

A Survey and Comparative Study of Different PageRank Algorithms

Tahseen A. Jilani
University of Karachi
Karachi, Pakistan

Ubaida Fatima
NED University
Karachi, Pakistan

Mirza Mahmood
Baig
NED University
Karachi, Pakistan

Saba Mahmood
IoBM
Karachi, Pakistan

ABSTRACT

Searching the World Wide Web is an NP complete problem with sparse hyperlink matrices. Thus searching the significant search results is a challenge. Google's PageRank attempted to solve this problem using computing of principle Eigenvalues termed as PageRank vector. After this, a number of techniques were developed to speed up the convergence patterns of pages in the PageRank algorithm. This is paper, we have reviewed a number of of PageRank computation techniques. The main objective of all these techniques is the convergence rate along with space and time complexities. In this paper, a comparative study is presented among Standard Power method, Adaptive Power Method and Aitken's method using SNAP Google web pages dataset.

Keywords

Google PageRank, Aitken's PageRank, Power method, Adaptive PageRank

1. INTRODUCTION

In contemporary computer generation, internet has become indispensable in our lives and cognizance is only a click away. We just open our desired search engines, like Google, Yahoo, Bing, and the search engine will show the webpages appropriate for our search.

Google's humongous triumph as a search engine can be ascribing to numerous elements, including its naiveness, acceleration and ease of employ. Conversely, the most eminent grounds for their triumph is due to their search algorithm; contrast to other search engines; Google provides the most significant ensue first.

Google as a search engine inevitably to be able to execute two errands. First it requires to acquire and retain all of the webpages it is able to; this attain by crawling the web ad indexing the data that it encounters. Second, it requires being able to figure out the order of pages resumed by any search survey. This is accomplished through Google's PageRank algorithm as proposed by Wills[1], which assesses each webpage and its status relative to other webpages. "Significance", as defined by PageRank algorithm, is subject on the number of other pages, linking to a webpage. This is a prejudiced factor which includes the importance of other pages, meaning that a high importance webpage will impart more importance to a linked page than an irrelevant page.

Lin et.al [2] presented the idea of lumping to reduce the large Google matrix into small block matrices by knitting dangling nodes into a one node. It was observe that reduced Google matrix had same non-zero Eigenvalues as the full Google matrix had by the application of Power method. The reduced matrix was further lumped into weakly non-dangling nodes to another single node with the same non-zero Eigenvalues as the Google

matrix. Kamvar, Haveliwala and Golub [3] detected that the web pages that converge more rapidly cannot be work out again and again. Considering this they reckon the simple algorithm named as Adaptive Algorithm to speeds up the estimation of PageRank up to 30%.Wills [1] studied the mystery behind the Google search engine. The PageRank score of a webpage symbolizes the probability that the random Web surfer select to view the web page. In 1998, Brin and Page originated the concept of personalized PageRank. Fu, Lin, Tsai [4] discussed that how Google PageRank have a great influence by damping factor. To replace the damping factor they proposed the design of modified algorithm ground on input-output ratio. By using this modified algorithm instead of selecting the most desirable damping factor they observe the same effect on computation as the previous PageRank algorithms have Ispen and Selee [5] introduced a fundamental algorithm of Google matrix for the computation of PageRank vector. All dangling nodes were lump into a one node by this novel algorithm. The algorithm proceeds by reckoning PageRank of Non-dangling nodes individually than reckoning PageRank for Dangling nodes. They also discuss the case when all Web nodes are dangling nodes by presenting Jordan decomposition. Kamvar et. al [6] posed a modified algorithm to speed up the computation of PageRank, to estimate the significance of web pages. Power Method is the primal method of computing PageRank of Google stochastic matrix. The algorithm proposed here is Quadratic Extrapolation to quicken the rate of convergence of PageRank vector. 25-300% were fastens the computation of PageRank by Quadratic Extrapolation method.

