UPPSALA
UNIVERSITET

U.U.D.M. Project Report 2012:14

# The Google Markov Chain: convergence speed and eigenvalues

Fredrik Backåker

Examensarbete i matematik, 15 hp
Handledare och examinator: Jakob Björnberg
Juni 2012

Department of Mathematics
Uppsala University

## Acknowledgments

I would like to thank my supervisor Jakob Björnberg for helping me writing this thesis.

# The Google Markov Chain: convergence speed and eigenvalues

## Contents

# 1 Introduction

There are many different search engines on the internet which help us find the information we want. These search engines use different methods to rank pages and display them to us in a way such that the most relevant and important information is showed first. In this thesis, we study a mathematical method that is a part of how PageRank,the ranking method for the search engine Google, ranks the order of which pages are displayed in a search. This method we look at uses pages as states in a stochastic Markov chain where outgoing links from pages are the transitions and the corresponding transition probabilities are equally divided among the number of outgoing links from the related page. The transition probability matrix that is given by this is then used to compute a stationary distribution where the page with the largest stationary value is ranked first, the page with the second largest is ranked second and so on. This method can be put into two variants, with a dampening factor or without. The variant without a dampening factor is the one we just described. In the other variant, which we study in this thesis, the dampening factor (often set to 0.85) is introduced mainly to ensure that the stationary distribution is unique. This variant is considered to be the most useful one and in this thesis we take a light look at how the dampening factor affects the computation of PageRank.
We will begin by going through some basic definitions for Markov chains and explain the Google PageRank in more detail. In the section after, we go through some general theory about the rate of convergence for Markov chains since it turns out that the eigenvalues of a transition probability matrix is connected to the convergence speed to its steady state. Further, we look at the second largest eigenvalue of the Google Markov chain and its algebraic multiplicity, which are the main factors that affect the convergence rate of the chain. Next, we go through some results of how the second eigenvalue of the Google Markov chain is limited by the dampening factor and by this, makes the choice of the dampening factor very important. We end by doing some simulations to check how different properties of PageRank are affected by choices of the dampening factor and in particular, which value of the dampening factor that is most adapted for a fast convergence speed of the Google Markov chain.

# 2 Definitions and background

## 2.1 Markov chains

A discrete time Markov chain is a stochastic process $\{X_n\}$ with finite state space *S* that satisfies the Markov property:

$$P(X_n=x_n|X_0=x_0,\ldots,X_{n-1}=x_{n-1})=P(X_n=x_n|X_{n-1}=x_{n-1})$$

for all $x_0,\ldots,x_n\in S$ and $n\geqslant 1$. In other words, the next step of a Markov chain is independent of the past and only relies upon the most recent state.
The chain is called time-homogenous if the transition probabilities do not change over time, i.e. if for each $i, j\in S$, $p_{ij}=P(X_n=j|X_{n-1}=i)$ does not depend on n.
In this case the probabilities $p_{ij}$ are the Markov chains transition probabilities when moving from state *i* to state *j*. Also let $p_{ij}^{(m)}=P(X_{m+n}=j|X_n=i)$ denote the transition probabilities in *m* steps, *m*=0,1,2... . The probabilities can be collected in a transition probability matrix, here denoted by *P*:

$$P=\begin{pmatrix} p_{00} & p_{01} & \cdots \\ p_{10} & p_{11} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

This matrix is called a stochastic matrix if all of the row vectors in it sum to one: $\sum_j p_{ij}=1.$ The Markov chain is said to be *irreducible* if it is possible to reach each state *i* from any other state *j,* in any number of steps. More formally, if

$$P(X_n=j|X_0=i)>0 \text{ for some } n\geq 0 \quad \forall i, j$$

A state *i* has period *k* if any return to state *i* occurs in multiples of *k* steps:

$$k=\text{greatest common divisor of the set}\{n:P(X_n=i|X_0=i)>0\}$$

If all the states in a Markov chain has period one, it is said to be *aperiodic*, i.e. the greatest common divisor of the return time to any state from itself is one.
The following result is standard and we do not prove it.

**Proposition 1**
A Markov chain that is irreducible and aperiodic with finite state space has a unique stationary distribution π, which is a probability vector such that $\pi=\pi P$. Additionally, the transition probabilities converges to a steady state when the number of steps goes to infinity in the sense that $\lim_{m\to\infty} p_{ij}^{(m)}=\pi_j$ for all *i,j* in *S*.

## 2.2 The Google PageRank

The Google PageRank is one of many methods that the search engine Google uses to determine the importance or relevance of a page. This method uses a special Markov chain which is used to compute the rank of web pages and this rank determines in which order the pages should be listed in a search in Google.

Let all the web pages Google communicates with be denoted by the state space *W*. The size of *W* is *n*, several billion pages. Let $C=(c_{ij})$ denote the connectivity matrix of *W*, which means that $C$ is a $n \times n$ matrix with $c_{ij}=1$ if there is an hyperlink from page *i* to page *j* and $c_{ij}=0$ otherwise. The number of outgoing links from page *i* are the row sums

$$s_i=\sum_{j=1}^{n} c_{ij}$$

If $s_i=0$ , it has no outgoing links and is called a dangling node. Let $T=(t_{ij})$ be given by $t_{ij}=c_{ij}/s_i$ if $s_i \geqslant 1$ and $t_{ij}=1/n$ if *i* is a dangling node. By this, $T$ can be seen as a transition probability matrix of the Markov chain with state space *W*. Furthermore, to define the Google Markov chain we include an additional parameter *d*, which is a dampening factor that can be set between 0 and 1. The transition probability matrix of the Google Markov chain is defined by:

