IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 14, Issue 3 (Sep. - Oct. 2013), PP 42-49
www.iosrjournals.org

Analysis of Rayleigh Quotient in Extrapolation Method to Accelerate the Computation Speed of PageRank

Ali Mohammed Abulgasim Abusbaiha[1], Agus Naba[2],
M. Aziz Muslim[3]

[1] *Student of Master Program, Department of Electrical Engineering, Faculty of Engineering, Brawijaya University, Malang, Indonesia*
[2] *Department of Physics, Faculty of Sciences, Brawijaya University, Malang, Indonesia*
[3] *Department of Electrical Engineering, Faculty of Engineering, Brawijaya University*

Abstract: *The development of techniques for computing PageRank efficiently for Web-scale graphs is very important since computing a PageRank vector of Web graphs containing a billion nodes can take several days. Previous method of computing a PageRank was by using extrapolation method, which got the value based on the convergence of eigen value. We propose to intersperse the algorithm with Rayleigh quotient in the hope to accelerate the calculation. The objective of this research are (1) to analyze dataset taken from the previous research which will be adapted to this research, (2) IIto develop an algorithm to compute PageRank and speed up computation using combination of quotient Rayleigh with extrapolation method, and (3)to analyze the performance of the algorithm. This research was conducted on 36 datasets taken from Stanford and Toronto University then computing by Matlab. The result is that, by the calculation, using Rayleigh quotient inside extrapolation method can speed up the computation speed of PageRank.*

Keywords: *PageRank, extrapolation method, Rayleigh quotient, power method*

I. Introduction

Search engines have played a vital role in the World Wide Web to enable user to discover certain information in the giant space database guaranteeing that they produce relevant and important query results, which are achieved through a democratic ranking system which value is obtained by ranking (Langville, 2005). This is done using the appealing of PageRank method which was first developed by Larry Page at Stanford University (Vise, 2005). PageRank is a method used by Google to calculate the closeness of a web page to the query term given by user. It is based on citation analysis founded by Garfield (1979). When all searching result factors, such as title, tag and keywords have been used, then Google uses PageRank to show the order of importance of pages that will be displayed in the main search results. PageRank results from a mathematical algorithm based on the web graph, created by all World Wide Web pages as nodes and hyperlinks as edges, taking into consideration authority hubs such as cnn.com or usa.gov. The PageRank of a page is defined recursively and depends on the number and PageRank metric of all pages (incoming links) that link to it. A page that is linked to many pages with high PageRank receives a high rank itself. If there is no link to a web page, then there is no support for that page.

In mathematics, the power iteration is an eigenvalue algorithm: given a matrix A where the algorithm will produce a number λ (the eigenvalue) and a nonzero vector v (the eigenvector), such that $Av = \lambda v$. The algorithm is also known as the Von Mises iteration (Eiermann, 2009). The method converges slowly if there is an eigenvalue close in magnitude to the dominant eigenvalue. For a given complex Hermitian matrix M and nonzero vector x, the Rayleigh quotient $R(M, x)$, is defined as (Eiermann, 2009):

$$R(M,x) = \frac{x^*Mx}{x^*x} \quad \ldots\ldots\ldots\ldots (1)$$

For real matrices and vectors, the condition of being Hermitian reduces to that of being symmetric, and the conjugate transpose x^* to the usual transpose x'. Note that *R(M, cx) = R(M, x)* for any real scalar $c \neq 0$. Recall that a Hermitian (or real symmetric) matrix has real eigenvalues. It can be shown that, for a given matrix, the Rayleigh quotient reaches its minimum value λ_{min} (the smallest eigenvalue of M) when x is v_{min} (the corresponding eigenvector). Similarly, $R(M,x)\lambda_{max}$ and $R(M, v_{max}) \leq \lambda_{max}$.

The Rayleigh quotient is used in min-max theorem to get exact values of all eigenvalues. It is also used in eigenvalue algorithms to obtain an eigenvalue approximation from an eigenvector approximation. Specifically, this is the basis for Rayleigh quotient iteration.The range of the Rayleigh quotient is called a numerical range (Eiermann, 2009).

