tail behaviour of the PageRank distribution. The result is that in a log-log plot, both in-degree and PageRank distributions plot as parallel lines for all values of the teleportation factor, as long as we focus on large PageRank values. In fact, comparing PageRank and in-degree does depend on the teleportation factor. However, the PageRank distribution of the top 10% of web pages obeys a power law with the same exponent as in the in-degree, independent of the teleportation factor.

Despite their results, the Litvak et al.'s model however, lacks the realistic dependencies between the PageRank values of the pages sharing a common neighbor. This is why the exact value of the multiplicative constant provided in Theorem 3.3.2 does not fit the results from their web crawls. Further work would be to reduce the assumptions made in Section 1.2 so that the generalized model can capture mainly the dangling node effect and the dependencies between PageRank values of the pages pointing to one certain web page.

In conclusion, Litvak et al. showed that in power law graphs, PageRank and in-degree follow the same power law distribution which varies only in a multiplicative constant. In the next chapter, we provide examples of graphs where the PageRank and in-degree do not follow similar tail distributions.

CHAPTER 4

PageRank and In-degree

4.1. Introduction

In this chapter, we supply some examples complementing the findings of [**14**]. Before we begin, let us have a quick review of the materials discussed in Chapter 2 on PageRank. The PageRank vector for a digraph G is calculated by first calculating the PageRank matrix

$$\mathbf{P} = \mathbf{P}(G) = \alpha \mathbf{P}_2 + \frac{(1-\alpha)}{n}\mathbf{J}_{n,n}, \tag{4.1}$$

where $\mathbf{J}_{n,n}$ is the $n \times n$ matrix of all 1's and $\alpha \in (0,1)$ is the teleportation constant. The matrix $\mathbf{P}$ defined on the left hand side of the equation above is the *Google matrix* (or *PageRank matrix*). The PageRank matrix is positive and stochastic, and therefore, is the transition matrix for some Markov chain.

The Markov chain attributed to the PageRank matrix converges to a stationary distribution $\mathbf{s}$. This convergence

is guaranteed as it is an ergodic Markov chain. Since $\mathbf{s}$ is the dominant eigenvector of the transition probability matrix of this Markov chain, we have that

$$\mathbf{s}^T\mathbf{P} = \mathbf{s}^T.$$

The vector $\mathbf{s}$ is called the *PageRank vector*, whose ith entry is the PageRank of the ith node of the graph (according to some fixed enumeration of the nodes). Hence, to calculate the PageRank vector of a graph, we should find the stationary distribution of the Google matrix $\mathbf{P}$ in (4.1).

A good approximation to the PageRank vector can be evaluated using the *Power method*, discussed earlier in Section 2.3. For this method, we start with an initial (arbitrary but fixed) non-negative, non-zero vector $\mathbf{s}_0$, and then define

$$\begin{aligned} \mathbf{s}_{t+1}^T &= \mathbf{s}_t^T \\ &= (\mathbf{s}_0^T)\mathbf{P}^t. \end{aligned} \tag{4.2}$$

After a sufficient number of iterations (normally 20 to 50 in practice; see [**3**]), $\mathbf{s}$ approximates the PageRank vector. The iterative process in (4.2), presents a useful alternative

for calculating the $\mathbf{s}$. In (4.2), there are two steps: first, we raise the Google matrix $\mathbf{P}$ to a power t and then multiply it by the vector $\mathbf{s}_0$. If we take $\mathbf{s}_0$ to be the vector of all 1's, then this multiplication will give the column sum of the matrix $\mathbf{P}$. Hence, the PageRank vector is simply the column sum of the limiting vector in the powers of the Google matrix (which is later on normalized to ensure it is stochastic).

The Google matrix, however, is a dense matrix and the Power Method calculations involving matrix multiplication become increasingly costly as higher powers are formed. An alternative is to only work with the sparse matrix $\mathbf{P}_2$. In this case, the stationary distribution of the uniform random walk is computed (not PageRank).

Litvak et al. [**14**] introduced numerical methods and a new model that proves that with certain assumptions, in power law graphs, the PageRank and in-degree distributions are similar. This result is interesting and of practical importance because PageRank calculations are costly when compared to the computation of in-degree. (To find the in-degree of the ith node, simply find the ith column sum of the adjacency matrix. The adjacency matrix of the web

graph is sparse.) Litvak et al. [14] proved that this result is true for power law graphs, but not for arbitrary graphs. The main goal of the coming sections is to provide examples of graphs whose PageRank and in-degree distributions are distinct.

4.2. Binary Trees

A *tree* is a connected, acyclic digraph; a *rooted tree* has a distinguished node called the *root*. A *binary tree* is a rooted tree in which every node other than the leaves have in-degree equalling 2. For a fixed $i \in \mathbb{N}$, the ith *row* of a binary tree consists of those nodes which are connected to the root by a directed path of length exactly $i - 1$. Define $T_2(r)$ to be a binary tree with r rows.

There are several interesting properties for binary trees. For instance, the set of nodes of $T_2(r)$ may be identified with a set of finite 0-1 sequences (or strings), with the root representing the empty sequence. Figure 4.1 displays such a binary string labelling.