$$P=dT+(1-d)(\frac{1}{n})\boldsymbol{E}$$

where $\boldsymbol{E}$ is the $n \times n$ matrix with only ones. This Markov chain can be described as a ”random surfer” who, with probability *d*, clicks on an outgoing link on the current web page with equal probabilites or, if the page has no outgoing links, chooses another page at random in *W*. Also, with probability 1-*d*, the surfer jumps to a page at random among all the pages *n*. The Google Markov chain is finite, irreducible and also aperiodic depending on which value *d* has. If $d<1$, the chain is aperiodic since all its states have a probability to jump back to them self and therefor periods that are equal to one. If $d=1$, we get $P=T$ and that the periodicity and irreducibility is completely determined by the outgoing links from all of the pages. By this, it is possible that two pages only link to each other and create a subset with a periodicity of two. If so, the chain is neither aperiodic nor irreducible. Then there is no unique stationary distribution and because the chain stays in a subset, the limit distribution $\pi_j$ depends on the starting state. This would be the most realistic case considering how the internet is structured and is the main reason why the dampening factor *d* is introduced. Further *d* affects, as we will see, the convergence speed of the Google Markov chain. In the computation of PageRank, *d* is usually set to 0.85[1] and then the Google Markov chain is finite, irreducible and aperiodic. Hence by Proposition 1 there exist a unique stationary distribution $\pi$. This stationary distribution is used to rank all the pages in *W* by letting the page with the largest $\pi_i$ be ranked first ,and the second largest be ranked second, and so on until all we get a Google PageRank for all the pages.

One way of computing the Google PageRank is done by simulating the transitions until you reach a (approximate) steady state and according to Brin and Page[1], the creators of Google, ”a PageRank for 26 million web pages can be computed in a few hours on a medium size workstation”.

# 3 Convergence speed
## 3.1 General theory of convergence speed

Since the Google PageRank consists of many billion pages, one might would like to know how fast this can be computed. This can be done by determining how fast the transition probability matrix of the Google Markov chain converges to its steady state as in Proposition 1. To find this rate of convergence, we need to go through some definitions and theorems.

Let $A$ be a square stochastic matrix of dimension $m$, $\boldsymbol{w}$ be a non-zero vector and $\lambda$ a scalar such that

$$A\,\boldsymbol{w}=\lambda\,\boldsymbol{w}$$

which is equivalent to $(A-\lambda l)\,\boldsymbol{w}=0$ (where $l$ is the identity matrix). Then $\lambda$ is said to be the right eigenvalue of $A$ corresponding to the eigenvector $\boldsymbol{w}$. In words, an eigenvector of a matrix is a non-zero vector that remains paralell to the original vector after being multiplied by the matrix and the eigenvalue of that eigenvector is the factor of which it is scaled when multiplied by the matrix. Eigenvectors can either be left or right eigenvectors, but the most commonly used is the right as described above. We say that $\lambda$ is a left eigenvalue if $\boldsymbol{z}^{\boldsymbol{T}}A=\lambda\,\boldsymbol{z}^{\boldsymbol{T}}$, where $\boldsymbol{z}$ is a non-zero vector (the left eigenvector). Each left eigenvalue is a right eigenvalue and vice versa, because if $\lambda_L$ is a left eigenvalue then

$$\begin{aligned}
&\boldsymbol{z}^{\boldsymbol{T}}A=\lambda_L\boldsymbol{z}^{\boldsymbol{T}}\\
\Leftrightarrow&(\boldsymbol{z}^{\boldsymbol{T}}A)^T=\lambda_L\boldsymbol{z}\\
\Leftrightarrow&A^T\boldsymbol{z}=\lambda_L\boldsymbol{z}\\
\Leftrightarrow&(A^T-\lambda_L l)\,\boldsymbol{z}=0\\
\Leftrightarrow&0=det(A^T-\lambda_L l)\\
&=det(A-\lambda_L l)^T\\
&=det(A-\lambda_L l)
\end{aligned}$$

This shows that $\lambda_L$ is also a right eigenvalue.

**Theorem 1**
$\lambda=1$ is always an eigenvalue of a stochastic $m\times m$ matrix $A$ associated with the right eigenvector $v=\mathbf{1}$ with all entries equal to 1. If a stationary distribution exist then the left eigenvector $u=\pi$.

*Proof:* Since $A\mathbf{1}=\mathbf{1}$ and $\pi A=\pi$. □

Let $\lambda_1,...,\lambda_m$ be the $m$ eigenvalues of $A$, assume these eigenvalues are distinct, and let $u_1,...,u_m$ be the corresponding left eigenvectors and $v_1,...,v_m$ be the corresponding right eigenvectors. We do not prove the following well-known fact.

**Theorem 2**
Let $A$ be an irreducible, aperiodic and stochastic $m\times m$ matrix, then $\lambda_1=1$ satisfies $\lambda_1>|\lambda_i|$ for any other eigenvalue $\lambda_i$.

We now perform some calculation that illustrate the relevance of the second largest eigenvalue for convergence speed.