In addition, extrapolation means creating a tangent line at the end of the known data and extending it beyond that limit. Linear extrapolation will only provide good results when used to extend the graph of an approximately linear function or not too far beyond the known data. If the two data points nearest the pointx_* to be extrapolated are (x_{k-1}, y_{k-1}) and(x_k, y_k) , a linear extrapolation gives the function:

$$y(x_*) = y_{k-1} + \frac{x_* - x_{k-1}}{x_k - x_{k-1}}(y_k - y_{k-1}) \text{ .... (2)}$$

(which is identical to linear interpolation if $x_{k-1} < x_* < x_k$). It is possible to include more than two points, averaging the slope of the linear interpolant, by regression-like techniques, on the data points chosen to be included. This is similar to a linear prediction.

In the PageRank algorithm, eigenvalue determines the limit of the value iteration besides epsilon / error value. Eigenvalues represent the largest divisor of eigen vector (which is a ranking of the web page). It can be concluded that if the iteration stops, it will get the eigenvalue corresponding to the dominant eigen vector. Ranking in Google algorithm is currently using the power method to get the value based on the convergence of eigenvalues. According to research conducted by (Oranova, 2009), the Google algorithm that uses the power method interspersed with Rayleigh quotient generates a significant acceleration. In addition, based on the relevant research (Kamvar, 2003) in which there was a modification in pagerank by adding extrapolation method, the research assumes that there is any other way to speed up computation. In this paper the researcher proposes to use the Rayleigh quotient (Oranova, 2009) in the extrapolation method to speed up ranking computation.

As the PageRank algorithm for determining the "importance" of Web pages has become a central technique in Web search, where the core of the PageRank algorithm involves computing eigenvector of the Markov matrix representing the hyperlink structure of the Web (Kamvar, 2003), computing PageRank quickly is important to reduce the lag time. Due to the fact, this paper has been based on three research problems including how to analyze the datasets taken from the previous research which would be adapted to this research, how to combine Rayleigh quotient in extrapolation method to increase speed of computation, and how to analyze the ranking based on this research.

II. Research Method

Prior to this research, there was a dataset collection obtained from previous research then carried out the data processing in order to obtain valid data. After that, the data was going to be used into the algorithm. The data in this research were taken from previous research using datasets from several universities. The most complete dataset was a dataset taken from the Web Graph whose copyright is owned by the Laboratory for Web Algorithmic by the number of pages up to 118, 142,155 pages.

The data includes: 1) Stanford Web Matrix datasets which contain 281,903 pages (nodes) stored in a file called stanford-web.dat. with the size of 62,728 KB which was downloaded from Stanford website research for computational algorithm and 2) Crawler Dataset Matrix containing 34 of datasets with the sample "abortion" dataset which consists of 3,340 pages (nodes) which was downloaded from University of Toronto.

The data were processed using Matlab. Once the dataset had been adapted to the needs of the format of this study, they were analyzed in two different methods, using the extrapolation method and combined with the Rayleigh quotient. The results of the two methods produce eigenvectors. The eigenvectors generated from the same dataset result in the number of rows / elements of the same eigenvectors but with different values. The study was done by some experiments in Brawijaya University in 2013. This study objective was to prove the hypothesis that using Rayleigh quotient in extrapolation method can speed up computation of PageRank. The experiments were performed step by step in several stages including: 1) collecting datasets from a variety of sources, 2) validating datasets that meet the rules of Markov matrix using Matlab software, 3) applying the datasets in the PageRank algorithm in extrapolation method and recording all the results, 4) doing free memory in Matlab to obtain free memory, and 5) using validation dataset to calculate PageRank using Rayleigh quotient with extrapolation method and recording the experiment results.

III. Implementation And Testing

The experiments were performed on a Pentium Dual Core CPU 2.00 GHz, with a Notebook memory 4Giga, with Windows Vista, Ultimate Edition platform. 36 datasets were analyzed using Matlab software version R2011b with Matlab source code from some previous researchers with some modifications.