Our goal is to calculate the PageRank for every node in the binary tree, and then compare the ranking with the

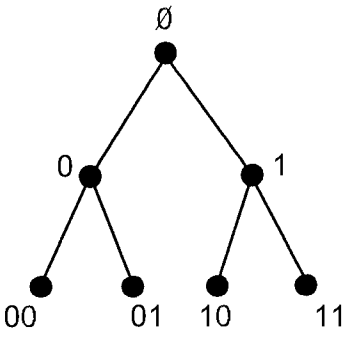

FIGURE 4.1. The binary tree $T_2(3)$ with its 0-1 labelling.

in-degree of the nodes. As we will see for small examples, the PageRank and in-degree distributions of binary trees do not correlate. We conjecture that this holds in general for all binary trees. For larger examples, while we do not prove this directly, we offer evidence for this conjecture by proving that the stationary distribution of the uniform random walk on the binary tree does not correlate with the in-degree distribution.

We first need some notation for $T_2(r)$. This will help to quickly recognize on which row each node is located. Let $x_{i,j}$ denote the i-th node on the j-th row of the binary tree. The Figure 4.2 shows such a labelling for $T_2(3)$.

Although the proof of the following lemma is folklore, we include it for completeness.

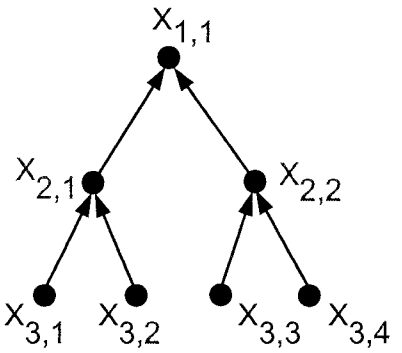

FIGURE 4.2. The binary tree $T_2(3)$ with $x_{i,j}$ labelling.

LEMMA 4.2.1. *Fix an integer* $r \geq 1$.

(1) *For* $0 < i \leq r$, *the number of nodes on the* i*-th row (assuming the root to be the 1st row) of the binary tree* $T_2(r)$, *is* 2^{i-1}.

(2) *The binary tree* $T_2(r)$ *has order* $n = 2^r - 1$.

We note that all throughout this chapter we assume the binary tree to be a *full* binary tree, meaning that all of the leaves are on the same level and every non-leaf node has two children.

Proof: For item (1), we perform induction on i. For the base step of the induction, consider the first row of nodes in $T_2(r)$. As $i = 1$, hence, $2^{i-1} = 2^0 = 1$. But there is exactly one node in the first row.

The induction hypothesis assumes that on row i, we have 2^{i-1} nodes. Moving to the row $i+1$, every node in row i has two children (since the binary tree is full) and so the number of nodes on row $i+1$ is twice the number of nodes on row i:

$$\begin{aligned} \#(\text{nodes on row } i+1) &= 2 \times \#(\text{nodes on row } i) \\ &= 2 \times 2^{i-1} \\ &= 2^i. \end{aligned}$$

For item (2), the total number of nodes in a full binary tree $T_2(r)$, is counted by adding up the total number of nodes on each row. Hence:

$$\begin{aligned} |T_2(r)| &= 2^1 + 2^2 + 2^3 + \ldots + 2^r \\ &= \sum_{i=0}^{r-1} 2^i \\ &= 2^r - 1. \qquad \square \end{aligned}$$

4.3. Calculating the Stationary Distribution

Calculating the exact PageRank vector for the binary tree would require us to compute all the powers of the dense

Google matrix. As we are presently not able to perform this calculation, we decided instead to calculate the stationary distribution of a uniform random walk on the binary tree. As PageRank is the stationary distribution of the uniform random walk with teleportation, our results are suggestive of the actual PageRank values. The proofs in this chapter are original work.

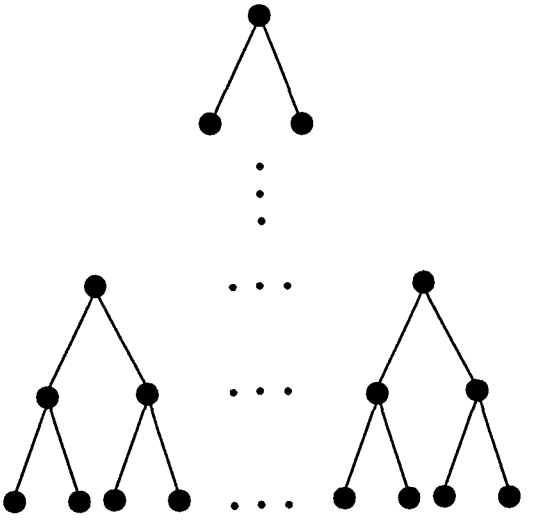

FIGURE 4.3. An arbitrary binary tree

A binary tree is depicted in Figure 4.3. The adjacency matrix (namely $\mathbf{P}_1$) for the binary tree has the following

structure.

$$\mathbf{P_1} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & \dots \\ 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

The matrix $\mathbf{P}_1$ contains a nice pattern: starting from the second row, every two consecutive rows are equal. The next pair of rows results by a single shifting of the previous pair of rows to the right. For $T_2(r)$, this pattern continues until the 1's reach the (2^{r-1})-th column of the matrix. Since the leaves of the tree have zero in-degree, the matrix will have a rectangular block of zeros of size $(2^r - 1) \times 2^{r-1}$ on its right side. The relative simplicity of this pattern allows a rigorous analysis of the uniform random walk on $T_2(n)$.