**Proposition 2**
$u_1,...,u_m$ form an orthogonal set of vectors, and so do $v_1,...,v_m$.

*Proof of Proposition 2*: The equations for the eigenvectors are: $u_i^T A = \lambda_i u_i^T$ and $A v_j = \lambda_i v_j$. By multiplication we find that

$$\begin{aligned} u_i^T A v_j = \lambda_i u_i^T v_j &\Leftrightarrow u_i^T A v_j = \lambda_j u_i^T v_j \\ &\Rightarrow \lambda_i u_i^T v_j = \lambda_j u_i^T v_j \\ &\Rightarrow (\lambda_i - \lambda_j) u_i^T v_j = 0 \\ &\Rightarrow u_i^T v_j = 0 \text{ if } \lambda_i \neq \lambda_j \end{aligned}$$

and since the eigenvalues are distinct the following equation holds:

$$u_i^T v_j = 0, \text{ if } i \neq j, \quad 1 \leqslant i, j \leqslant m. \tag{1}$$

by this we see that eigenvectors of different eigenvalues are orthogonal to each other.

□

Further, we can scale the eigenvectors so that

$$u_i^T v_i = 1 \text{ for all } i \in [1, m]. \tag{2}$$

Collect the left eigenvectors $u_i$ of $A$ in $U$ so that $u_1$ is the first column in $U$, $u_2$ is the second, and so on. Collect the right eigenvectors $v_i$ in $V$ the same way.

$$U = (u_1, u_2, \ldots, u_m), V = (v_1, v_2, \ldots, v_m)$$

From (1) and (2) we get that:

$$U_i^T V_i = 1, \tag{3}$$

and also, from the theory of matrices, that $V_i U_i^T = 1$. Further, let $\Lambda$ be a diagonal matrix with the eigenvalues of $A$ as entries, i.e.

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \lambda_m \end{pmatrix}$$

Since $V$ consists of the right eigenvectors, we get the equation:

$$AV = V\Lambda \tag{4}$$

By (3) and (4) we get:

$$U^T AV = U^T V\Lambda = \Lambda$$

Which can be rewriten to:

$$A = V\Lambda U^T = \sum_{i=1}^{m} \lambda_i v_i u_i^T$$

We then take the power of $n$ of $A$ to get

$$A^n = V\Lambda^n U^T$$

Since,

$$\begin{aligned} A^n &= \underbrace{(V\Lambda U^T)\cdot(V\Lambda U^T)\cdots(V\Lambda U^T)\cdot(V\Lambda U^T)}_{n \text{ factors}} \\ &= V \underbrace{\Lambda(U^T V)\cdot\Lambda(U^T V)\cdots\Lambda(U^T V)\cdot\Lambda}_{n \text{ factors}} U^T \\ &= V\Lambda^n U^T \end{aligned}$$

By this we get the spectral decomposition $A^n = \sum_{i=1}^{m} \lambda_i^n v_i u_i^T$

We can rewrite this as $\left|A^n-\lambda_1^n v_1 u_1^T\right|=\left|\sum_{i=2}^{m}\lambda_i^n v_i u_i^T\right|\leqslant\sum_{i=2}^{m}\left|\lambda_i^n\right|\left|v_i\right|\left|u_i^T\right|$

Further let the eigenvalues other than $\lambda_1$, i.e. $\lambda_2,\lambda_3,...,\lambda_m$, be arranged such that $\lambda_1>\left|\lambda_2\right|\geqslant\ldots\geqslant\left|\lambda_m\right|$

The results above is an argument which shows that $\lambda_2$ is related to the difference of $\left|A^n-\lambda_1^n v_1 u_1^T\right|$ when all eigenvalues are different. In fact[4], there is even a way to show that also if the eigenvalues are not distinct then (*) holds.

Ensure (by rearranging the eigenvalues if necessary) that, if any $|\lambda_i|$ for $i\geq 3$ is equal to $|\lambda_2|$, then $r_i$, the algebraic multiplicity of $\lambda_i$, is less than or equal to $r_2$.

This arrangement of the eigenvalues is necessary for the Perron-Frobenius Theorem, where the algebraic multiplicity of the second eigenvalue is related to the convergence speed for a transition probability matrix to its steady state.

We state the theorem and then look at an example to make the rate of convergence and the algebraic multiplicity more clear.

**Theorem 3**

Let the eigenvectors be chosen so that $u_i^T v_i=1$, where $u_i$ is the left eigenvector and $v_i$ is the right eigenvector. Then we get the formula:

$$A^n=\lambda_1^n v_1 u_1^T+O\left(n^{r_2-1}\left|\lambda_2\right|^n\right). \qquad (*)$$

**Example 1** *Rates of convergence via Perron-Frobenius Theorem*

If $P$ is a stochastic, irreducible and aperiodic matrix with state space $S=\{1,...,m\}$. Then the first eigenvalue is $\lambda_1=1$ with eigenvectors $v_1=\mathbf{1}$, $u_1=\pi$, and therefore by (*)

$$P^n=\mathbf{1}\pi^T+O\left(n^{r_2-1}\left|\lambda_2\right|^n\right)$$

and if the eigenvalues are distinct and the absolute value of these are different we even get

$$P^n=\mathbf{1}\pi^T+O\left(\left|\lambda_2\right|^n\right).$$

By this we see that smaller $|\lambda_2|$ gives a higher rate of convergence.

If we do not arrange the eigenvalues and count their multiplicity, as described above, we might get a convergence speed that is not true. For example if the eigenvalues were equal to 0.5 and -0.5, and say that their algebraic multiplicity are 4 respectively 1. Then, since we choose the eigenvalue with the largest multiplicity, we get from theorem 2 that

$$P^n=\mathbf{1}\pi^T+O\left(n^{4-1}\left|0.5\right|^n\right)$$

If we ordered the eigenvalues so that $\lambda_2=-0.5$ instead, we get

$$P^n=\mathbf{1}\pi^T+O\left(n^{1-1}\left|-0.5\right|^n\right)$$

which is not true for the rate of convergence.

