

Implementing Quotient Rayleigh in Power Method to Improve the Computation Speed of Pagerank

Daniel Siahaan and Zainal Arifin

Department of Informatics, Institute of Technology Sepuluh, Surabaya, Indonesia

Abstract: PageRank is one of popular web page ranking mechanisms, which is used by Google. It works based on link analysis. The computation of existing pagerank consumes a very significant time, i.e., in the scale of days, due to the fact that it has to find Eigen values of billion of web pages off-line. Previous attempts on reducing the computation time by shorting the time for Eigen value convergence have been carried out based on several methods, such as extrapolation, sparse linear system and quadratic extrapolation. This study proposes a new approach, i.e., using Quotient Rayleigh in the power method in order to shorten the convergence of dominant Eigen values. This modified power method resulted in a significant improvement on PageRank performance. The interpolation over n-size web pages also shows a consistent performance of the proposed approach.

Key words: Eigen dominant, link analysis, page ranking pagerank, power method, quotient rayleigh

INTRODUCTION

Popular search engines, such as Google, Yahoo and Altavista have played an important role in the World Wide Web. Search engine has been a killing application for internet users. A world wide web without a search engine is unimaginable nowadays. The search engine allow users to find certain information in the gigantic hyperspace database as web faster than it can be imagine before. Search engine also ensures that it produces relevant and important query results, which are achieved through a democratic ranking system (Langville, 2005). Google as the top most search engine in the internet nowadays, states that it has crawled through >4.2 billion web pages. This is done using their popular PageRank method as its ranking system.

The calculation of page ranks using the PageRank method is currently done offline in Google. Crawler results are converted into Markov chains, with the size of $n \times n$. With around 4.2 billion pages handled by Google, this Markov chains is a gigantic chains. The Pagerank method calculates and ranks an eigen value of each web page. This decremented order of eigen values represents the relevance and importance of the query results. This calculation takes >3 days. With the increasing number of web pages, this calculation will tend to take longer in the near future.

This research proposes the use of Quotient Rayleigh to find the dominant eigen value, which improves the convergence time of Markov chains calculation. With the decreasing of convergence time, it is expected to reduce

the PageRank computation time to <3 days, which means that the PageRank method performance will be improved significantly.

MATERIALS AND METHODS

PageRank was first developed by Larry Page at Stanford University (Vise and Malseed, 2005). Later on, the research was joined by Sergey Brin on 1995 and continued as part of a research project about a new kind of search engine. This later produced a functional prototype, called Google. While, just one of many factors, which determine the ranking of Google search results, PageRank becomes the foundation of web search engines of Google since then. PageRank is a method that Google used to measure the relevance of a web page to the query term given by user. It is based on citation analysis founded by Garfield (1979) at the University of Pennsylvania. When all searching result factors, such as title, tag and key words have been used, then Google uses PageRank to shows the order of importance of pages that will be displayed as in the main search results.

In principle, search engine does the following processes when processing a user query:

- Finding pages that match the given key word (s)
- Ranking the pages locally based on traditional information retrieval methods, such as key word matching, keyword closeness, hypertext tags level, occurrence frequency, etc.

- Merging this result with the global ranking scores produced through PageRank as follows:

$$PR(A) = (1-d) + d \left(\frac{PR(T_1)}{C(T_1)} + \frac{PR(T_2)}{C(T_2)} + \dots + \frac{PR(T_n)}{C(T_n)} \right)$$

Where:

- PR(A) = Page rank of page (A)
- D = Damping factor (0.8)
- PR(T₁) = Page rank to page (T₁)
- C(T₁) = Number of link in page (T₁)
- PR(T_n)/C(T_n) = Ratio of pages have link to (A)

Markov chains: Determination process of PageRank begins by forming an n×n matrix, called hyperlink matrix A. Where, n is the number of web pages. If a web page i has link $i \geq 1$ to other web pages and web pages i has a link to webpage j, then element at line- i and column-j of matrix A (A_{ij}) = 1/i, while other element of A_i = 0. Therefore, A_{ij} represents the likelihood of a random surfer may choose a link from webpage i to webpage j. This matrix of link graphs is called a markov matrix or markov chains. In this research, the markov matrix is obtained from the dataset from previous researches, where markov matrix is stochastic and irreducible (Austin, 2006).