Consider the $\mathbf{P}_2$ matrix for $T_2(r)$. Recall that the $\mathbf{P}_2$ matrix is just $\mathbf{P}_1$ without zero rows. In the binary tree, the root or the node $x_{1,1}$ is the unique dangling node, where the random surfer would become stuck in the uniform random walk. Hence, we will assume that the root is pointing to

all other nodes in the graph; this assumption turns $\mathbf{P}_2$ into a stochastic matrix. The $\mathbf{P}_2$ matrix for the binary tree is therefore,

$$\mathbf{M} = \mathbf{P}_2 = \begin{pmatrix} 1/n & 1/n & \dots & 1/n \\ 1 & 0 & 0 & \dots \\ 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix},$$

where $n = |V(T_2(r))|$. Throughout, let

$$\mathbf{e} = \mathbf{J}_{n,1} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}.$$

We now state the main results of this section

THEOREM 4.3.1. *Let* $\mathbf{H}$ *be the* $\mathbf{P}_2$ *matrix of the binary tree* $T_2(r)$. *Fix* k *a positive integer. Define* $[\mathbf{H}^k]_{p,i}$ *to be the sum of the column corresponding to the node* $x_{p,i}$. *For all*

$k \geq 1$ and $1 \leq p \leq r$,

$$[\mathbf{H}^k]_{p,i} = [\mathbf{H}^k]_{p,j},$$

where $1 \leq i \leq j \leq 2^{p-1}$. In particular, the column sums of any two nodes on the same row in $T_2(r)$ are equal.

THEOREM 4.3.2. *Let $\mathbf{H}$ be the $\mathbf{P}_2$ matrix of the binary tree $T_2(r)$. For all $k \geq 1$, $1 \geq i \geq j \geq 2^{p-1}$ and $1 \leq p \leq r-1$, we have that*

$$[\mathbf{H}^k]_{p,i} = 2[\mathbf{H}^k]_{p+1,j} + \frac{1}{n}[\mathbf{H}^k]_{1,1};$$

where $n = 2^r - 1$.

Note that $\frac{1}{n}[\mathbf{H}]_{1,1}$ does not depend on either p or j. We defer the proofs of Theorem 4.3.1 and 4.3.2 until the following section. We have however, the following corollary.

COROLLARY 4.3.3. *Let $\mathbf{s}$ be the stationary distribution vector of the $\mathbf{P}_2$ matrix of the binary tree $T_2(r)$.*

(1) *For any two nodes on the same row of $T_2(r)$, the corresponding entries in $\mathbf{s}$ are equal.*

(2) *For all $1 \leq p \leq r-1$, the entry of $\mathbf{s}$ corresponding to any node on the p-th row is approximately twice*

the entry corresponding to any of the nodes on the $(p+1)$*-st row of* $\mathbf{s}$.

Proof: For the proof of (1), by definition we have that

$$\mathbf{s}^T = \lim_{t\to\infty} \mathbf{e}^T\mathbf{H}^{\mathbf{t}}. \tag{4.3}$$

Now apply $[\cdot]_j$ to both sides of (4.3), representing the sum of the j-th column (or as in this case, the j-th element of the vector) on both sides of the limit:

$$\begin{aligned}
[\mathbf{s}^T]_j &= [\lim_{t\to\infty} \mathbf{e}^T\mathbf{H}^{\mathbf{t}}]_j \\
&= \lim_{t\to\infty} [\mathbf{e}^T\mathbf{H}^{\mathbf{t}}]_j \\
&= 1 \cdot \lim_{t\to\infty} [\mathbf{H}^{\mathbf{t}}]_j.
\end{aligned}$$

Note that $[\mathbf{s}]_j = [\mathbf{e}^T\mathbf{H}^{\mathbf{t}}]_j$ represents the j-th element of the vector. By Theorem 4.3.1, since $\mathbf{H}$ is the $\mathbf{P}_2$ matrix of the binary tree $T_2(r)$, for all t and for $1 \le i, j \le 2^{p-1}$,

$$[\mathbf{H}^t]_{p,i} = [\mathbf{H}^t]_{p,j}.$$

But the column sum $[\mathbf{H}^t]_{p,i}$ represents the stationary value for the node $x_{p,i}$, namely $[\mathbf{s}]_{2^{p-1}+i-1}$; similarly $[\mathbf{H}^t]_{p,j}$ is the stationary value for the node $x_{p,j}$, namely $[\mathbf{s}]_{2^{p-1}+j-1}$. Hence,

the corresponding $(2^{p-1}+i-1)$-th and $(2^{p-1}+j-1)$-th entries in the stationary vector are the same. As i and j were arbitrary, any two nodes on the same row have equal entries in $\mathbf{s}$. The proof of item (1) follows.

