Rayleigh quotient with bolzano booster for faster convergence of dominant eigenvalues

M Zainal Arifin [a,b,1] , Ahmad Naim Che Pee [b,2], Sarni Suhaila Rahim [b,3], Aji Prasetya Wibawa [a,4,*]

[a] Universitas Negeri Malang, Jl Semarang, Malang, Indonesia
[b] Universiti Teknikal Malaysia Melaka, Jl Durian Tunggal, Melaka, Malaysia
[1] arifin.mzainal@um.ac.id; [2] naim@utem.edu.my; [3] sarni@utem.edu.my; [4] aji.prasetya.ft@um.ac.id
* corresponding author

ARTICLE INFO

Article history
Received September 6, 2021
Revised January 8, 2022
Accepted January 14, 2022
Available online March 31, 2022

Keywords
PageRank
Optimization
Bolzano method
Rayleigh quotient
Eigenvalue

ABSTRACT

Computation ranking algorithms are widely used in several informatics fields. One of them is the PageRank algorithm, recognized as the most popular search engine globally. Many researchers have improved the ranking algorithm in order to get better results. Recent research using Rayleigh Quotient to speed up PageRank can guarantee the convergence of the dominant eigenvalues as a key value for stopping computation. Bolzano's method has a convergence character on a linear function by dividing an interval into two intervals for better convergence. This research aims to implant the Bolzano algorithm into Rayleigh for faster computation. This research produces an algorithm that has been tested and validated by mathematicians, which shows an optimization speed of a maximum 7.08% compared to the sole Rayleigh approach. Analysis of computation results using statistics software shows that the degree of the curve of the new algorithm, which is Rayleigh with Bolzano booster (RB), is positive and more significant than the original method. In other words, the linear function will always be faster in the subsequent computation than the previous method.

1. Introduction

Pagerank is an algorithm used to determine the ranking of a web page (called a node) democratically by constructing a graph between nodes and the reference direction from one node to another, represented by the Markov matrix. Referrals from these nodes can affect the rankings of other nodes. The more referrals a node receives, the more likely it has a higher rating than the others.

Internet users like this democratic ranking because it benefits, especially fairness in search results, but its calculation is complex. As the search ranking comparison is an opportunity, the computational results provide eigenvectors because the PageRank algorithm's ranking equals the matrix element. An eigenvector is a vector in which a single column has the value 1. The Pagerank algorithm calculates rankings using the power method, which iterates until the dominant eigenvalue is determined. The PageRank algorithm is utilized extensively in sports, election simulation, and education. Compared to other ranking algorithms, the disadvantage of this technique is that Pagerank requires a lengthy computation period. Hence, the calculations are performed offline for several days. For this, we need a method that expedites the ranking calculation.

To enhance ranking calculations, researchers have made improvisations. Research on the optimization of ranking computation using the PageRank algorithm has been carried out in several universities. Most of the research has computation results that can accelerate the achievement of the dominant eigenvalues by combining other algorithms, the Rayleigh quotient in PageRank algoritm that compute using Power method iteration. Rayleigh worked as a linear value and was inserted into the PageRank algorithm. PageRank algorithm computes using the power method so that Rayleigh will be inside the power method; hence, the dominant eigenvalues converge faster [1]. Several researchers have discussed the Rayleigh quotient: Siahaan [2], and Kuleshov [1]. Optimization of calculation time makes this matter continue to be studied because the size of the Markov matrix is quite large. The internet development and the increase of web pages increase the size of the Markov matrix and the smaller the probability of each node.

The Rayleigh Quotient obtained in the power method calculates the eigenvalues by comparing the eigenvectors in each iteration. The Rayleigh quotient helps the Power method to speed up ranking calculations. Rayleigh has the advantage of searching eigenvalues, and the convergence is getting better with the same result. However, the weakness of the Rayleigh quotient is that when the matrix size gets bigger, the vector comparison process will also take longer, so Rayleigh needs further optimization to help speed up the calculations.