There were two scenarios in the experiments. The first scenario processes consisted of experiments using 36 datasets from the previous tests. The experiments were performed one after another with the data to refresh the memory. The method used was the method of extrapolation taken from Sepandar D. Kamvar, Google Laboratory. Checking the extrapolation converges was done by looking for residual between previous eigenvectors with eigenvectors as iteration. The second scenario performed experiments using the same 36 datasets from previous tests and tried them one after the other. However, the method used was taken from Google Labs with adding Rayleigh quotient to the source codes. Furthermore, there was a residual comparison done between the eigenvalues of the previous dominant and those of the current one. The algorithm has been downloaded and already written with Matlab code. Figure 1 shows the code of the method of extrapolation.

```
while (residual >= epsilon)
 prevpi=pi;
 k=k+1;
 pi=alpha*pi*H + (alpha*(pi*a)+1-alpha)*v;
 residual=norm(pi-prevpi,1);
 if (mod(k,l))==0
  % 'quadratic extrapolation'
   nextpi=alpha*pi*H + (alpha*(pi*a)+1-alpha)*v;
   nextnextpi=alpha*nextpi*H + (alpha*(nextpi*a)+1-alpha)*v;
   y=pi-prevpi;   nexty=nextpi-prevpi;   nextnexty=nextnextpi-prevpi;
   Y=[y' nexty'];
   gamma3=1;
   % do modified gram-schmidt QR instead of matlab's [Q,R]=qr(Y);
   [m, n] = size(Y);
   Q = zeros(m,n);
   R = zeros(n);
   for j=1:n
     R(j,j) = norm(Y(:,j));
     Q(:,j) = Y(:,j)/R(j,j);
     R(j,j+1:n) = Q(:,j)'*Y(:,j+1:n);
     Y(:,j+1:n) = Y(:,j+1:n) - Q(:,j)*R(j,j+1:n);
   end
   Qnextnexty=Q'*nextnexty';
   gamma2=-Qnextnexty(2)/R(2,2);
   gamma1=(-Qnextnexty(1)-gamma2*R(1,2))/R(1,1);
   gamma0=-(gamma1+gamma2+gamma3);
   beta0=gamma1+gamma2+gamma3;
   beta1=gamma2+gamma3;
   beta2=gamma3;
   nextnextpi=beta0*pi+beta1*nextpi+beta2*nextnextpi;
   nextnextpi=nextnextpi/sum(nextnextpi);
   pi=nextnextpi;
   %'end quadratic extrapolation'
 end
 pi=pi/sum(pi);
end
numiter=k;
time=toc;
```

Figure 1 Matlab Code for Extrapolation Method

Based on figure 1 something that needs attention is the iteration will stop when the condition of residual >= epsilon are not met by the algorithm that we have a hypothesis that the change of residual value will affect the speed of convergence by not changing the number of iterations intake. This is because the resulting eigen vector has the same value.