For the proof of item (2), since $\mathbf{s}$ is the stationary distribution of the $\mathbf{P}_2$ matrix of the binary tree, we can apply Theorem 4.3.2 to $\mathbf{H}$. As in the proof of the previous corollary item, we use the definition of $\mathbf{s}$, given in (4.3) for the entries i and j corresponding to the nodes lying on two consecutive rows p and $p+1$:

$$\begin{aligned}
[\mathbf{s}^T]_i &= [\lim_{t\to\infty} \mathbf{e}^T\mathbf{H}^{\mathbf{t}}]_i \\
&= \lim_{t\to\infty}[\mathbf{e}^T\mathbf{H}^{\mathbf{t}}]_j \\
&= 1\cdot\lim_{t\to\infty}[\mathbf{H}^{\mathbf{t}}]_j,
\end{aligned}$$

Using part (1) of the corollary, by Theorem 4.3.2, and the fact that the jth node is located on the next row right after, we can write for any fixed $t>0$ and for $1\le i,j\le 2^{p-1}$,

$$[\mathbf{H}^t]_{p,i} = 2\times[\mathbf{H}^t]_{p+1,j} + \frac{1}{n}[\mathbf{H}^t]_{1,1}.$$

But the column sum $[\mathbf{H}^t]_{p,i}$ represents the stationary value for the node $x_{p,i}$, and similarly $[\mathbf{H}^L]_{p+1,j}$ is the stationary value for the node $x_{p+1,j}$, any node on the $(p+1)$-th row. Hence, the corresponding $(2^{p-1}+i-1)$-th entry is approximately twice the $(2^{p-1}+j-1)$-th entry in the stationary vector. As i and j were arbitrarily chosen, any two nodes on the two consecutive rows will have this property in $\mathbf{s}$. □

By Corollary 4.3.3, we see that the stationary distribution decreases from the largest value at the root to the smallest value at the leaves. This is analogous to a power law, since the leaves are the most abundant nodes. However, the in-degree distribution has either values 0 or 2. In particular, the stationary distribution for the random walk and the in-degree distribution are quite different. We conjecture that an analogous difference occurs when comparing the PageRank distribution with the in-degree distribution.

Before moving on to the next section, let us make this conjecture more plausible by comparing the PageRank and in-degree for $T_2(4)$. The graph of $T_2(4)$ is given in Figure 4.4.

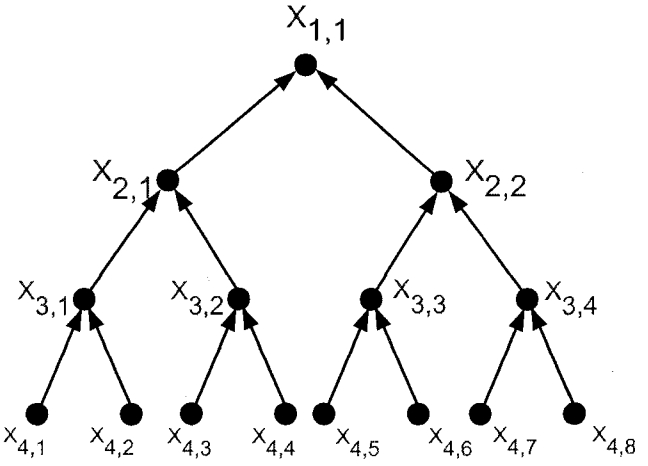

FIGURE 4.4. The directed binary tree $T_2(4)$.

After calculating the adjacency matrix $\mathbf{P}_1$, we change the row one zero values to 1/15 to recover the dangling node $x_{1,1}$. The result of this step is the $\mathbf{P}_2$ matrix given by

$$\mathbf{P}_2 = \begin{pmatrix} 1/15 & 1/15 & 1/15 & \dots \\ 1 & 0 & 0 & \dots \\ 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Assuming the teleportation factor α to be equal to 0.85 and using the power method with $t = 20$ iterations, we

calculate the PageRank vector to be approximately

$$(0.28 \quad 0.14 \quad 0.14 \quad 0.06 \quad 0.06 \quad 0.06 \quad 0.06 \quad 0.02$$
$$0.02 \quad 0.02 \quad 0.02 \quad 0.02 \quad 0.02 \quad 0.02 \quad 0.02)$$

Therefore, the nodes in $T_2(r)$ can be ranked by PageRank as

$$(1 \quad 2 \quad 2 \quad 3 \quad 3 \quad 3 \quad 3 \quad 4 \quad 4 \quad 4 \quad 4 \quad 4 \quad 4 \quad 4 \quad 4),$$

which implies that the root is the highest ranked page (as expected). Moreover, all nodes on the same row, have equal rank.

As we can see in Figure 4.4, the in-degree of all the nodes of $T_2(4)$ is 2, except for the leaves which have 0 in-degree. The in-degree vector, written **ID**, has its first seven entries equal to 2 (corresponding to the non-leaf nodes) and the next eight elements equal to 0 (corresponding to the leaves

of the tree):

$$\mathbf{ID} = \begin{pmatrix} 2 \\ \vdots \\ 2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Thus, the in-degree ranks the pages as

$$(1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2),$$

implying that the first seven pages in the graph (that is, all the non-leaf nodes) have equal ranking. All the leaves come in second position.

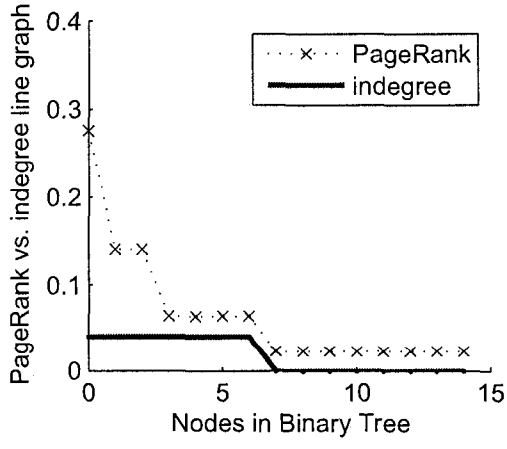

FIGURE 4.5. PageRank versus in-degree for $T_2(4)$.