A markov chain is a line of random variable, X₁, X₂, X₃,.... given a markov property that is a state that currently happens, where past and current states are independent and unbounded.

$$Pr(X_{n+1}=x | X_n, \dots, X_1, X_0=x_0) = Pr(X_{n+1}=x | X_n=x_n)$$

A probability of moving from state i (page i) to state j (state j) in n steps (click) is defined as:

$$p_{ij}^{(n)} = Pr(X_n = j | X_0 = i)$$

and for 1-step transition can be defined as:

$$p_{ij} = Pr(X_1 = j | X_0 = i)$$

The markov matrix is calculated until it produces convergence of eigen values. A markov chain is called ergodic if there is a path from a state to another state, or if all states are aperiodic and positive recurrent, given that the probability is not zero.

For any markov chain that is ergodic, there is a unique average number of visitations for each state. It means that for a period of time, a number of visitations of each state is proportional regardless the original state of the visitation (Austin, 2006).

Eigen vector and eigen value: If A is a n×n matrix, then a nonzero vector x in Rⁿ is an eigen vector of A if Ax is a magnitude multiplication of x that is:

$$Ax = \lambda x \tag{1}$$

for any scalar λ . λ is eigen value of A and x is eigen vector that corresponds with λ . To find an eigen values of matrix A of size n×n, Eq. 1 can be rewritten as follows:

$$Ax = \lambda Ix \tag{2}$$

or equally written as:

$$(\lambda I - A)x = 0 \tag{3}$$

In order the scalar λ is considered as an eigen value, there should be a nonzero solution of this equation is:

$$\det(\lambda I - A) = 0$$

The scalar that fulfills this equation is eigen value of A. This is called a characteristic equation of A. Thus, $\det(\lambda I - A)$ is a polynomial of λ that is, a characteristic polynomial of A.

If A is matrix of size n×n, therefore the characteristic polynomial of A should be n and coefficient of λ^n are 1. Therefore, the characteristics polynomial of matrix n×n has the form of:

$$\det(\lambda I - A) = \lambda^n + c_1 \lambda^{n-1} + \dots + c_n$$

Power method: In practical problems, computing large matrix to find eigen values is a complex computation problem, that implies time and computation resource consuming.

In this research, we propose the use of a simpler algorithm to find the eigen values, that is, power method. Power method produces approximate eigen values given the highest absolute value of corresponding eigen vector.

An eigen value of matrix A is called dominant eigen value of A if its absolute value is bigger than the rest of absolute eigen values. The eigen vector that corresponds with the dominant eigen value is called dominant eigen vector of A.

Let assume A is matrix of size n×n that can be diagonalized by dominant eigen value. If x_0 is any non-zero vector in Rⁿ, then vector:

$$A^p x_0 \tag{4}$$

is usually a good approximation of dominant eigen vector if exponent p is big (Langville, 2005). Iteration process of such vector computation is called power method. Power method most of the time produce vector with large component.

To solve this, the approximation eigen vector is scaled down in each iteration so that the component will be positioned between +1 and -1. This can be achieved by multiplying the approximation eigen vector with its inverse component that has the biggest absolute value.

For example, given matrix A that has eigen value λ and x is correspond eigen vector. If $\langle \cdot, \cdot \rangle$ represents the result in Euclids, then:

$$\frac{\langle x, Ax \rangle}{\langle x, x \rangle} = \frac{\langle x, \lambda x \rangle}{\langle x, x \rangle} = \lambda \frac{\langle x, x \rangle}{\langle x, x \rangle} = \lambda$$

Therefore, if x is the approximation of dominant eigen value, then the dominant eigen value of λ_1 can be approximated by:

$$\lambda_1 = \lambda \frac{\langle \tilde{x}, A\tilde{x} \rangle}{\langle \tilde{x}, \tilde{x} \rangle}$$

which is called Quotient Rayleigh (Gumerov, 2003).

Eigen values estimation using power method: The proposed solution is focused on improving power method which is utilized for estimating eigen values during Pagerank calculation. In Fig. 1a (Brin and Page, 1998), the iteration will stop when the eigen value is convergence on corresponding eigen vector. While, in Fig. 1b, there is a filtering mechanism on the eigen value, so that the eigen value resulted from Quotient Rayleigh will move faster toward the convergence value. And this is proven.