Modification of Extrapolation Algorithm with Rayleigh Quotients

Matlab source code of the extrapolation method was modified using Rayleigh quotients based on the calculation of residual values to achieve much faster convergence.

```
while (residual >= epsilon)
 prevpi=pi;
 k=k+1;
 pi=alpha*pi*H + (alpha*(pi*a)+1-alpha)*v; %pi=xnew
 residual=norm(pi-prevpi,1);
 if (mod(k,l))==0
  % 'quadratic extrapolation'
   nextpi=alpha*pi*H + (alpha*(pi*a)+1-alpha)*v;
   nextnextpi=alpha*nextpi*H + (alpha*(nextpi*a)+1-alpha)*v;
   y=pi-prevpi;   nexty=nextpi-prevpi;   nextnexty=nextnextpi-prevpi;
   Y=[y' nexty'];
   gamma3=1;
   % do modified gram-schmidt QR instead of matlab's [Q,R]=qr(Y);
   [m, n] = size(Y);
   Q = zeros(m,n);
   R = zeros(n);
   for j=1:n
     R(j,j) = norm(Y(:,j));
     Q(:,j) = Y(:,j)/R(j,j);
     R(j,j+1:n) = Q(:,j)'*Y(:,j+1:n);
     Y(:,j+1:n) = Y(:,j+1:n) - Q(:,j)*R(j,j+1:n);
   end
               Qnextnexty=Q'*nextnexty';
   gamma2=-Qnextnexty(2)/R(2,2);
   gamma1=(-Qnextnexty(1)-gamma2*R(1,2))/R(1,1);
   gamma0=-(gamma1+gamma2+gamma3);
   beta0=gamma1+gamma2+gamma3;
   beta1=gamma2+gamma3;
   beta2=gamma3;
   nextnextpi=beta0*pi+beta1*nextpi+beta2*nextnextpi;
   nextnextpi=nextnextpi/sum(nextnextpi);
   pi=nextnextpi;
                %rayleigh quotient
               lambda = (pi'*nextpi)/(pi'*pi);
                residual=sum((lambda-pi)/n);
%'end quadratic extrapolation'
 end
 pi=pi/sum(pi);
end
numiter=k;
time=toc;
```

Figure 2 Modification of Extrapolation method by using Rayleigh Quotient

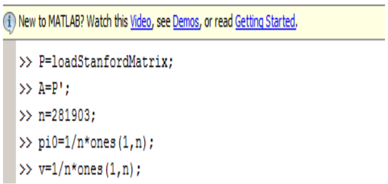
Matlab code figure 2 shows that the addition of two lines of program code. The first additional line `lambda = (pi'*nextpi)/(pi'*pi);`would calculate the corresponding dominant eigenvalues which used as a comparison with the value of epsilon. While code on `residual=sum((lambda-pi)/n);` is the command to find the difference in eigenvalues with the value of *pi.*

Experiment on Stanford Dataset

Stanford dataset consist of two dataset, i.e. Stanford Web Matrix which was including 281,903 pages/nodes and Stanford Berkeley Web Matrix with 683,446 pages/nodes.

a. Processing dataset and getting *A* matrix

This step would process the dataset, getting *A* markov matrix, and setting for the parameter of algorithm. (See Figure 3)



New to MATLAB? Watch this Video, see Demos, or read Getting Started.

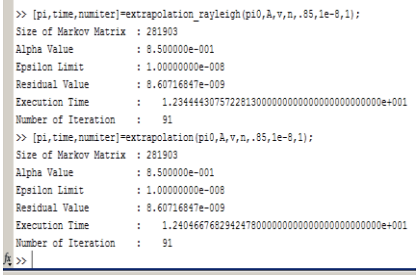
```
>> P=loadStanfordMatrix;
>> A=P';
>> n=281903;
>> pi0=1/n*ones(1,n);
>> v=1/n*ones(1,n);
```

Figure 3 Parameter configuration and getting *A* matrix

After the Stanford Matrix was executed, then the value of *P* would be filled by Stanford matrix which contained 281,903 pages / nodes. In the existing manual on owner's web dataset, we mentioned that the matrix had to be transposed first, so we got the *A* as Markov matrix that would be used in this experiment. After giving an initial value to pi0 (eigen vector) where this value was the starting vector at iteration 0

b. Executing the code

After all the parameters and the requirements to run the application had been determined, the next step was to run the Matlab code.



```
>> [pi,time,numiter]=extrapolation_rayleigh(pi0,A,v,n,.85,1e-8,1);
Size of Markov Matrix  : 281903
Alpha Value            : 8.500000e-001
Epsilon Limit          : 1.00000000e-008
Residual Value         : 8.60716847e-009
Execution Time         :    1.23444430757228130000000000000000000000e+001
Number of Iteration    :    91
>> [pi,time,numiter]=extrapolation(pi0,A,v,n,.85,1e-8,1);
Size of Markov Matrix  : 281903
Alpha Value            : 8.500000e-001
Epsilon Limit          : 1.00000000e-008
Residual Value         : 8.60716847e-009
Execution Time         :    1.24046676829424780000000000000000000000e+001
Number of Iteration    :    91
>>
```