In Figure 4.6, the values for PageRank and in-degree are plotted in one graph, so we can see the difference between their rankings. As it is evident from the figure, the PageRank and in-degree in this example do not correlate: PageRank follows a rough power law, while the in-degree is a step function.

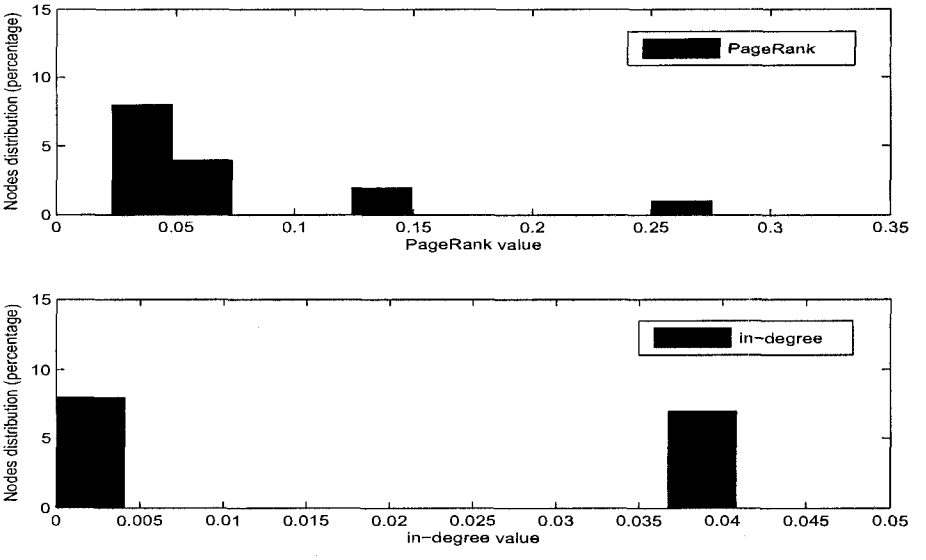

FIGURE 4.6. PageRank versus in-degree distributions for the binary tree $T_2(4)$.

This is more clearly sketched through the plot of the PageRank distribution of the nodes versus the in-degree distribution of the nodes, given in separate histograms in Figure 4.6.

4.4. Proofs of Main Results

Consider the nodes on two consecutive rows of $T_2(r)$. Let us say the first row starts with the node $x_{i,1}$ and the second row starts with the node $x_{i+1,1}$. Note that we already know there are 2^{i-1} nodes on the i-th row and 2^i nodes on the $(i+1)$-st row. It is essential to know which column in the adjacency matrix the node $x_{i,j}$ will be presented by. Always counting from top to bottom and left to right, the node $x_{i,j}$ is the j-th node on the i-th row. So, counting the nodes, we have total of $2^{i-1}-1$ nodes before the i-th row begins; adding it up with the j nodes until we reach $x_{i,j}$, we will have that the node $x_{i,j}$ is the $((2^{i-1}-1)+j)$-th node in $T_2(r)$. Hence, we now know that the in-degree, and the stationary distribution of the node $x_{i,j}$ will be presented through the $(2^{i-1}+j-1)$-th column of the corresponding adjacency matrices.

Before stating the proof for the theorems, we need lemmas from linear algebra:

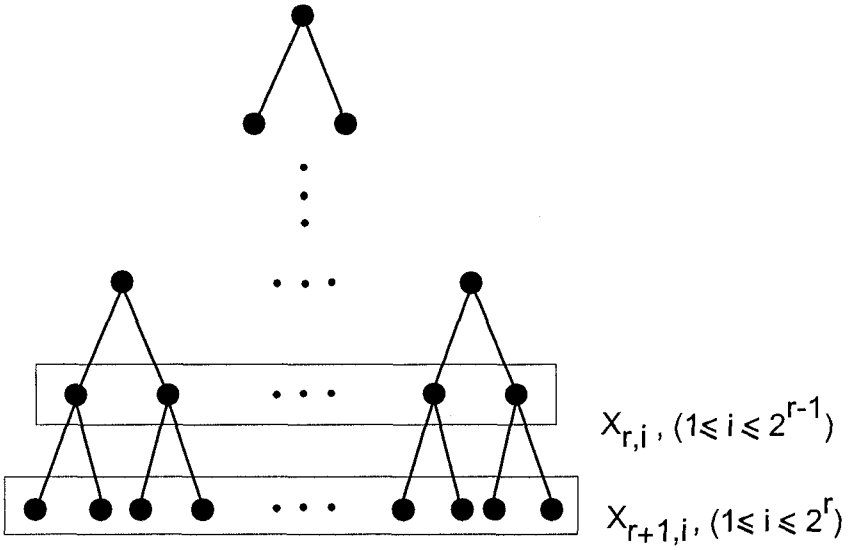

FIGURE 4.7. The general form of a binary tree with the labeled nodes. Here $x_{r,i}$ denotes the i-th node on the r-th row.

LEMMA 4.4.1. *For every matrix* $\mathbf{A}$, *and a vector*

$$\alpha = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix},$$

we have that

$$A\begin{pmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_n\end{pmatrix} = [\mathbf{A}_1|\mathbf{A}_2|\ldots|\mathbf{A}_n]\begin{pmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_n\end{pmatrix}$$
$$= \alpha_1\mathbf{A}_1 + \alpha_2\mathbf{A}_2 + \ldots + \alpha_n\mathbf{A}_n,$$

where $\mathbf{A}_i$ *stands for the* i*-th column of* $\mathbf{A}$.