The approximation of eigen value by power method uses a simple iteration, which continues until it reaches a value where, it is convergence. The value is called eigen value. Given a markov matrix A with the size of $n \times n$ with an n-size non-zero eigen vector, so that

$$Ax_i = \lambda_i x_i \text{ for } i = 1, 2, 3, \dots, n$$

Where:

$$|\lambda_1| > |\lambda_2| > |\lambda_3| > \dots > |\lambda_n|$$

if $u^{(0)}$ is any vector where

$$\|u^{(0)}\|_2 = 1$$

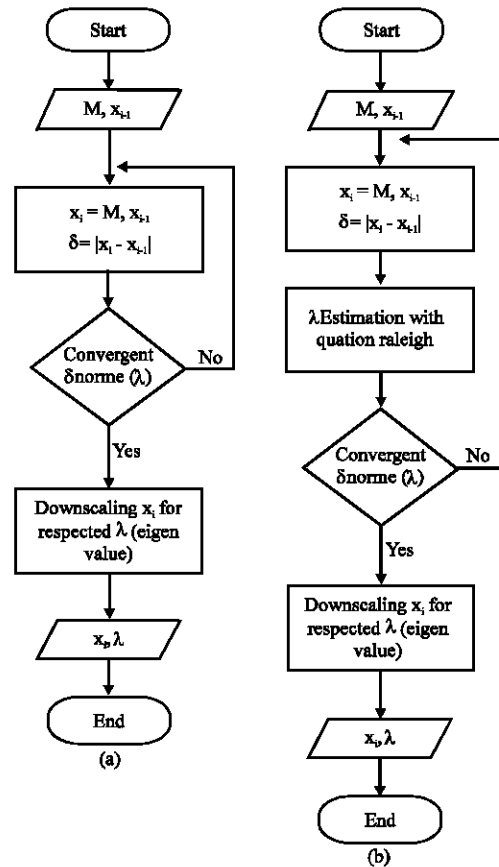


Fig. 1: Flowchart of eigen values estimation with power method, (a) existing method and (b) proposed method

Given a symmetrical matrix A, Eq. 4 can be computed as follows: While not convergence ($\text{norm}(v^{k+1}, u^k)$) do

$$v^{(k+1)} = Au^{(k)}$$

$$c_{k+1} = \frac{1}{\|v^{(k+1)}\|_2}$$

$$u^{(k+1)} = c_{k+1}v^{(k+1)}$$

End while.

It clearly shows that $u^{(k+1)} = c_{k+1} Au^{(k)}$, therefore:

$$u^{(1)} = c_1 Au^{(0)}$$

$$u^{(2)} = c_2 Au^{(1)} = c_2 c_1 A^2 u^{(0)}$$

$$\dots$$

$$u^{(k)} = c_k c_{k-1} \dots c_1 A^k u^{(0)}$$

$$= d_k A^k u^{(0)}$$

So that,

$$d_k = \prod_{j=1}^k c_j$$

Because eigen vector is unbounded-linear, then any vector can be made as a linear combination from eigen vector. Hence,

$$u^{(0)} = \sum_{i=1}^n \alpha_i x_i$$

then:

$$\begin{aligned} u^{(k)} &= d_k A^k \left(\sum_{i=1}^n \alpha_i x_i \right) \\ &= d_k \left(\sum_{i=1}^n \alpha_i A^k x_i \right) \\ &= d_k \left(\sum_{i=1}^n \alpha_i \lambda_i^k x_i \right) \end{aligned}$$

and because

$$\|u^{(k)}\|_2 = 1$$

then:

$$d_k = \frac{1}{\|A^k u^{(0)}\|_2}$$

If it is assumed that:

$$\begin{aligned} \lambda_1 &= \lambda_2 = \lambda_3 = \dots = \lambda_p \\ |\lambda_1| &> |\lambda_{i+p}|, \text{ for } i = 1, 2, 3, \dots, n-p \\ \sum_{i=1}^p |\alpha_i| &\neq 0 \end{aligned}$$