Figure 4 Execution of Extrapolation and Extrapolation with Rayleigh

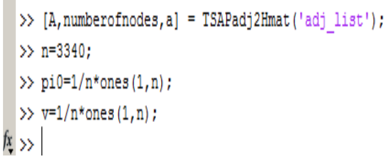
Based on Figure 4, important information would be showed in the execution result of the program code, i.e. the size of matrix, alpha value, epsilon limit, residual value, execution time. The number of iteration could be used as an analysis in this experiment.

Experiment on Crawler (Toronto University) Dataset

Experiment in this dataset (which consists of 34 datasets) was similar to that of the Stanford Matrix. In this explanation a dataset with the topic "abortion" in which consists of 3,340 pages / nodes would be selected.

a. Processing dataset and getting *A* matrix

This step will process the dataset, get *A* markov matrix, and set for the parameter of algorithm. (See Figure 5)



```
>> [A,numberofnodes,a] = TSAPadj2Hmat('adj_list');
>> n=3340;
>> pi0=1/n*ones(1,n);
>> v=1/n*ones(1,n);
>>
```

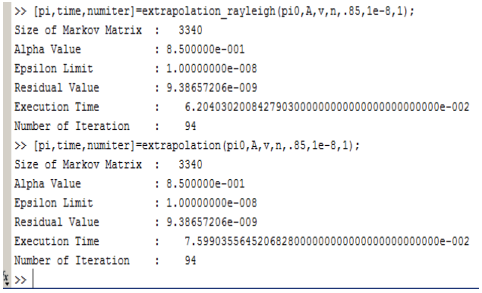
Figure 5 Parameter configuration and getting *A* matrix

Once the command was executed by [*A*, number of nodes, a] = TSAPadj2Hmat ('adj_list'); then the value of A would be filled by abortion matrix containing 3,340 pages / nodes. In the existing manual on owner's web dataset, we mentioned that the obtained matrix was the Markov matrixA that would be used in this experiment.

After giving an initial value to pi0 as initial eigenvector, then the final configuration was to provide value as teleportation of vector v (1-by-n row vector).

b. Executing the code

After all the parameters and the requirements to run the application has been determined, the next step is to run the Matlab code.



```
>> [pi,time,numiter]=extrapolation_rayleigh(pi0,A,v,n,.85,1e-8,1);
Size of Markov Matrix  :   3340
Alpha Value            : 8.500000e-001
Epsilon Limit          : 1.00000000e-008
Residual Value         : 9.38657206e-009
Execution Time         :    6.20403020084279030000000000000000000000e-002
Number of Iteration    :    94
>> [pi,time,numiter]=extrapolation(pi0,A,v,n,.85,1e-8,1);
Size of Markov Matrix  :   3340
Alpha Value            : 8.500000e-001
Epsilon Limit          : 1.00000000e-008
Residual Value         : 9.38657206e-009
Execution Time         :    7.59903556452068280000000000000000000000e-002
Number of Iteration    :    94
>> |
```

Figure 6 Execution of Extrapolation and Extrapolation with Rayleigh

Figure 6 shows important information as result visible after the execution of the program code, i.e. the size of matrix, alpha value, epsilon limit, residual value, execution time, and the number of iteration in which the results could be used as an analysis in this experiment.

IV. Experiment Results

Table 1 shows the results of calculations performed the dataset measuring 742 until 683.446 nodes. It was found that using the Rayleigh quotients could speed up time calculation compared to the calculation using the extrapolation without Rayleigh algorithm. This was because the Rayleigh quotient error detection was performed on the dominant eigenvalue calculation. It was more efficient than checking by the extrapolation by comparing the error vector eigenvalues.

Table 1 Iteration Result Calculation of Pagerank.