LEMMA 4.4.2. *For all matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *(with appropriate sizes for matrix multiplication), the multiplication can also be evaluated as follows:*

$$\mathbf{AB} = \mathbf{A}[\mathbf{B}_1|\mathbf{B}_2|\ldots|\mathbf{B}_n] = [\mathbf{AB}_1|\mathbf{AB}_2|...|\mathbf{AB}_n].$$

Proof of Theorem 4.3.1: We shall proceed by induction on the parameter k, which is the power of the matrix $\mathbf{H}$. For $k = 1$, the theorem holds since in $\mathbf{H}^1$, all columns have the same sum (equal to $2+\frac{1}{n}$), except for the leaves of the binary tree which are total of 2^{r-1} nodes with column sum equal to $1/n$.

We now assume that the theorem holds for $k = i$ and move forward to prove it for $k = i+1$. Using Lemmas 4.4.1 and 4.4.2, we have that

$$\begin{aligned}[\mathbf{H}^{k+1}]_{r,i} &= [\mathbf{H}^k \cdot \mathbf{H}]_{r,i} \\ &= [\mathbf{H}^k]_{r+1,2i-1} + [\mathbf{H}^l]_{r+1,2i} + 1/n[\mathbf{H}^k]_1\end{aligned}$$

Similarly,

$$\begin{aligned}[\mathbf{H}^{k+1}]_{r,j} &= [\mathbf{H}^k \cdot \mathbf{H}]_{r,j} \\ &= [\mathbf{H}^l]_{r+1,2j-1} + [\mathbf{H}^l]_{r+1,2j} + 1/n[\mathbf{H}^k]_1\end{aligned}$$

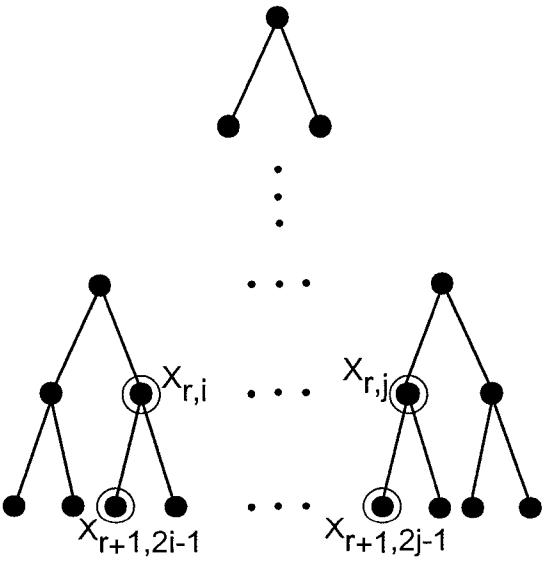

FIGURE 4.8. The location of some of the nodes used in the proof of Theorem 4.3.1.

However, if the nodes $x_{r,i}$ and $x_{r,j}$ are assumed to be on the same row (which is the case, since $1 \leq i \leq j \leq 2^{r-1}$ and the total number of nodes on the r-th row is 2^{r-1}), the nodes $x_{r+1,i}$, $x_{r+1,i+1}$, $x_{r+1,j}$ and $x_{r+1,j+1}$ are on the same row. See Figure 4.8.

Now, using the induction hypothesis, we have that:

$$[\mathbf{H}^k]_{r+1,2i-1} = [\mathbf{H}^k]_{r+1,2i} = [\mathbf{H}^l]_{r+1,j} = [\mathbf{H}^l]_{r+1,j+1}.$$

Hence,

$$[\mathbf{H}^{l+1}]_{r,i} = [\mathbf{H}^{l+1}]_{r,j}.$$

The final step of the induction is carried out and hence, for all $k \geq 1$,

$$[\mathbf{H}^k]_{r,i} = [\mathbf{H}^k]_{r,j}. \qquad \square$$

One interesting point to consider is that, not only are the above sums equal, but also the value for each element on row r, is approximately twice the value for the elements on row $r+1$. To verify this, we prove Theorem 4.3.2.

Proof of Theorem 4.3.2: Using the results of Theorem 4.3.1 we have that

$$(4.4)[\mathbf{H}^{k+1}]_{p,i} = [\mathbf{H}^k]_{p+1,2i-1} + [\mathbf{H}^k]_{p+1,2i} + 1/n[\mathbf{H}^i]_{1,1}.$$

But since both nodes $x_{p+1,2i-1}$ and $x_{p+1,2i}$ are located on the same row, they have equal column sums:

$$[\mathbf{H}^k]_{p+1,2i-1} = [\mathbf{H}^k]_{p+1,2i}.$$

Hence,

$$(4.5) \qquad [\mathbf{H}^{k+1}]_{p,i} = 2 \times [\mathbf{H}^k]_{p+1,2i} + 1/n[\mathbf{H}^i]_{1,1}.$$

By Theorem 4.3.1 applied to each row, all the nodes will have the same column sum value. By considering the equations (4.4) and (4.5), and not considering the effect of the root (which is carried as a constant all along the equations) it is seen that the column sum for the nodes on the pth row is approximately twice the column sum of each of the nodes on the $(p+1)$-th row. □

4.5. Conclusions for Binary Trees

In Sections 4.3, we calculated the stationary distribution of the uniform random walk on the binary tree. This vector, while not equal to the PageRank vector, is closely related to it. We proved in Theorems 4.3.1 and 4.3.2 that the values of the stationary distribution for the nodes on each row is the same and it reduces to approximately half for every row we move away from the root. This behaviour is suggestive of a power law degree distribution. On the other hand, all nodes of the binary tree have in-degree 2, except the leaves which have in-degree 0. The binary tree is, therefore, an example which shows that in-degree and the stationary distribution of the uniform random walk are not correlated. We conjecture that such a difference exists between in-degree and the PageRank distributions for binary trees.