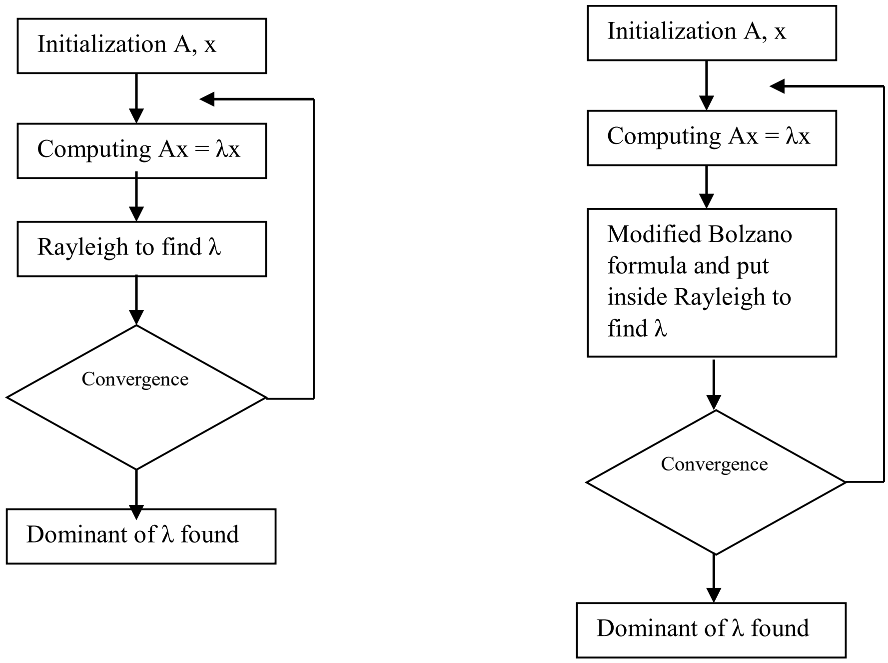
Currently, optimization efforts are still being carried out, which include: optimization by using nodes [3]–[18]using other methods such as gauss-seidel [4], changing the weight of each link associated with nodes, [13]–[18], and the last is using the extrapolation method [5]–[12]. Rayleigh optimization on the PageRank gives excellent and fast calculation results because the convergence becomes faster without changing the dominant eigenvalues [19]–[21]. This algorithm will be enhanced using the Bolzano method to provide more computational acceleration. This research uses several datasets taken from the Stanford Network Analysis Platform. The modified Bolzano method will be used inside Rayleigh Quotient to predict the next dominant eigenvalue on each iteration because Bolzano can guarantee that it is linear function will be convergence only in the linear formula and not yet applied on PageRank optimization [22], [23]. In addition, several studies have been carried out to accelerate the convergence of a linear line at a scalar value to support these research objectives with the Bolzano method. Rayleigh Quotient produces a dominant eigenvalue from each iteration on Power Method, but its eigenvalue needs to improve using the prediction value from Bolzano Method. The modified Bolzano entered into the Rayleigh quotient, assuming that the method can make the convergence of the dominant eigenvalues in the Rayleigh quotient faster without changing the corresponding eigenvalues and eigenvectors.

2. Method

This research approach employs Bolzano's formula, which has been established and then integrated into Rayleigh so that the findings are consistent with the hypothesis. However, this phase of the proof is not straightforward. It takes a mathematical method to demonstrate that this proof is correct and legitimate. In this study, Fig. 1 compares the old approach and the novel method, highlighting the differences between the two.

At the initialization time, the computer determines the initial value of the eigenvector x, which will be iteratively calculated until convergence. In the old technique, the Rayleigh quotient is utilized to identify the dominating value of λ. Still, in the most recent method given in this study, the Bolzano method aids in determining the value of λ to do faster computing.