Then,

$$\begin{aligned} u^{(k)} &= \frac{\lambda_1^k \sum_{i=1}^p \alpha_i x_i + \sum_{j=p+1}^n \alpha_j \lambda_j^k x_j}{\left\| \lambda_1^k \sum_{i=1}^p \alpha_i x_i + \sum_{j=p+1}^n \alpha_j \lambda_j^k x_j \right\|_2} \\ u^{(k)} &= \frac{\left(\frac{\lambda_1}{|\lambda_1|}\right)^k \sum_{i=1}^p \alpha_i x_i + \sum_{j=p+1}^n \alpha_j \left(\frac{\lambda_j}{|\lambda_j|}\right)^k x_j}{\left\| \sum_{i=1}^p \alpha_i x_i + \sum_{j=p+1}^n \alpha_j \left(\frac{\lambda_j}{|\lambda_j|}\right)^k x_j \right\|_2} \end{aligned}$$

thus,

$$u^{(k)} = \gamma_k \sum_{i=1}^p \alpha_i x_i + o\left(\left|\frac{\lambda_{p+1}}{\lambda_1}\right|^k\right)$$

where, $|\gamma_k|$ tends to be constant even if $k \rightarrow \infty$. Every linear combination of the eigen vector that corresponds with the same eigen value is also also eigen vector.

Then, base on the Eq. 4, the eigen value computation algorithm using power method is scaled down using this mechanism. This will result in the order of x_0, x_1, x_2, \dots which, their approximation against dominant eigen vector is increasingly improved.

Eigen value estimation using modified power method: The motivation is to ensure that the convergence of eigen values can be reached faster than the previous method.

The eigen value approximation, i.e. dominant eigen value, by adopting power method with Quotient Rayleigh is the major issue in this research. Quotient Rayleigh is used because matrix A is symmetrical (Chowdury and Dasgupta, 2003).

Given any vector v and a value μ so that $\|Av - \mu v\|_2$ is minimum. By writing $k = v^H Av$ and assuming that $\|v\|^2$ (Wilkinson, 1965), we can conclude that:

$$\begin{aligned} \|Av - \mu v\|_2^2 &= (Av - \mu v)^H (Av - \mu v) \\ &= v^H A^H Av - k k^H + (\mu^H - k^H)(\mu - k) \end{aligned}$$

If it is observed that the minimum can be reached for $\mu = k = v^H Av$, the quantity is introduce as Quotient Rayleigh, which corresponding to v, which we written as μ_R . The word Quotient is used if and only if the scalar of v cannot be obtained, then it should be $v^H Av / v^H v$, it is clear that if:

$$Av - \mu_R v = \eta_R$$

therefore,

$$v^H \eta_R = 0$$

For any μ , it can be obtained:

$$Av - \mu v = \eta, (\|\eta\|_2 = \varepsilon)$$

Then,

$$Av - \mu_R v = \eta_R, (\|\eta_R\|_2 \leq \varepsilon)$$

Which is a circular disc with its center μ_R , which has a small radius corresponds to any given μ .

Given any normal matrix A, $Av - \mu_R v = \eta$, where $\|v\|_2 = 1$ and $\|\eta\|_2 = \varepsilon$. It is known that n-1 eigen values satisfy the relation: $|\lambda_i - \mu_R| \geq \alpha$, where, α is a constant. Then a relevant eigen values $\lambda_2, \lambda_3, \dots, \lambda_n$ can be obtained. If A is normal, then it has an eigen vector ortonormal system x_i , ($i = 1, \dots, n$), thus, it can be written as:

$$v = \sum_1^n \alpha_i x_i, 1 = \sum_1^n |\alpha_i|^2$$

Given the previous equation:

$$\eta = \sum_1^n \alpha_i (\lambda_i - \mu_R) x_i$$

so that:

$$\begin{aligned} \varepsilon^2 &= \sum_1^n |\alpha_i|^2 |\lambda_i - \mu_R|^2 \geq \sum_2^n |\alpha_i|^2 |\lambda_i - \mu_R|^2 \\ &\geq a^2 \sum_2^n |\alpha_i|^2 \end{aligned}$$

therefore, it can be deduced that:

$$|\alpha_1|^2 = 1 - \sum_2^n |\alpha_i|^2 \geq 1 - \varepsilon^2 / a^2$$

if θ is defined with the following relation:

$$\alpha_1 = |\alpha_1| e^{i\theta}$$

hence,

$$ve^{-i\theta} = |\alpha_1| x_1 + \sum_2^n \alpha_i e^{-i\theta} x_i$$

then:

$$\begin{aligned} \|ve^{-i\theta} - x_1\|_2^2 &= (|\alpha_1| - 1)^2 + \sum_2^n |\alpha_i|^2 \\ &\leq \frac{\varepsilon^4}{(1 + |\alpha_1|)^2} + \frac{\varepsilon^2}{a^2} \leq \frac{\varepsilon^4}{a^4} + \frac{\varepsilon^2}{a^2} \end{aligned}$$