Wu and Wei [7]were originated the extrapolation method for the calculation of PageRank vector based on Ritz values to merge it with Arnoldi-type method. As Arnoldi-type algorithm delivered by Golub and Grief was not efficient for larger values of damping factor. They examine how to periodically entwine extrapolation and Arnoldi-type method. The rate of convergence of this new algorithm was inferred. Numerical behaviors of this novel algorithm were recognized by Numerical experiments. Yu et.al. [8] improved lumping the web matrix for five type pages and present a integrated form of widely appreciated lumping methods for PageRank. They prove that their novel policy for lumping demotes the PageRank problem to a littler form, and the full PageRank vector can be easily derived by some recursion designs. First order derivative is computed in lumping algorithm for PageRank. They observe in Numerical tests that their new algorithm support large matrices data and support when they are using large values of damping factor

Kamvar et. al[3] posed a modified algorithm to speed up the computation of PageRank, to estimate the significance of web pages. The algorithm proposed by them is Quadratic Extrapolation to quicken the rate of convergence of PageRank vector. With their research the convergence rate decreased from 25% to 300%.

The paper is organized as follows. In section 2, a brief review of Directed acyclic graphs (DAG) approaches of the World Wide Web, concepts of dangling nodes and development of Google matrix is presented. In section 3, a review of different numerical methods for PageRank vector is presented. In Section 4, simulation applications are presented using the different PageRank techniques. Finally a comparative study is presented with conclusion and future work.

2. THE HYPERLINK ANATOMY OF THE WEB

The number of pages in the web may be demonstrated as nodes in a directed graph. The edges between nodes represent links between graphs. A graph of a simple 5-node web is demonstrated in Figure 1 below. The directed edge from node a to node b signifies that page a links to page b. However, page b does not link to page a, so there is no edge from node b to node a.

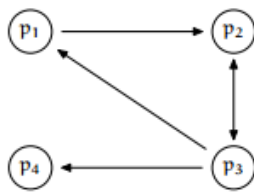


Fig 1: DAG of webpages

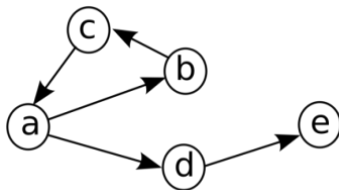


Fig 2: Graphical Demonstration of five-page Web

Usually, most important pages will have more inlinks. Inlinks from important pages will also have a more effect on PageRank for a particular page than inlinks from marginal pages. The calculation of PageRank is recursive.

The method for evaluating PageRank starts by demonstrating the directed Web graph as the square matrix of dimension $n \times n$ named "Adjacency Matrix", A , where n is the number of webpages. If webpage i has $l_i \geq 1$ links to other webpages and webpage i links to the webpage j of $A_{ij} = \frac{1}{l_i}$. Else, $A_{ij} = 0$. For the above directed graph Adjacency matrix is:

$$A = \begin{bmatrix} 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Where page e is a dangling node that is row 5 of matrix A .

Links are number of non-zero elements in Adjacency matrix or Hyperlink matrix.

2.1 Dangling Nodes

The Web pages with no outlinks are called dangling nodes. Further, the remaining Web pages, having at least one outlink

are identified as Non-Dangling nodes. For the execution of PageRank we must have to resolve how to deal with dangling nodes and this decision has influence on PageRank that we will compute.

For above graph dangling node can be handled by replacing dangling node row of Hyperlink matrix A by the probability distribution vector, w , the stochastic n -dimensional row vector that sum equals to 1, n is number of Web pages or nodes. The resultant matrix can be of the form, $P = A + dw$, where d is a dangling node vector, n -dimensional column vector that is:

$$d_i = \begin{cases} 1, & \text{if } l_i = 0 \\ 0, & \text{else} \end{cases}$$

$$w = [w_1 w_2 w_3 \dots \dots \dots w_n]$$

The most appropriate selection for w is the uniform row vector [3] that is represented by:

$$w = \left[\frac{1}{n} \frac{1}{n} \dots \frac{1}{n} \right]$$

Similarly for above directed graph w will be:

$$w = \left[\frac{1}{5} \frac{1}{5} \frac{1}{5} \frac{1}{5} \frac{1}{5} \right]$$

This novel matrix P should not contain any zero row means that all nodes have at least one outlink. On this ground matrix P for above 5 Web page directed graph can be written as:

$$P = A + dw = \begin{bmatrix} 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix}$$

$$P = \begin{bmatrix} 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix}$$

2.2 Google Matrix

Google figure the matrix as:

$$G = \alpha P + (1 - \alpha) \mathbf{1}v$$

Where,

α is damping factor, $0 \leq \alpha < 1$

$\mathbf{1}$ is a column array containing all elements equal to 1.

v is a personalization vector that is n - dimensional column vector.