4.6. PageRank of Random and Power Law Graphs

We provide some experimental results from simulations of both random digraphs and power law graphs. The results here corroborate the theoretical ones correlating PageRank

and in-degree in power law graphs described in previous sections of this chapter.

We first consider random digraphs. A *random digraph* has n nodes, and the probability of an edge between two distinct nodes occurs independently with probability $1/2$. See Figure 4.9 for a randomly sampled digraph with 100 nodes. It may be proven that as the number of nodes n tends to infinity, we have a binomial distribution for in-degree. This follows from the fact that the degree of a node is asymptotically concentrated on $n/2$. (See Theorem 3.11 in [**2**].)

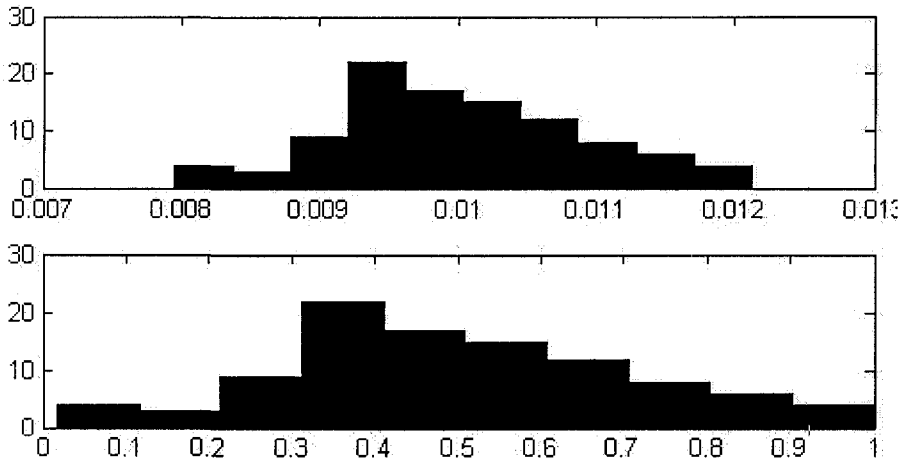

FIGURE 4.9. PageRank versus in-degree in a random digraph with 100 nodes.

However, the actual web graph has a power law, not binomial, degree distribution. We therefore include the PageRank distribution of a power law (undirected) digraph (produced using the freely available software Pajek). In the histogram in Figure 4.10, the distributions of PageRank and in-degree for a power law digraph with $1,200$ nodes is plotted. As is evident, both distributions follow similar power laws.

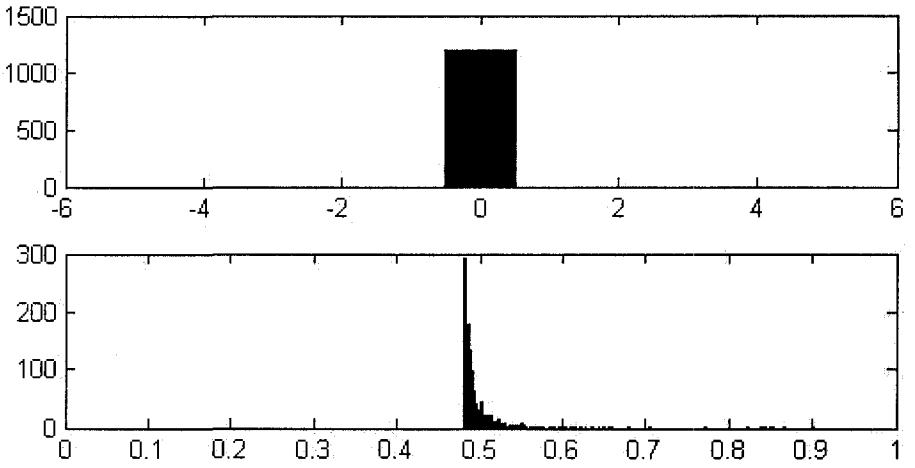

FIGURE 4.10. PageRank versus in-degree in a power law digraph with $1,200$ nodes.

CHAPTER 5

Conclusion and Future Work

We surveyed the mathematics of PageRank, which is a link-based ranking algorithm measuring the popularity of nodes in a digraph. The study of PageRank presented in this thesis combines graph theory, Markov chains, stochastic calculus, and statistics. In Chapter 2, we defined PageRank and summarized its key properties. We implemented a PageRank calculator in Matlab. With this calculator, we experimented with different graphs and compared their PageRank and in-degree distributions. In Chapter 3, we studied the recent work of Litvak et al. [**14**]. They proved (under certain assumptions) that in power law graphs, the PageRank and in-degree distributions follow power law distributions with the same exponent. In Chapter 4, we considered binary trees as a counterexample to the assertion that PageRank and in-degree possess similar distributions. The analysis of the PageRank of the class of binary trees is

significant, since it demonstrates that in general we cannot correlate PageRank with in-degree.