Therefore, if $a > \varepsilon$, then $ve^{-i\theta}$ is the best approximation on x_1 . The constant factor $e^{-i\theta}$ is a less significant factor. As the value α tends to decreased, then the result will also become smaller. When α is equal to ε , then the result will be insignificant. Hence that μ_R is Quotient Rayleigh (Gumerov, 2003).

For any symmetrical matrix A, a temporary assumption is used on eigen matrix x_0 . Then, the iterative computation is conducted in order to obtain the approximation over the corresponding eigen vector with eigen value l . In each iteration a Quotient Rayleigh is used to know the maximum eigen value, so that it can be convergence.

While not convergence $abs(l_k - l_{k-1})$ do

$$\begin{aligned} v^{(k+1)} &= Au^{(k)} \\ c_{k+1} &= \frac{1}{\|v^{(k+1)}\|_2} \\ l_k &= \frac{\|v^{(k+1)}\|}{\|u^{(k)}\|} \end{aligned}$$

percent estimation of dominant eigen values

$$u^{(k+1)} = c_{k+1} v^{(k+1)}$$

End while.

RESULTS AND DISCUSSION

To measure the performance of the proposed power method, several tests have been conducted using a number variety of dataset. The result is also compared and analyzed against the existing power method to find the accuracy of the computation model in processing PageRank. This test also measures the efficiency of the proposed method in producing the dominant eigen values. To project the behavior of the system that uses the proposed model in n-size pages, we do an interpolation.

Experimental data: This experimentation is carried out by using dataset from previous researches. Table 1 explains the sources of the dataset used in the testing scenario.

Stanford University datasets were crawled in 2002 and used by Sepandar D. Kamvar in his research on ranking the search results of search engine. The datasets that were downloaded for the test are Stanford and Stanford Berkeley web matrixes. Stanford web matrix has 281, 903 pages or around 2.3 million paths with the size of 64.2 MB. While, Stanford Berkeley web matrix has 683, 446 pages or around 7.6 million paths with the size of 125.8 MB.

Toronto University datasets were crawled in 2004 and is used by a number of researches at Toronto University. There are 34 datasets that has been used for the testing, with the size starting from 742 until 11, 659 pages. These datasets was crawled based on the keyword searching and each has 3 separate files which also use for other researches related to path analysis.

Milano University datasets were crawled between 2000 and 2005. Datasets were crawled by many researches from various universities. It contains several datasets, starting from one that has 325, 577 pages until 1, 382, 909 pages. Table 2 shows the dataset from University Milano.

Result and analysis: The tests on the proposed method have been conducted at the Software Engineering Laboratory of ITS (Institut Teknologi Sepuluh Nopember). The tests were conducted using a PC Intel P4-3 Giga Quad Core with 5 GB memory. Table 3 shows the result of the experiment, where power method with Quotient Rayleigh

Table 1: Dataset for testing

Dataset	Location	No. of pages	Owner
Stanford Web Matrix, 2002	http://cs.stanford.edu	281, 903	Sepandar D. Kamvar
Stanford Berkeley Web Matrix, 2002	http://cs.stanford.edu	683, 446	Sepandar D. Kamvar
Crawler	http://cs.toronto.edu	3, 340 to 11, 659	Panayiotis Tsaparas
WebGraph	http://law.dsi.unimi.it	325, 557 to 118, 142, 155	Laboratory for web algorithmics

Table 2: Dataset from University Milano

Graph	Crawl year	Nodes	Arcs	Provider
Cnr-2000	2000	325,557	3,216,152	NagaokaUT
Eu-2005	2005	862,664	19,235,140	DSI
In-2004	2004	1,382,908	16,917,053	NagaokaUT