2.3 Personalization vector

Personalization vector is added to control the resultant PageRank vector that can be biased to prefer certain kind of pages after landing to Dangling node. It represent uniform probability distribution over all nodes.[1]

By any time, with probability $(1 - \alpha)$, a surfer touring any Web page will jump to a random node. The destination of the random jump is selected according to the probability distribution displayed in \vec{v} is known *teleportation*. [3]

Google matrix G is a Markov matrix that is a row stochastic matrix. Such that $0 \leq G_{ij} \leq 1$, where G_{ij} represents each element of Google matrix and sum of each row of G is equal to 1. In numerous research articles values of α was taken between 0.85 and 0.99 for the computation of PageRank vector.[1]

3. NUMERICAL METHODS FOR COMPUTING PAGERANK VECTOR

In the following, we have reviewed a number of iterative techniques that are used for PageRank computation for numerical approximation to deal with problems in mathematical inference. These techniques include a number of sub techniques as well.

3.1 Simple Power Method

For the determination of leading eigenvalue and its corresponding eigenvector the method famous for its simplicity is power method. It can be applied to approximate the dominant eigenvalue of the Google matrix. Eigenvalues must be arranged as

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_n$$

λ_1 must be strictly greater than λ_2 that is larger than or equal to the remaining Eigenvalues. Therefore this method is known for its consistent and authentic executions.[1]

3.1.1. Aggregation or Disaggregation Iterative Approach

Aggregation or Disaggregation iterative method is used to expeditiously compute PageRank vector than the Power Method as studied by Zhu, Yu and Li [10]. Aggregation or Disaggregation iterative approach develop from the concept of Markov chains. Ipsen and Kirkland [11] proved that the iterative aggregation or disaggregation had the fastest convergence rate as compared to Power method for the computation of PageRank vector.

3.1.2. Lumping

Lin et. al. [2] presented the idea of lumping to reduce the large Google matrix into small block matrices by knitting dangling nodes into a one node. It was observe that reduced Google matrix had same non-zero Eigenvalues as the full Google matrix had by the application of Power method. Through lumping concept a numerous calculation can be avoided for computation of full PageRank vector.

3.2 Adaptive Method

Kamvar, Haveliwala and Golub [3] detect that the web pages that converge more rapidly cannot be work out again and again. Considering this they reckon the simple algorithm named as

Adaptive Algorithm to speeds up the estimation of PageRank up to 30%.

3.3 Aitken's Method

The linear convergence of the iterative methods can be improved with the help of the Aitken's method. Let $\{p_n\}$ be a sequence which converges to its limit p linearly. That is, there exists a positive number λ such that:

$$\lim_{n \rightarrow \infty} \frac{|p_{n+1} - p|}{|p_n - p|} = \lambda$$

Observe the following. For sufficiently large n,

$$\frac{|p_{n+1} - p|}{|p_n - p|} \approx \lambda$$

Algebraically these two formulas are equivalent, but numerically the first one is more stable than the second one. The sequence $\{\hat{p}_n\}$ converges to p more rapidly. This method is called Aitken's Δ^2 Method. Observe the estimate \hat{p}_n depends on estimates p_n, p_{n+1} and p_{n+2} . So \hat{p}_0 can be computed after \hat{p}_2 is computed. [6] [12]

Algorithm: Aitken's Δ^2 Method:

Let $\{p_n\}$ be generated by a method which has a linear convergence.

Having p_0, p_1 and p_2 compute

$$\hat{p}_0 = p_0 - \frac{(p_1 - p_0)^2}{p_2 - 2p_1 + p_0}, \quad \text{and for } n = 1, 2, \dots$$

Compute p_{n+2} ;

1. Compute

$$\hat{p}_n = p_n - \frac{(p_{n+1} - p_n)^2}{p_{n+2} - 2p_{n+1} + p_n}; \text{ and}$$

2. The algorithm terminates and $p \approx \hat{p}_n$ if $|\hat{p}_n - \hat{p}_{n-1}| < \epsilon$.

3.4 Quadratic Extrapolation Method

This method intensifies the convergence of power method by periodically deducting off estimates of the non-principal eigenvector, from the present iteration of Power Method. [12]