## 3.2 Convergence speed and eigenvalues of Google´s Markov Chain

Now that we know the importance of the second eigenvalue we can continue by looking at some results done by Haveliwala and Kamvar[2]. These results are related to the second eigenvalue of the Google Markov chain transition probability matrix and to the dampening factor $d$, which we will see is relevant to the result in Theorem 3.

**Theorem 4**

The Google Markov chains transition probability matrix, $P=dT+(1-d)\left(\frac{1}{n}\right)\boldsymbol{E}$, has a second eigenvalue that satisfies $\left|\lambda_2\right|\leqslant d$

**Theorem 5**

If there exist at least two irreducible closed subsets in $T$, then the second eigenvalue of $P$ is $\lambda_2=d$

In [2], the result is stated and proved for any rank-one row-stochastic matrix $\boldsymbol{E}$, but in our case we only consider when $\boldsymbol{E}$ is the $m\times m$ matrix with only ones. In applications the most common choice is to use the $\boldsymbol{E}$ we have chosen. Upon reading [2] we found that the proofs of their results became considerably simpler in our case and we therefore now give the proofs for these simpler cases in detail. Proofs of both these theorems rely on the following lemma:

**Lemma 1**
If $v\neq\mathbf{1}$ is a right eigenvector of $P$ then $v$ is a right eigenvector of $T$.
If $Pv=\lambda v$ then $Tv=\frac{\lambda}{d}v$.

*Proof*: Let $\lambda_i$ denote the eigenvalues of $P$ with associated eigenvector $v_i$, and let the eigenvalues of $T$ be denoted by $\mu_i$ with associated eigenvector $w_i$ for i=2,3,...,$m$.

If we look at the equation for the eigenvalues of $P$ we get:

$$P\,v_i=dT\,v_i+(1-d)(\frac{1}{m})\,\boldsymbol{E}\,v_i=\lambda_i v_i$$

Each row of $\boldsymbol{E}$ equals $\mathbf{1}^{\mathbf{T}}$. By proposition 2, since $\mathbf{1}$ is also the first right eigenvector of $P$, we get $\mathbf{1}^{\mathbf{T}}v_i=0$ and by this, that $\boldsymbol{E}v_i=0$. The equation left is:

$$dT\,v_i=\lambda_i v_i$$

Next, we can divide by $d$ to get:

$$T\,v_i=\frac{\lambda_i}{d}v_i$$

Then we let $w_i=v_i$ and $\mu_i=\lambda_i/d$. Rewrite the equation to

$$T\,w_i=\mu_i w_i$$

and we see that $v_i$ is an eigenvector of $T$ as well.

□

With this we can continue to the proofs of the theorems.

*Proof of theorem 4*: For this to be proven, we need to look at different values of $d$.
Let $\lambda_i$ denote the eigenvalues of P with associated eigenvector $v_i$, and let the eigenvalues of T be denoted by $\mu_i$ for i=2,3,...,$m$.

When $d$=0 we get the equation $P=(\frac{1}{m})\,\boldsymbol{E}$, which has $\lambda_2$=0 and therefore the theorem holds.
When $d$=1 we get the equation $P=T$ and since $T$ is stochastic, $|\lambda_2|\leqslant 1.$ , the theorem holds for this case as well.
To prove the theorem for the case when 0<d<1, we use the lemma we proved before.
From Lemma 1 we know that $T\,v_2=\frac{\lambda_2}{d}v_2$ and by this, we see that the eigenvalues of $P$ and $T$ is scaled by: $\lambda_2=d\mu_i$. Since $T$ is stochastic and therefore the eigenvalues $|\mu_i|\leqslant 1,$ we get that $|\lambda_2|\leqslant d$ and Theorem 4 is proved.

□

*Proof of theorem 5*: It is a standard fact (from page 126 of [3]) that the multiplicity of eigenvalue 1 of a stochastic matrix is equal to the number of irreducible closed subsets. So $\exists \geqslant 2$ linearly independent eigenvectors *a,b* of *T* with

$$Ta = a\,,\, Tb = b$$

We want to construct a vector *x* with $T\,x = x$ and $\mathbf{1}^T x = 0.$

Set $x = \alpha a + \beta b\,,$ α and β are scalars. By this we get $T\,x = \alpha Ta + \beta Tb = \alpha a + \beta b\,,$ so $T\,x = x$ for all α,β.

Further, we want $\mathbf{1}^T x = \alpha\,\mathbf{1}^T a + \beta \mathbf{1}^T b$ to be equal to 0. If $\mathbf{1}^T a = 0$ and $\mathbf{1}^T b = 0,$ we can choose any α,β. Otherwise, we assume $\mathbf{1}^T a \neq 0$ and choose $\alpha = -\beta \dfrac{\mathbf{1}^T b}{\mathbf{1}^T a}.$

By this, we see that *x* is an eigenvector of *T* with eigenvalue 1 and from lemma 1 we get that *x* is an eigenvector of *P* with eigenvalue $\dfrac{\lambda_2}{d} = 1 \Leftrightarrow \lambda_2 = d\,.$

□

Since *T* most likely has at least two irreducible closed subsets (as mentioned in section 2.2 when two pages only link to each other, also see figure 1 below) it is obvious that the choice of the dampening factor is related to the rate of which the Google Markov chain converges.

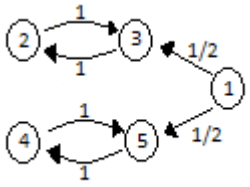


***Figure 1****: 2 and 3creates an irreducible closed subset, and so do 4 and 5.*

To make these results on the second eigenvalue more clear we look at some small examples.