Number	Name of dataset	∑ Nodes/URL	Extrapolation		Extrapolation with Rayleigh		% (Rayleigh/ Extrapolation)
			∑ Iteration	Time (s)	∑ Iteration	Time (s)	
1	Abortion	3,340	94	0.075	94	0.062	82,7%
2	affirmative action	2,523	97	0.053	97	0.047	88,7%
3	Alcohol	4,594	92	0.082	92	0.077	93,9%
4	amusement parks	3,410	89	0.058	89	0.050	86,2%
5	Architecture	7,399	93	0.132	93	0.104	78,8%
6	Armstrong	3,225	88	0.051	88	0.050	98,0%
7	automobile industries	1,196	92	0.033	92	0.025	75,8%
8	basket ball	6,049	77	0.073	77	0.070	95,9%
9	Blues	5,354	92	0.096	92	0.095	99,0%
10	Cheese	3,266	90	0.059	90	0.051	86,4%
11	classical guitar	3,150	86	0.062	86	0.053	85,5%
12	complexity	3,564	82	0.066	82	0.050	75,8%
13	computational complexity	1,075	88	0.036	88	0.029	80,6%
14	computational geometry	2,295	32	0.025	32	0.021	84,0%
15	death penalty	4,298	83	0.081	83	0.062	76,5%
16	Genetic	5,298	88	0.090	88	0.072	80,0%
17	Geometry	4,326	88	0.082	88	0.076	92,7%
18	globalization	4,334	90	0.072	90	0.066	91,7%
19	gun control	2,955	95	0.061	95	0.053	86,9%
20	iraq war	3,782	87	0.058	87	0.052	89,7%
21	Jaguar	2,820	95	0.061	95	0.058	95,1%
22	Jordan	4,009	92	0.073	92	0.072	98,6%
23	moon landing	2,188	92	0.053	92	0.049	92,5%
24	movies	7,967	84	0.123	84	0.102	82,9%
25	national parks	4,757	90	0.085	90	0.083	97,6%
26	net cencorship	2,598	92	0.063	92	0.062	98,4%
27	randomized algorithm	742	81	0.029	81	0.018	62,1%
28	Recipes	5,243	93	0.092	93	0.091	98,9%
29	Roswell	2,790	95	0.064	95	0.058	90,6%
30	search engines	11,659	84	0.317	84	0.291	91,8%
31	shakespeare	4,383	91	0.085	91	0.070	82,4%
32	table tennis	1,948	77	0.044	77	0.035	79,5%
33	vintages car	3,460	87	0.071	87	0.067	94,4%
34	Weather	8,011	87	0.069	87	0.064	92,8%
35	stanford web	281,903	91	12.404	91	12.344	99,5%
36	stanfordberkeley	683,446	92	13.787	92	13.157	95,4%

Table 1 explains time of calculation for each dataset. The percentage in the last column is calculated by dividing time execution using Rayleigh on extrapolation method with time execution using extrapolation method. Detail of the result of pagerank is described as an example in table 2.

From the calculation of multiple datasets mentioned above, it appears that the Rayleigh quotient used in extrapolation algorithm can speed up calculation times. With the experimental dataset between 724 up to 683.446 pages, we can see that the more pages, the longer time calculations. Table 2 shows that the eigenvector calculation result in the same eigenvalues, either using PageRank algorithm or Rayleigh quotients in pagerank algorithm.

Table 2 Example of Comparison Eigenvectors Extrapolation and Extrapolation with Rayleigh

Number of Nodes/Pages	Eigenvector with extrapolation	Eigenvector with extrapolation and rayleigh
1	5.1797094e-007	5.1797094e-007
2	5.8837501e-006	5.8837501e-006
3	2.6524302e-006	2.6524302e-006
4	4.4873122e-006	4.4873122e-006
5	5.1797094e-007	5.1797094e-007
6	4.4553539e-006	4.4553539e-006
7	5.1797094e-007	5.1797094e-007
8	4.9652251e-007	4.9652251e-007
9	1.9021194e-006	1.9021194e-006
10	1.9233442e-006	1.9233442e-006
11	6.4599975e-007	6.4599975e-007
12	4.2404149e-006	4.2404149e-006
13	2.5895378e-007	2.5895378e-007
14	5.1797094e-007	5.1797094e-007
15	6.0248852e-007	6.0248852e-007

Both method produce the same eigen vector even the computation time is varying between each dataset.