3.4.1 Definition of Dominant Eigenvalue and Dominant Eigenvector

Let $\lambda_1, \lambda_2, \dots$ and λ_n be the Eigenvalues of an n x n matrix A. λ_1 is called the dominant eigenvalue of A if

$$|\lambda_1| > |\lambda_i|, \quad i = 2, \dots, n$$

The eigenvectors corresponding to λ_1 are called dominant eigenvectors of A. [13]

3.5 Power Method

The power method for approximating Eigenvalues is an iterative. First we assume that the matrix A has a dominant eigenvalue with corresponding dominant eigenvectors. Then we choose an initial approximation x_0 of one of the dominant

eigenvectors of A. This initial approximation must be a nonzero vector in \mathbf{R}^n . Finally we form the sequence given by

$$x_k = Ax_{k-1} = A(A^{k-1}x_0) = A^k x_0$$

For greater powers of k, and by correctly scaling the vector, we can achieve a fine approximation of the dominant eigenvector of A. A dominant eigenvector of the matrix A was approximated by the power method. [13][14]

3.6 Power Method with Scaling

It is good to do “scaling” for each approximation before proceeding to the next iteration. One method to achieve this task of scaling to find the greatest component of that has the largest absolute value and then multiplies the matrix by the reciprocal of this component. The resulting matrix will have the less than or equal to 1 components. [13][14]

A link from any page u to page v can be perceived as manifest that v is “paramount” page. Extent of significance of v allocate by u is directly proportional to the essentialness of u. Accidentally surfer touring page u at time k, thereafter surfers opt a node v_i from among u’s out neighbor with probability $1/\text{deg}(u)$ at time $k+1$. P is the stochastic transition matrix from I to j page with $P_{ij}=1/\text{deg}(i)$. For P to be authentic transition matrix, P should not comprises of rows consisting of all zeros. To recover this P’ introduce where each node has at least one outlink.

Dangling node n-dimensional column vector is d’, n is number of pages, v is n-dimensional column vector representing a uniform probability distribution over all nodes and e is n-dimensional column vector where every element is 1.

The consequence of D is customizing the distribution given by v to update the transition probabilities so that a surfer visiting a dangling page randomly moves to any other page in succeeding time.

The Markov chain outlined by P’ has a unique stationary probability distribution if P is aperiodic and irreducible; the latter holds for the Markov chain persuade by the Web graph. The former holds iff G is strongly connected, which is typically not the fact for the Web graph. In circumstance of computing PageRank, the standard way of ensuring this property is to add a new set of complete transition graph. In matrix notation, we construct the irreducible Markov matrix P’’ as follows:

$$E = e \cdot v^T$$

$$P'' = cP' + (1 - c)E$$

The consequence of E at each time step with probability $(1 - c)$ is when a surfer touring any node move to a random Web page instead of following at will outlink. This is referred to as *teleportation*.

The terminus of the random move is chosen consequently to the probability distribution specified by v.

By delineating the vector v specified in above Equation to be non-uniform, so that D and E add artificial transitions with non-uniform probabilities. The outcome PageRank vector can be prejudiced to opt certain kind of pages for this reason v is known to be *personalization* vector. [3]

4. PAGERANK COMPUTATIONS USING DIFFERENT METHODS

In this section we have compared a number of different methods that are used for computing PageRank. For each method an algorithm is also presented.

4.1 PageRank Computation by Power Method:

An iterative method that is Power Method is used to estimate the stationary distribution PageRank vector $\pi^{(k+1)}$ from $\pi^{(k)}$ as $\pi^{(k+1)} = G\pi^{(k)}$ until $\pi^{(k+1)}$ converges to a preferred tolerance level. When $\pi^{(k+1)}$ converges, that vector is the eigenvector for the specified matrix and dominant eigenvalue. The PageRank algorithm is an employment of power method.

Algorithm: 1 PageRank by Power Method

```
function PageRank(G,  $\pi^0$ )
repeat
 $\pi_{k+1} \pi_{k+1} = G \pi_k \pi_k$ ;
 $r = \|\pi_{k+1} \pi_{k+1} - \pi_k \pi_k\|$ ;
until  $r < \epsilon$ 
return  $\pi_{k+1} \pi_{k+1}$ ;
```

where r is the difference between the k^{th} iteration to the $(k + 1)^{th}$ iteration and ϵ is the intended convergence level.

4.2 Adaptive Method:

Kamvar et.al. [3] studied the rate of convergence of PageRank. Various pages converge to their true PageRank rapidly. Pages which have low rate of convergence have high PageRank.