Tabel 3: Computation time of pagerank

Dataset		Page rank with existing power method		Page rank with modified power method		Efficiency (%)
Name	Pages	Iteration	Time (sec)	Iteration	Time (sec)	
Randomized algorithms	742	82	0.0309	75	0.0160	48.22
Computational complexity	1075	89	0.0160	78	0.0150	6.25
Automobile industries	1196	93	0.0320	77	0.0160	50.03
Table tennis	1948	78	0.0320	51	0.0160	50.00
Moon landing	2188	93	0.0470	77	0.0150	68.09
Computational geometry	2292	33	0.0160	30	0.0150	6.25
Affirmative action	2523	98	0.0470	77	0.0160	65.98
Net censorship	2598	93	0.0629	78	0.0159	74.72
Roswell	2790	96	0.0620	89	0.0320	48.39
Jaguar	2820	96	0.0619	43	0.0159	74.31
Gun control	2955	96	0.0779	73	0.0320	58.92
Classical guitar	3150	87	0.0780	81	0.0319	59.10
Armstrong	3225	89	0.0630	81	0.0310	50.79
Cheese	3266	91	0.0780	75	0.0309	60.38
Abortion	3340	95	0.0940	71	0.0310	67.02
Amusement parks	3410	90	0.0620	75	0.0310	50.00
Vintage cars	3460	88	0.0619	73	0.0309	50.08
Complexity	3564	83	0.0780	75	0.0309	60.38
Iraq war	3782	88	0.0779	80	0.0310	60.21
Jordan	4009	93	0.0780	75	0.0310	60.25
Death penalty	4298	84	0.0940	74	0.0469	50.11
Geometry	4326	89	0.0790	77	0.0470	40.52
Globalization	4334	91	0.0930	53	0.0309	66.77
Shakespeare	4383	92	0.0939	85	0.0470	49.95
Alcohol	4594	93	0.1089	71	0.0309	71.63
National parks	4757	91	0.0940	69	0.0469	50.11
Recipes	5243	94	0.1089	70	0.0469	56.93
Genetic	5298	89	0.1090	78	0.0469	56.97
Blues	5354	92	0.1250	80	0.0629	49.68
Basket ball	6049	78	0.1090	57	0.0469	56.97
Architecture	7399	94	0.1870	75	0.0940	49.73
Movies	7967	85	0.1410	61	0.0620	56.03
Weather	8011	88	0.1720	63	0.0780	54.66
Search engines	11659	85	0.7179	71	0.5159	28.14
Stanford	281903	92	16.0780	42	6.0940	62.10
Cnr-2000	325557	90	11.9690	47	4.6719	60.97
Stanford berkeley	683446	93	28.1090	40	9.2340	67.15
Eu-2005	862664	84	50.5940	32	16.1710	68.04
In-2004	1382908	90	57.5780	41	20.2970	64.75

performs more efficient, in the term of computation time, rather than the existing method, i.e. between 6.25 and 74.72%. It also shows that the number of pages is corresponding to the computation time.

Table 4 shows the resulted eigen vector from Pagerank algorithm using former Power Method and Modified Power Method. It shows that both methods produce identical eigen values. Which means that the proposed methods behave.

Interpolation: The behavior of the modified power method is also observed by conducting an interpolation over

n-size pages. The interpolation is conducted using SPSS 7 version 14. The behavior of Pagerank is observed based on the power function. Figure 2 shows the behavior of both methods.

Based on the power function, the behavior of both methods is observed and analyzed, whether the proposed method always performs better than existing method.

This is proven as follow: from the statistical analysis over the data using the tool, it can produce function of each method.

Table 4: Eigen values of both methods on stanford dataset

Vector eigen Metode pangkat	Vector eigen kuoseien rayleigh
5.3336616e-007	5.3336616e-007
1.1717819e-004	1.1717819e-004
8.2715887e-007	8.2715887e-007
5.3336616e-007	5.3336616e-007
8.5753614e-007	8.5753614e-007
3.8695920e-006	3.8695920e-006
6.7729037e-007	6.7729037e-007
1.2308450e-006	1.2308450e-006
3.5557744e-006	3.5557744e-006
1.3601884e-006	1.3601884e-006
6.0267646e-007	6.0267646e-007
8.1868437e-007	8.1868437e-007
1.8665718e-006	1.8665718e-006
3.2990891e-006	3.2990891e-006
5.5589956e-007	5.5589956e-007
5.3702230e-007	5.3702230e-007
1.8087374e-006	1.8087374e-006
1.4792320e-006	1.4792320e-006