Several problems remain open related to the work described in this thesis. We list two few such problems here, which we will consider in the future.

(1) Derive the PageRank distribution of binary trees for all orders. We conjecture that such a distribution follows a power law, with PageRank decreasing as we move further from the root node. All nodes on the same row should have the same PageRank value. More generally, we would like to compute the PageRank of m-ary trees, where $m > 2$ (in these digraphs, all nodes except the leaves have constant in-degree equalling m).

(2) Livak et al. [**14**] made certain unrealistic assumptions in order to rigorously analyze PageRank using stochastic equations. For example, they assumed that all nodes have constant out-degree. They also assumed that the PageRank of nodes pointing to a similar page are independently distributed (in fact,

web pages that point to a similar page have correlated PageRank distributions). Can their analysis be generalized if these assumptions are removed?

Appendix

We include the original code used in the computational results in this thesis.

```
function M = MatRead (pjkInput)
fid = fopen(pjkInput);
A = fgetl(fid);
v = sscanf(A, '%*s %d');
for i = 1:v+1
    A = fgetl(fid);
end;
for i = 1:v
    i;
  L = fgetl(fid);
  M(i,:) = str2num(L);
end;
fclose(fid);
----------------------------
```

```
function ID =InDegree (X,n)
%Calculates the indegree or column sum
%of a random H-matrix of size n.

ID = sum(X);
% Now to normalize ID:
ID = (ID - mean(ID))/max(abs((ID-mean(ID))));
ID = (ID + 1)/2;
save ID;
end
----------------------------

function Inl = Initial (n)
%Generates the initial column matrix
%for the PageRank algorithm for size n.

Inl = zeros(1,n);
for i=1:n
    Inl(i)=1/n;
end%for
```

```
%return Inl;
save Inl;
-----------------------------
function J = RandomSample(n);
%Gives a random H-matrix of size n.

X = randint(n,n);
for j=1:n
    for i=1:n
        if i==j
            X(i,j)=0;
        end%if
    end%for
end%for
y = sum(X,2);
for j=1:n
    for i=1:n
        J(i,j)=X(i,j)/y(i);
    end%for
end%for
save J;
```

```
---------------------------

% Plots the histogram for both InDegree and

% PageRank for T_2(4)

hold on;

X=[0; 1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14];

PR_15_1 = [0.2755; 0.1402; 0.1402; 0.0648; 0.0648;

0.0648; 0.0648; 0.0231; 0.0231; 0.0231; 0.0231; 0.0231;

0.0231; 0.0231; 0.0231];

ID_15_1 = [2; 2; 2; 2; 2; 2; 2; 0; 0; 0; 0; 0; 0; 0; 0];

ID_15_2 = (1/49) * ID_15_1;

subplot(2,1,1); hist(PR_15_1);

subplot(2,1,2); hist(ID_15_2);
```

Bibliography

[1] L. Bechetti,C. Castillo, The distribution of PageRank follows a power law only for particular values of the damping factor, In *Proceedings of the 15th International Conference on World Wide Web*, 2006.

[2] A. Bonato, *A Course on the Web Graph*, Graduate Studies in Mathematics **69**, American Mathematical Society, Providence, Rhode Island, 2008.

[3] S. Brin, L. Page, Anatomy of a large-scale hypertextual web search engine, In: *Proceedings of the 7th International World Wide Web Conference*, 1998.

[4] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, J. Wiener, Graph structure in the web, *Computer Networks* **33** (2000) 309-320.

[5] F.R.K. Chung, L. Lu, *Complex Graphs and Networks*, American Mathematical Society, U.S.A., 2006.

[6] R. Diestel, *Graph Theory*, Springer-Verlag, New York, 2005.

[7] D. Donato, L. Laura, S. Leonardi, S. Millozi, Large scale properties of the web graph, *Eurapean Physical Journal* (2004) **38** 239-243.

[8] R. Durrett, *Random Graph Dynamics*, Cambridge University Press, New York, 2006.

[9] S. Fortunato, A. Flammini, F. Menczer, A. Vespignani, The egalitarian effect of search engines, Technical Report 0604203, arXiv/Physics, 2006.

[10] M.L. Goldstein, S.A. Morris, G.G. Yen, Problems with fitting to the power-law distribution, *Eurapean Physical Journal* **41** (2004) 255-258.

[11] G.R. Grimmett, D.R. Stirzaker, *Probability and Random Processes*, Oxford University Press, Oxford, 2001.

[12] O. Hernandez-Lerma, J.B. Lassere, *Markov chains and Invariant Probabilities*, Birkhauser, Berlin, 2000.

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

[14] N. Litvak, Y. Volkovich, W. Scheinhardt, In-degree and PageRank of web pages: why do they follow similar power laws?, In: *Proceedings of the 4th Workshop on Algorithms and Models for the Web-Graph*, 2006.

[15] C.D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, 2000.

[16] M. Mitzenmacher, E. Upfal, *Probability and Computing*, Cambridge University Press, New York, 2005.

[17] C. Moler, The world's largest matrix computation, *Matlab News and Notes*, October 2002, pp. 12-13.

[18] M.E.J. Newman, Power laws, Pareto distributions and Zipf's law, *Contemporary Physics* **46** (2005) 323-351.

[19] E. Seneta, *Non-negative Matrices and Markov chains*, Springer, New York, 2006.