They proposed that for quick processing the pages that have fast convergence rate means pages that converges in few iterations excluded the calculations of subsequent iterations for that pages.

By Power method in algorithm: 1, we estimate $\pi^{(k+1)}$ from π^k as $\pi^{(k+1)} = G\pi^{(k)}$. Power method involves repetitious reckoning for pages that have previously converged.

Suppose C contains the converged $n - m$ pages for the iteration π^k , and suppose N contains the not converged m pages. Now current iteration may be written as:

$$\pi^k = \begin{bmatrix} \pi_N^k \\ \pi_C^k \end{bmatrix}$$

The matrix G may also rend into two sub-matrices. Let G_N be the $m \times n$ sub-matrix associated to the pages that have not converged, and G_C the $(n - m) \times n$ sub-matrix associated to those pages that have converged. The Power method would be:

$$\begin{bmatrix} \pi_N^{k+1} \\ \pi_C^{k+1} \end{bmatrix} = \begin{bmatrix} G_N \\ G_C \end{bmatrix} \begin{bmatrix} \pi_N^k \\ \pi_C^k \end{bmatrix}$$

The vector π_C^k demonstrates the pages that have already converged, so no further calculation is needed for π_C^k . The next iteration will be:

$$\pi_N^{(k+1)} = G_N \pi^k$$

$$\pi_C^{(k+1)} = \pi_C^k$$

Thus the computations of PageRank for already converged pages are not recomputed for the desired tolerance level.

Algorithm: 2 PageRank by Adaptive Method

```
function AdaptivePageRank(G,  $\pi^0$ )
repeat
 $\pi_N^{k+1} = G_N \pi^k$ ;
 $\pi_C^{k+1} = \pi_C^k$ ;
[N,C]=detectConverged( $\pi^k, \pi^{k+1}, \epsilon$ );
 $r = \|G\pi^k - \pi^k\|_1$ ;
until  $r < \epsilon$ ;
return  $\pi^{k+1}$ ;
```

Kamvar et.al [3] also proposed a Filter-Based Adaptive method for the computation of PageRank. The matrix G can be symbolized by Filter-based Adaptive Method as:

$$G' = \begin{bmatrix} G_N \\ \mathbf{0} \end{bmatrix},$$

So that the sub-matrix G_C replaced by zero matrix. The matrix G' is sparse as contrast to G, which computes PageRank in less time, since non-zero entries slow matrix multiplication. G' can be characterized as:

$$G'_{ij} = \begin{cases} 0, & \text{if } i \in C, \\ G_{ij}, & \text{otherwise} \end{cases}$$

The PageRank vector of the converged pages are build as:

$$\pi_i'^k = \begin{cases} \pi_i^k, & \text{if } i \in C \\ 0, & \text{otherwise.} \end{cases}$$

In matrix G' the converged pages have been replaced zero matrix, the PageRank of those pages must be added to the PageRanks of the nonconverged nodes for the present iteration.

Algorithm: 3 The Filter-Based Adaptive PageRank

```
Algorithm
function filterAdaptivePR(G,  $\pi^0$ )
repeat
 $\pi_N^{k+1} = G' \pi^k + \pi^k$ ;
periodically,
[N,C]=detectConverged( $\pi^k, \pi^{k+1}, \epsilon$ );
 $A' = \text{filter}(A', N, C)$ ;
 $\pi^k = \text{filter}(\pi^k, C)$ ;
periodically,  $r = \|G\pi^k - \pi^k\|_1$ ;
until  $r < \epsilon$ ;
return  $\pi^{k+1}$ ;
```

The Modified algorithm is more polished than the previous two algorithms introduced by Kamvar et.al. They further reduced repetitious computation of PageRank by excluding the recompilation of the nodes in N due to inlinks from C. The matrix G can be represented as:

$$G = \begin{bmatrix} G_{NN} & G_{NC} \\ G_{CN} & G_{CC} \end{bmatrix},$$

Where G_{NN} are the entries associated to links within nodes that have not converged to the preferred tolerance level. Correspondingly, G_{NC} are the entries associated to linked pages that have converged to that pages that have not converged and accordingly. Since the PageRank of a given page is the sum of the PageRanks of inlinking pages, calculations can be done for only nonconverged nodes that have inlinks. Computation of PageRank for matrices G_{NN} and G_{NC} .