**Example 2**

Consider a small web of 3 pages with the transition probability matrix

$$T = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/2 & 0 & 1/2 \\ 1/2 & 1/2 & 0 \end{pmatrix}$$

and a dampening factor set to 0.85. Then the transition probability matrix of the Google Markov chain in this case is $P = 0.85 \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/2 & 0 & 1/2 \\ 1/2 & 1/2 & 0 \end{pmatrix} + \dfrac{0.15}{3} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1/20 & 19/40 & 19/40 \\ 19/40 & 1/20 & 19/40 \\ 19/40 & 19/40 & 1/20 \end{pmatrix}$

The eigenvalues of *P* are computed by $det[P - \lambda I_3] = 0.$ We then get from using matlab that the second eigenvalue is $\lambda_2 = -0.4250$ and according to Theorem 4 $|\lambda_2| \leqslant d\,,$ which is true since $|-0.4250| < 0.85\,.$

**Example 3**

Consider another small web with 3 pages where the transition probability matrix is

$$T = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and the dampening factor is set to 0.85. In this case we get that $P$ is

$$P=\begin{pmatrix} 1/20 & 19/40 & 19/40 \\ 1/20 & 9/10 & 1/20 \\ 1/20 & 1/20 & 9/10 \end{pmatrix}$$

Since $T$ has at least two irreducible closed subsets, the second eigenvalue must be equal to dampening factor. From computation we get that $\lambda_2=0.85$ and we see that the result from Theorem 5 is true.

# 4 Simulations

Now that we know, from Theorem 4 and 5, that our choice of $d$ will affect $\lambda_2$ and further the convergence rate of the Google Markov chain, it is of interest to simulate some random networks in Matlab to investigate how different values of $d$ affect some factors of the computation of PageRank. We begin by giving a short description of how our simulation of random networks is done and then move on to our results.

As we mentioned earlier the internet is structured in a way so that the pages link to each other and create more than one closed irreducible subset. Therefore, we consider this in our simulation of networks to make them somewhat realistic and also remove links which link to the page them self to prevent single pages from creating subsets. To do this, we randomize small transition probability matrices of desired size and then put them into the diagonal in another bigger matrix which becomes our $T$. For example, if we randomize three smaller matrices $A,B$ and $C$, it looks like

$$T=\begin{pmatrix} A & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & C \end{pmatrix}$$

(the 0's represent matrices of zeros of the same size as the small matrices). Then if a page links to itself we remove that link by putting a zero there instead. In other words, the diagonal of $T$ contains only zeros. After we have done this, if a page in $T$ does not have a link we randomize a page to link to other than the page itself. We then use this modified $T$ in our Google Markov chain $P$ to do some tests.

## 4.1 Multiplicity of the second eigenvalue

The first thing we tested was how $r_2$, the multiplicity of $\lambda_2$, was affected by different values of $d$. Recall (*). Which shows that apart from $|\lambda_2|$ itself, the main factor affecting the convergence speed is $r_2$. If we choose a small $d$, we would limit $\lambda_2$ to also being small and by this get a high convergence rate depending on the size of $r_2$. But if $r_2$ is bigger for smaller $d$ it would lower the rate of convergence and then the choice of $d$ would not give us the convergence speed as desired. When testing this we found that different $d$ does not affect the multiplicity of $\lambda_2$ and therefore a choice of $d$ close to 0 would give a fast convergence speed. For example, in a test with $d$=0.85 for a network of 1000 pages we got $r_2$=3 and when changing $d$ to 0.1 and 0.01 for the same matrix we got that the multiplicity still was 3.(the x-axis shows the multiplicity of the second eigenvalue and the y-axis is the number of simulations)

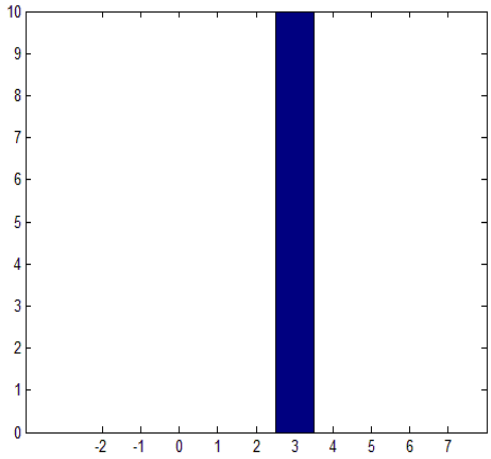


***Figure 2:*** *d=0.1*

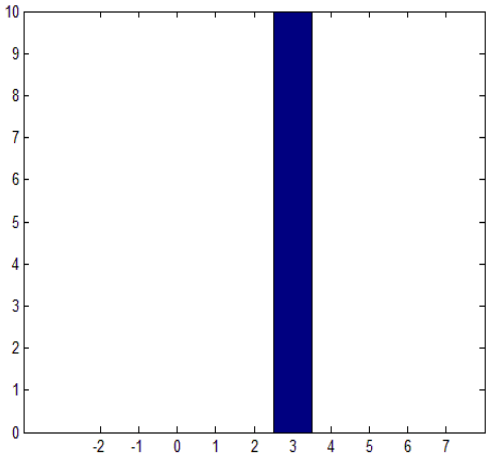


***Figure 3:*** *d=0.01*

## 4.2 Quality of the limit distribution

The next question would then be why *d* should not be set to a very small number. After all, smaller *d* mean faster convergence. In the extreme case *d*=0, convergence would be almost immediate. However, since then $P=(\frac{1}{m})\boldsymbol{E}$ the limit distribution is always uniform regardless of the structure of the internet itself. Intuitively, larger *d* means that *P* is "closer" to *T* and hence the PageRank should give a more realistic picture. We have not found any quantitative formulation of this intuition in the literature, and have therefore simulated to obtain the PageRank for different *d.* In these tests we simulate transitions for networks of size 1000 for different *d*. The following figures shows the number of times each state in *P* is visited when simulating 10 000 transitions and choosing a starting state at random, (the x-axis shows the states and the y-axis shows the number of visits).