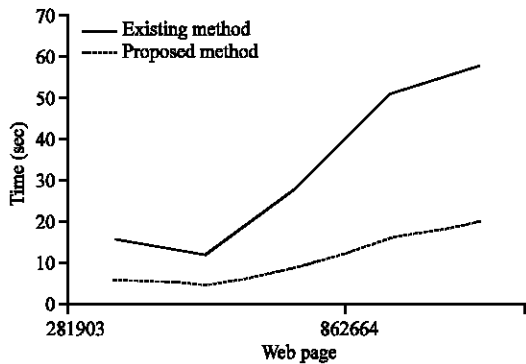


Fig. 2: The computation time of pagerank on existing and proposed methods with dataset ≥ 11.659

$$Y_{pm} = 6.841654919567834E-006 x^{1.140876913258862}$$

$$Y_{rayleigh} = 4.505598117253745E-006 x^{1.093797446996893}$$

Where:

- Y_{pm} = Y (computation time) functional equation of pagerank using existing power method
- $Y_{rayleigh}$ = Y (computation time) functional equation of pagerank using modified poer method
- x = Number of pages

Because the objective is to find any adjacent point, then both equations is written as follow:

$$Y_{pm} = Y_{rayleigh}$$

$$0.1814 = \log\left(\frac{x^{1.093797446996893}}{x^{1.140876913258862}}\right)$$

$$1.518448266 = x^{1.093797446996893-1.140876913258862}$$

$$x = 0.00014026 \approx 0$$

There is only one adjacent point present for the equations, i.e., $x \approx 0$ and it is shown from Fig. 2 that the function of proposed method is always above the existing method. This means that it is always true that the proposed method always performs better than existing methods for any number of pages bigger than 0. Given the number of existing pages in Google is 4.2 billion, it will take the existing method around 7.54 days to find the eigen values, while the proposed method will need only 1.75 day.

CONCLUSION

Ranking web page is the heart of search engine which used for determining significant searching results. One of the best ranking methods currently is PageRank, which is used by Google. From the preliminary test on sample dataset, it is shown that the advanced PageRank, which makes use of Quotient Rayleigh in the computation of power method performs significantly faster that is 64.4%, than the existing method. The interpolation also shows that given the amount of existing collection of pages that Google has the proposed method will reduce 25% of the existing computation time. This is due to the computation characteristic of Quotient Rayleigh, which allows the convergence of eigen value to happen far earlier. There is still a further research on observing the behavior of the new method on a system that similar to the existing search engines' architecture, such as Google.

ACKNOWLEDGEMENTS

We would like to thank Massimo Santini and Sebastian Vigno for allowing us to use University Milano dataset for testing our method.

REFERENCES

Austin, D., 2006. How Google Finds Your Needle in the Web's Haystack, Monthly Essays on Mathematical Topics. AMS Feature Column. <http://www.ams.org/featurecolumn/archive/pagerank.html>.

Brin, S. and L. Page, 1998. The anatomy of large-scale hypertextual web search engine. Computer Networks and ISDN Syst., 30: 107-117.

Chowdury, I. and S.P. Dasgupta, 2003. Computatios of Rayleigh Damping Coefficients for Large Systems, Indian Institutes of Technology. The Electronic J. Geotech. Eng., Vol. 8.

Garfield, E., 1979. Citation Indexing. Its Theory and Application in Science, John Wiley and Sons Inc. ISBN: 0471025593. <http://garfield.library.upenn.edu/ci/contents.pdf>.

- Gumerov, N., 2003. Rayleigh's Quotient. http://www.umiacs.umd.edu/~shaohua/enee739q_cmsc858c/RayleighsQuotient.pdf.
- Langville, M., 2005. A survey of eigenvector methods for web information retrieval. *SIAM Rev.*, 47 (1): 135-161.
- Vise, D. and M. Malseed, 2005. *The Google Story*, Bantam Dell, New York, pp: 32-44. ISBN: 0-553-80457-X.
- Wilkinson, J.H., 1965. *Algebraic Eigenvalue Problem*, Clarendon Press, Oxford University, Great Britain, pp: 172-178. ISBN-0: 19-853418-3.