Analysis Method of Calculation Time Rank with Rayleigh Quotients

Based on the results of the time experiment, the good data for experimental results by extrapolation method and the Rayleigh quotient would be processed and determined in graphs. The graphs were useful to determine the pattern of behavior of the system, such as the point of intersection of the lines, shift or about the curvature of the line.

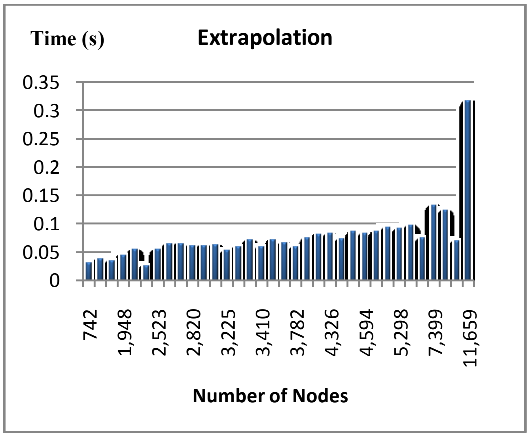


Figure 7 Extrapolation graph.

Figure 7 appears that the larger the size of the pages / nodes in a matrix may result in continuously rising curve. The increase of this curve is proportional to the number of pages, in addition to it, curves illustrate that the longer of the PageRank calculation would require a lot of time with a linear increasing.

The curves in Figure 8 show the time execution after Rayleigh quotient inserted in the extrapolation method.

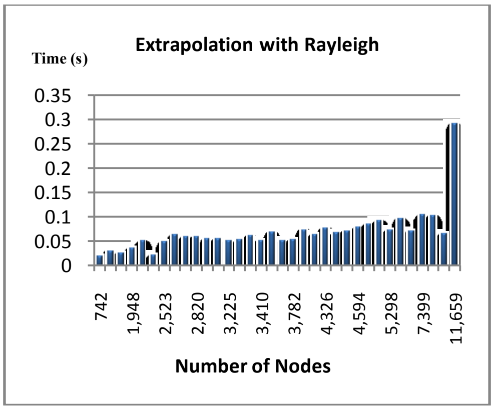


Figure 8 Extrapolation with rayleigh graph of time execution and number of nodes.

The curve in Figure 8 shows that if the number of pages increases linearly over time then it will also affect the calculation time. In the Figures 7 and 8, there is no difference in the behavior of the curve.
There are few datasets with various sizes from a very large to some small-sized bit. It will be shown that the size of the dataset is uniform in the number. More clearly, it can be seen in the Figure 9.

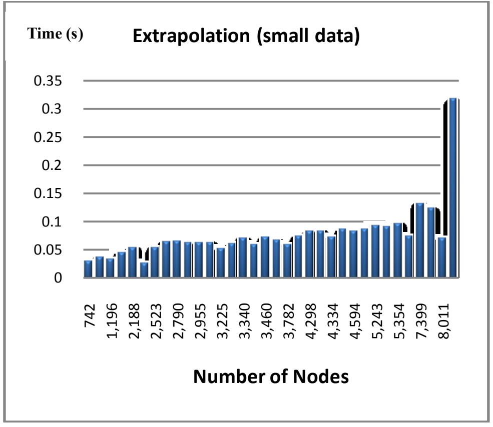


Figure 9 Extrapolation line graph for small data with x as the number of pages and y as time execution.

Figure 9 explains about the curve between the number of pages and execution times. It shows that increasing the number of pages will affect the value of y, in this case the execution time will grow up.
The curve in Figure 10 provides an overview of the relationship between the number of pages in the algorithm and the execution time in extrapolation method that was inserted with the Rayleigh quotient.