Algorithm: 4 Modified Adaptive Method

```
function ModifiedAdaptivePR(G,  $\pi^0$ )
repeat
 $\pi_N^{k+1} = G_{NN} \pi_N^k + y$ ;
 $\pi_C^{k+1} = \pi_C^k$ ;
periodically,
[N,C]= detectConverged( $\pi^k, \pi^{k+1}, \epsilon$ );
 $y = G_{NC} \pi_C^k$ ;
periodically,  $r = \|G\pi^k - \pi^k\|_1$ ;
until  $r < \epsilon$ ;
return  $\pi^{k+1}$ ;
```

Here we figure two extrapolation techniques: Aitken's extrapolation and Quadratic extrapolation, both of which try to estimate the stationary distribution.

Assume we obtain $\pi^{(k)}$ after k iterations. And now we approximate it with the first two π, u_2 :

$$\pi^{(k)} = \pi u_2;$$

Then we use a mixture of the previous iterations to get a new estimation to the true eigenvector. As we favor $\pi^{(k+2)}$ is closer to the true value than $\pi^{(k)}$, so we substitute the latter with the former in our practice. Also, we have to re-normalize the computing outcome by setting as zero those negative substances caused by the subtraction job.

Approximating the iterations with the first three eigenvectors of the Google matrix, which is called the quadratic extrapolation.[6]

Algorithm: 5 Aitken's Extrapolation

```
function y = Aitkenpagerank( $\pi^{(k)}, \pi^{(k+1)}, \pi^{(k+2)}$ )
 $g = (\pi^{(k+1)} - \pi^{(k)})^2$ ;
 $h = \pi^{(k+2)} + 2\pi^{(k+1)} + \pi^{(k)}$ ;
 $f = g . h$ ;
 $y = \pi^{(k+2)} - f$ ;
renormalize y;
return
```

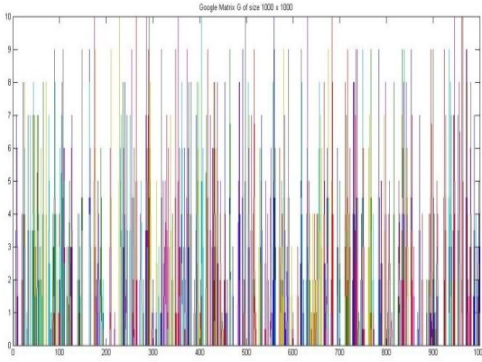


Fig 3: Google Matrix of 1000 x 1000

In the following, we have presented comparison Tables of Number of Iterations(Computational Time) for Estimating PageRank by Power Method, Adaptive Method and Aitken’s Extrapolation Method:

Table 1. Google Matrix of 50000 x 50000 with tolerance level of 1×10^{-8}

α	PageRank	Adaptive PageRank	Aitken’s PageRank
0.7	44(4.1)	44(18.4)	87(1.03)
0.85	50(4.1)	57(18.5)	NAN
0.9	53(5.3)	57(22.9)	NAN
0.99	58(6.4)	58(28.32)	NAN

Table 2. Tolerance level of 1×10^{-3}

α	PageRank	Adaptive PageRank	Aitken’s PageRank
0.7	24(1.9)	24(6.2)	38(4.1)
0.85	27(2.5)	27(6.5)	NAN
0.9	28(2.5)	34(19.4)	NAN
0.99	30(2.9)	34(24.3)	NAN

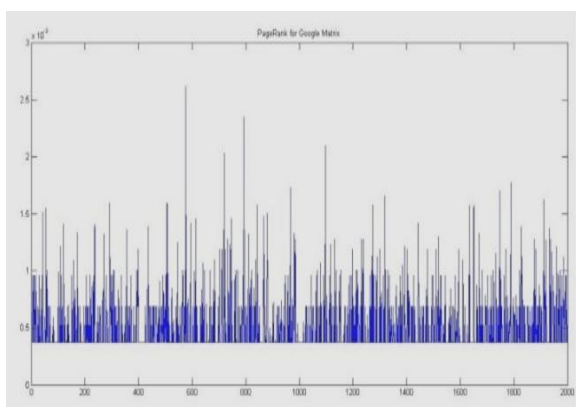


Fig 4: PageRank vector for Google Matrix

Table 3. Comparison Table of Number of iteration(Elapsed Time, in sec) for Estimating PageRank: Considering, Damping Factor $\alpha = 0.7$, Tolerance level $\epsilon = 1 \times 10^{-3}$