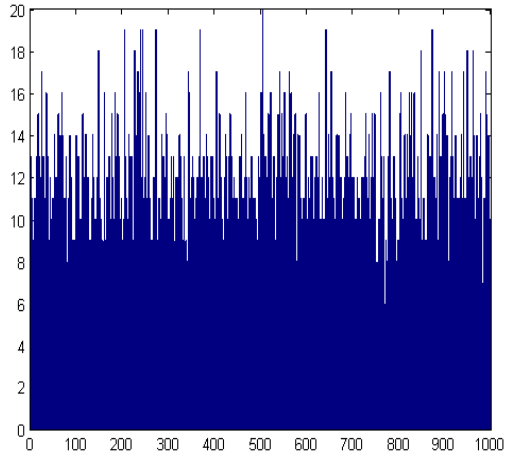


***Figure 3:*** *d=0.1*

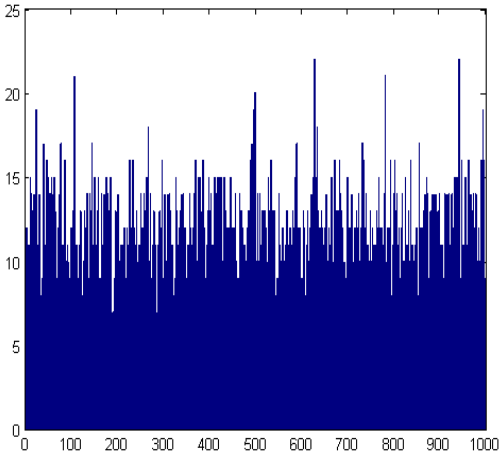


***Figure 4:*** *d=0.5*

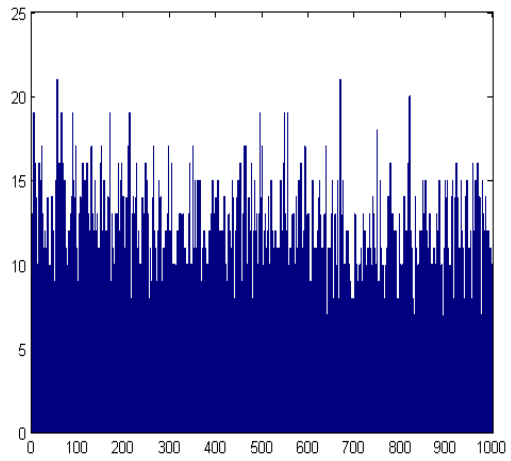


***Figure 5:*** *d=0.85*

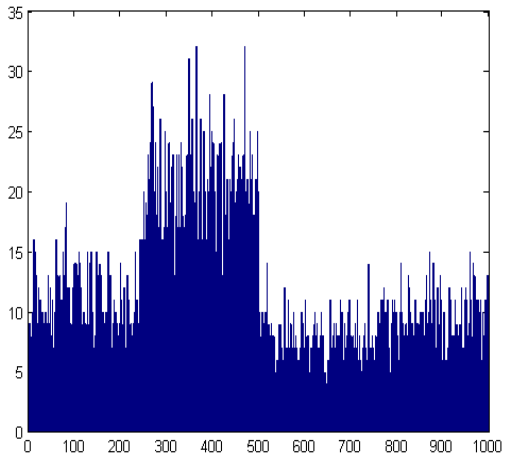


***Figure 6:*** *d=0.99*

These visits are then counted and used to compute a steady state for *P* and in turn the PageRank of the pages. From the figures, we see that the number of times a state is reached are evenly spread for lower values of *d* and thereby gives a more uniform steady state than for higher *d.* The PageRank for low *d* would then be calculated generally from $(\frac{1}{m})\boldsymbol{E}$ and most of the structure from *T* would be lost. To investigate this further, we ran a test with a specified *T* (see figure 7).

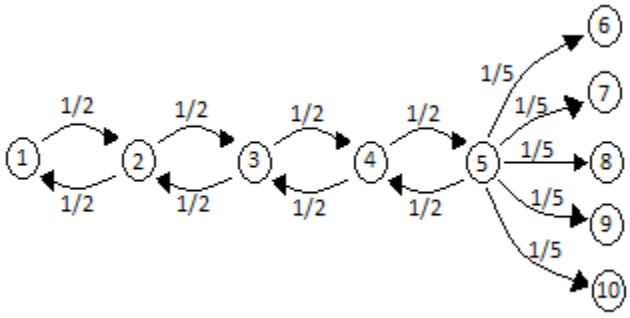


***Figure 7:*** *States 6,7,8,9 and 10 have links to each other(not illustrated) and create an irreducible subset.*

This time, we simulated 1000 transitions for different values of *d,* and did this 1000 times for each *d* to get the mean value for the distance between uniform distribution and our simulated stationary distribution.