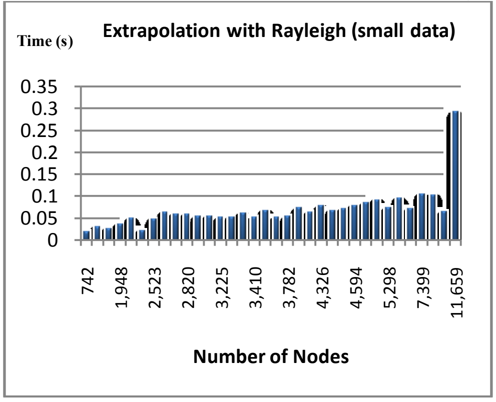


Figure 10 Extrapolation with rayleigh line graph for small data with x is the number of pages and y is time execution.

Figure 10 shows that after the Rayleigh quotient is inserted in extrapolation method, the bigger the number of pages on this calculation the longer the execution time.

As a conclusion of the analysis, the curves between the number of pages or size of the matrix with the length of the execution time can be seen in Figure 11.

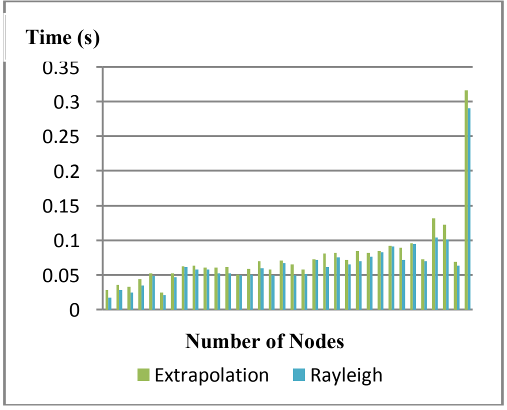


Figure 11 Comparison of time execution between Extrapolation and Extrapolation with Rayleigh for small data with x as the number of pages and y as time execution.

Figure 11 describes the position of the extrapolation algorithm execution time which is implied by the Rayleigh quotient and the curve is always better (lower execution time) compared to the extrapolation method.

V. Conclusion

After the results had been analyzed using the curve to determine the behavior of the algorithm, the conclusions of this study are drawn as follows. The PageRank algorithm analyses using the power method and modified by the extrapolation method had been done and the result showed that the more the number of pages, the more the time to calculate. PageRank algorithm which was already modified by extrapolation method still could be improved by adding the Rayleigh quotient inside the extrapolation method. The result was shown in the graph as linier curve approximation. The computation speed by using the Rayleigh quotient inside the extrapolation method was faster than that without using the Rayleigh quotient.

Recommendation

Since it was an individual research, it is suggested that the next research can be done in a group to develop better algorithms. Moreover, further research can use some methods that have not been used to accelerate the calculation of PageRank. Some method that already using by yahoo or altavista can be integrated in this algorithm to make a good improvement on computation speed. It is also recommended that this research should have used the most recent dataset to easily know the behaviour of the system and the Markov matrix and supported from search engine owner like Google.

References

[1]. Bala. 2011. An Overview of Efficient Computation of PageRank.
[2]. Eiermann. 2009. The Numerical Solution of Eigen Problems. TU Bergadademie Freiberg.
[3]. Garfield. 1979. Citation Indexing: Its Theory and Application in Science, Technology,and Humanities. New York: John Wiley.
[4]. Kamvar. 2003. Extrapolation method for Accelerating PageRank Computations. Stanford University
[5]. Langvilleand, Meyer. 2005. Deeper Inside PageRank. Internet Mathematics, Vol. 1(3):335-380.
[6]. Mitra. 2012. CA Based Moore Filter in SEO. IJARCSSE
[7]. Oranova, Arifin. 2009. Implementing of Quotien Rayleigh in Power Method to Accelerate PageRank Computation. Medwell Journal Vise, David and Malseed , Mark. 2005. The Google Story. p. 37. ISBN ISBN 0-553-80457-X.