Matrix size $n \times n$	PageRank	Adaptive PageRank	Aitken’s PageRank
100	24(50)	25(119)	31(78.2)
200	26(62)	28(224)	97(570)
300	26(62)	28(334)	97(690)
400	27(112)	28(534.7)	365(383)
500	29(155.6)	28(668)	NAN
600	56(1099)	59(3445)	NAN
700	76(2549)	69(854.9)	NAN

Table 4. Comparison Table of Number of iteration(Elapsed Time) for Estimating PageRank: Considering, Damping Factor $\alpha = 0.85$, Tolerance level $\epsilon = 1 \times 10^{-3}$

Matrix size $n \times n$	PageRank	Adaptive PageRank	Aitken’s PageRank
100	24(0.5)	25(1.1)	72(3)
200	27(0.7)	28(1.6)	NAN
300	26(0.8)	33(2.3)	NAN
400	29(1.903)	33(4.5)	NAN
500	32(2.353)	35(7.9)	NAN
600	694(133.02)	430(624.8)	NAN
700	750(200)	650(850)	NAN

5. CONCLUSION AND FUTURE DIRECTIONS

Web search has become an integral part of modern information access, posing many interesting challenges in developing effective and efficient strategies for ranking search results. One of the most well-known Web-specific ranking algorithms is PageRank – a technique for computing the authoritativeness of web pages using the hyperlink graph of the Web. Although PageRank is largely an offline computation, performed while preprocessing and indexing a Web crawl before any queries have been issued, it has become increasingly desirable to speed up this computation. Rapidly growing crawl repositories, increasing crawl frequencies, and the desire to generate multiple topic-based PageRank vectors for each crawl are all motivating factors for our work in speeding up PageRank computation.

For each data size used in this study, the adaptive method required more iterations than Standard Power method, but the average cost in reckoning time for every iteration is lower for big data of web. Aitken’s method works appropriately on small values of damping factor α .

We aim to work on fuzzification of page rank algorithms as the graph connecting nodes are inherited with fuzziness due to search agent’s environmental uncertainties.

6. REFERENCES

- [1] Wills S.R, Google's PageRank: The Math Behind the Search Engine,2006
- [2] Lin, Shi and Wei, On computing PageRank via lumping the Google matrix, *Journal of Computational and Applied Mathematics* 2009; 224(2):702-708
- [3] Kamvar, Haveliwala and Golub, Adaptive methods for the computation of PageRank, *Linear Algebra and its applications*,(2004) 51-65.
- [4] Fu Hwai-Hui, Lin Dennis and Tsai Hsein-Tang, Damping Factor in Google page ranking, *Applied Stochastic Models in Business and Industry*,2006,Vol 22,431-444
- [5] Ipsen and Selee, PageRank computation, with special attention to Dangling Nodes, *SIAM J. MATRIX ANAL. APPL.*, (2007),1281-1296
- [6] Kamvar et. al, Extrapolation Methods for Accelerating PageRank Computations, *Twelfth International World Wide Web Conference (WWW 2003)*, May 20-24, 2003.
- [7] Wu and Wei, Arnoldi versus GMRES for computing pageRank: A theoretical contribution to Google's PageRank problem, *ACM Transactions on Information Systems (TOIS)* ,(2010),28 (3), 11
- [8] Yu, Miao, Wu and Wei, Lumping algorithms for computing Google's PageRank and its derivative, with attention to unreferenced nodes, *Inf Retrieval* (2012) 15:503–526
- [9] Langville and Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings*
- [10] Zhu,Yu and Li, Distributed PageRank computation based on iterative aggregation-disaggregation method, *CIKM '05 Proceedings of the 14th ACM international conference on Information and knowledge management*, (2005),pages 578-585.
- [11] Ipsen and Kirkland, *Convergence Analysis of an Improved PageRank Algorithm*
- [12] <http://mathfaculty.fullerton.edu/mathews/n2003/AitkenSteffensenMod.html>
- [13] http://ergodic.ugr.es/cphys/LECCIONES/FORTRAN/power_method.pdf
- [14] Anton and Rorres, *Elementary Linear Algebra*, 7th edition.
- [15] Pavel Berkhin, A survey on PageRank computing, *Internet Mathematics* 2 (2005), no. 1, 73–120.