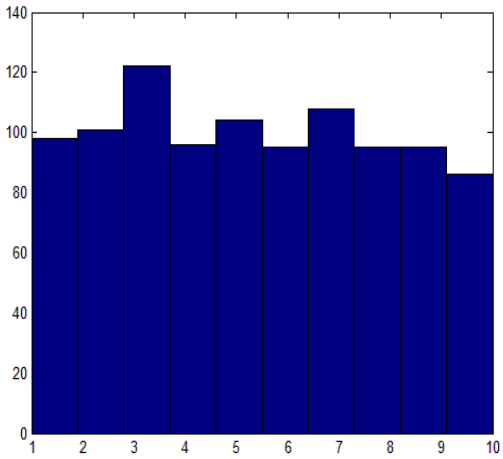


***Figure 8:*** *d=0.1*

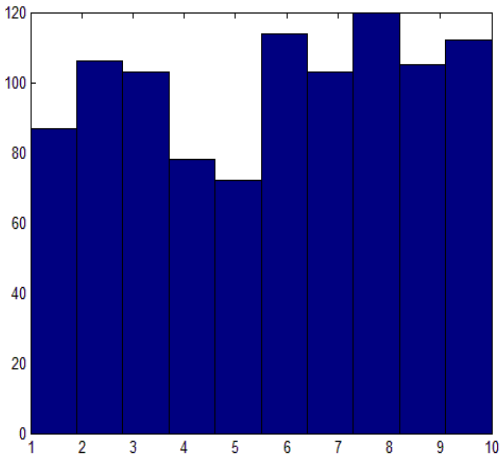


***Figure 9:*** *d=0.5*

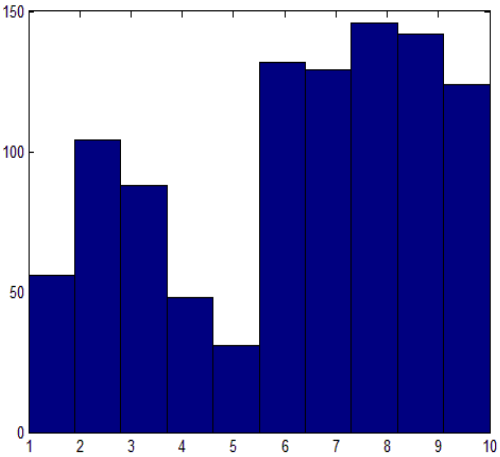


***Figure 10:*** *d=0.85*

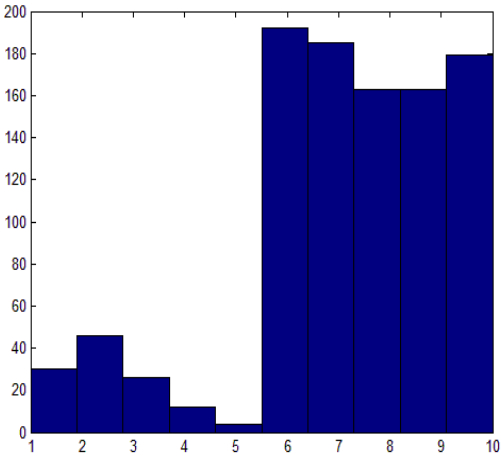


***Figure 11:*** *d=0.99*

In the figures above we see the number of visits for a simulation for each value of *d* we tested and this time it is even more clear that lower values of *d* gives a more uniform steady state. The distances we measured between uniform distribution and our steady state for these *d* shows that our statement is correct:

| Value of *d* | 0.1 | 0.5 | 0.85 | 0.99 |
|---|---|---|---|---|
| Distance: $\sqrt{\sum_{i=1}^{m}\left(\pi_i-\frac{1}{m}\right)^2}$ | 0.0028 | 0.0030 | 0.0053 | 0.0679 |

Smaller values of *d* give shorter distance to the uniform distribution and to make this obvious, we plotted the distance from simulations for values of *d* between 0.01 and 0.99 (the x-axis shows the value of *d* and the y-axis shows the distance).

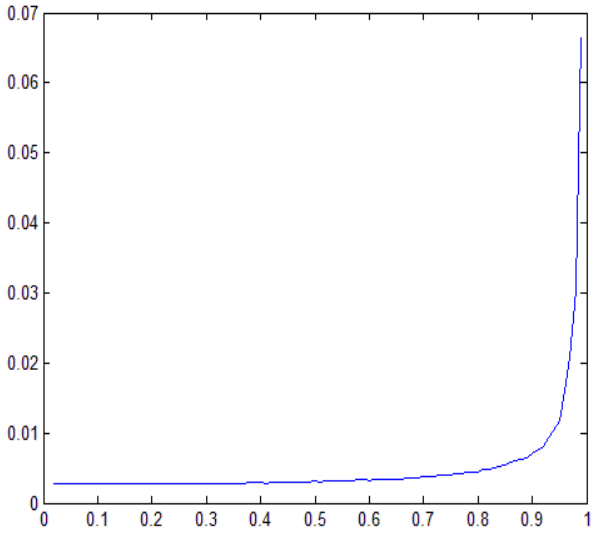


***Figure 12:*** The distance for different values of *d*

# 5 Conclusion

From the results in section 3, we have learned that the Google Markov chains second largest eigenvalue and the algebraic multiplicity of it, is directly affecting the convergence speed of the chain to a stationary distribution. Furthermore we have seen that the dampening factor restrains the second largest eigenvalue to be less than or equal to the value of the dampening factor or to be equal to the dampening factor in the case when there exists two or more closed irreducible subsets in the set of pages we use. In section 4 we see, from different simulations of networks, that different values of the dampening factor does not change the multiplicity of the second eigenvalue and therefore a choice of a very small dampening factor would give faster convergence speed than larger choices, such as 0.85. But from tests of the quality of the limit distribution, we discover that setting the dampening factor to a low value will change the structure of the transition probability matrix of the Google Markov chain and transform the limit distribution into being a uniform distribution. Since outgoing links from pages are the main factor in this method of computing a Google PageRank, a stationary distribution which is uniform would mean that all of the pages have the same PageRank. By this, we would have received a fast computation of a PageRank where we have lost almost all of the information from our original network and this would not be very useful. From these results we see that setting the dampening factor to 0.85, as the creators of Google did, might give us a good combination of good quality of the limit distribution and fast convergence speed for the Google Markov chain.

# 6 References

[1] L. Page and S. Brin. “The Anatomy of a Large-Scale Hypertextual Web Search Engine”. Computer Networks and ISDN Systems 30 (1998) 107- 117
[2] T.H. Haveliwala and S.D. Kamvar, “The second eigenvalue of the Google matrix”, Stanford University, Computer Science Department.
[3] D.L. Isaacson and R.W. Madsen. “Markov Chains: Theory and applications”, chapter IV, pages 126-127. John Wiley and sons, Inc. New York, 1976.
[4] P.Brémaud. “Markov Chains: Gibbs fields, Monte Carlo simulation and Queues”, chapter 6, pages 195-199. Springer-verlag, New York, 1999.
[5] G.R. Grimmett, D.R. Stirzaker, “Probability and random processes”, 2nd Ed, Oxford Science Publications, Oxford, 1992.

# Matlab-Code

```
% Program for determining the multiplicity of the second eigenvalue.
% n is the number of smaller matrices put into the diagonal in a bigger matrix.
% p is prob of a link
% d is the dampening factor
% loop is the number simulations

function [P] = randNet(n,p,d,loop)

TEST=zeros(1,loop);

for k=1:loop;


N=4*n;
T=zeros(N);
for i=1:4
    D=rand(n);
    D=(D<=p*ones(n));
    a=(i-1)*n+1;
    b=a+n-1;
    T(a:b,a:b)=D;
end

for i=1:N
    T(i,i)=0;
    if (sum(T(i,:))==0)
        j=randi([1,N-1]);
        if j<i
            T(i,j)=1;
        else
            T(i,j+1)=1;
        end
    end
end


T=T./repmat(sum(T,2),1,4*n);



P=d*T+(1-d)/(4*n)*ones(4*n);

Eig=eig(P);
 RES = zeros(size(eig(P)));
for i = 1: size(eig(P))

    RES(i) =(abs(Eig(i)) > d-0.000001 );

end

TEST(1,k)=sum(RES)-1;
sum(RES)-1
end

hist(TEST)
```

```
% Program for determining Pagerank and distance to uniform distribution.
% n is the number of smaller matrices put into the diagonal in a bigger matrix.
% p is the probability of a link.
% d is the dampening factor.
% trans is the number of transitions.

function [P] = Pagerank(n,p,d,trans)

N=4*n;
T=zeros(N);
for i=1:4
    D=rand(n);
    D=(D<=p*ones(n));
    a=(i-1)*n+1;
    b=a+n-1;
    T(a:b,a:b)=D;
end

for i=1:N
    T(i,i)=0;
    if (sum(T(i,:))==0)
        j=randi([1,N-1]);
        if j<i
            T(i,j)=1;
        else
            T(i,j+1)=1;
        end
    end
end


T=T./repmat(sum(T,2),1,4*n);

P=d * T + (1-d)/(4*n)* ones(4*n);

nmb=trans;

states=zeros(1,nmb);
states(1)=randi([1,4*n]);


levels=cumsum(P,2);

pi=zeros(1,4*n);
count=0;
for i=1:nmb-1
 u=rand;
 j=1;

 while u>levels(states(i),j)
  j=j+1;
 end;
 states(i+1)=j;
 for k=1:4*n
 if states(i)==k
     count=count+1;
 pi(k)=count;
 end
 end;
end;
```

```
pi=pi/sum(pi);
unif=ones(1,N)/N;
norm(pi-unif)

hist(states(1000:nmb),4*n)

% Program for determining Pagerank and distance to uniform distribution for a
specified Markov chain.

% d is the dampening factor.
% trans is the number of transitions.
% loop is the number simulations

function [P] = Pagerank3(d,trans,loop)

T=[0   1   0   0   0   0   0   0   0   0;
   1   0   1   0   0   0   0   0   0   0;
   0   1   0   1   0   0   0   0   0   0;
   0   0   1   0   1   0   0   0   0   0;
   0   0   0   0   0   1   1   1   1   1;
   0   0   0   0   0   0   1   1   1   1;
   0   0   0   0   0   1   0   1   1   1;
   0   0   0   0   0   1   1   0   1   1;
   0   0   0   0   0   1   1   1   0   1;
   0   0   0   0   0   1   1   1   1   0];


T=T./repmat(sum(T,2),1,10);


TEST=zeros(1,loop);
dampen=zeros(1,99);
dval=zeros(1,99);
for m=1:98;
    P=d * T + (1-d)/(10)* ones(10);
for l=1:loop;

nmb=trans;

states=zeros(1,nmb);
states(1)=1;



levels=cumsum(P,2);

pi=zeros(1,10);
count=0;
for i=1:nmb-1
 u=rand;
 j=1;

 while u>levels(states(i),j)
  j=j+1;
 end;
 states(i+1)=j;
 for k=1:10
 if states(i)==k
     count=count+1;
 pi(k)=count;
```

```
 end
 end;
end;

pi=pi/sum(pi);
unif=ones(1,10)/10;
TEST(1,l)=norm(pi-unif);
end;

dampen(1,m)=mean(TEST);

dval(1,m)=d;
d=d-0.01
end
plot(dval(1:99),dampen(1:99))
```