The research method is divided into several steps: finding each character of the two functions between Rayleigh and Bolzano, finding the intersection, making adjustments to Bolzano to make it easier to work on Rayleigh, proving by mathematical induction, and validating the formulas that have been developed. Fig. 2 shows a fundamental understanding of the Bolzano and Rayleigh formulas is required. Both are characterized by empirical evidence and have a linear nature in which the function performs a basic linear line calculation, such as $y = ax + b$. When the characteristics of both formulas are known, an intersection can be drawn between them to demonstrate that they share the same characteristics. In addition, the intersection of the two will aid in formula proofs that adhere to mathematical standards.



(a) (b)

Fig. 1. Comparison between the old (a) and the new method (b).

When the intersection is known, the next step is to demonstrate using mathematical induction if they are valid for any value of n, where n is the size of the square matrix used in the iteration calculation. Induction demonstrates that the two qualities are compatible, but mathematical validation is also required to demonstrate that the two formulas may be combined. After the validation is complete, the new formula, a combination of the Bolzano technique and the Rayleigh quotient, can be validated using a dataset discussed in the following step.

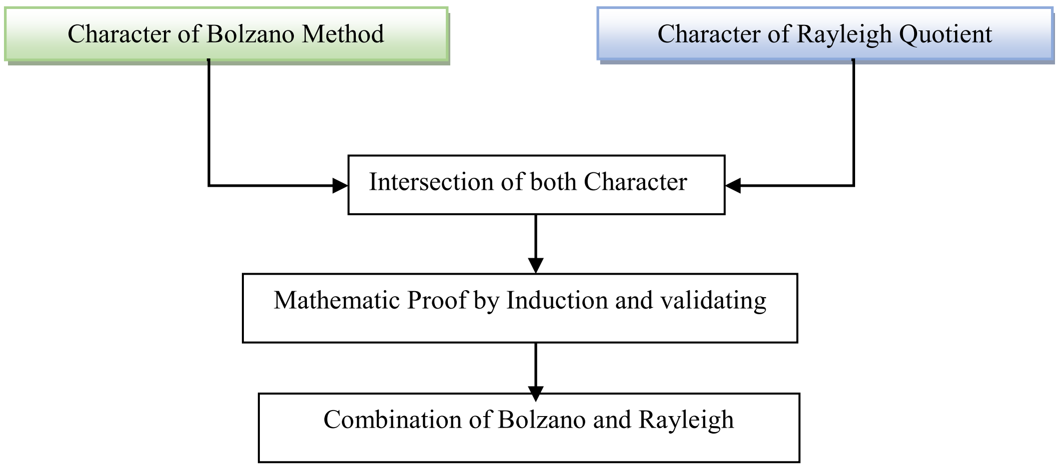


Fig. 2. Combining Bolzano and Rayleigh Quotient.

2.1. Sensitivity of Linear System

Sensitivity is a way of knowing whether the influence of a scalar outside the Bolzano algorithm will affect the result of an eigenvector [24], [25], [26], [27]. In this research, sensitivity was carried out on the Bolzano method, Rayleigh quotient, and the results of the combination of both algorithms. When a mathematical equation is multiplied by a scalar, the scalar will affect the final result [28], [29], but in this research, the final result is the eigenvector obtained when the iteration stops because the dominant eigenvalue has been reached.

The variable used to determine this sensitivity is α, with a value of α = 0.85 obtained from a previous study by Page [30]. The effect of the value of α is expected not to change the dominant value of the eigenvalue and only affect outside the eigenvector. Pseudocode of dominant computing eigenvalue can be seen as in Fig. 3.

```
1. Given markov matrix A and x0
2. epsilon = 10-8
3. repeat :
4.       yk = Axk
5.       xk+1 = yk/norm(yk)
6.       rk = xk+1(Axk+1)
7. until norm(Axk-rkxk) < epsilon
```

Fig. 3. Dominant computing eigenvalue

The calculation will converge within the epsilon limit so that the Markov A matrix is expected to have properties that can help achieve convergence. Matrix A is not a natural and democratic matrix if convergence cannot be obtained.

The formula in finding eigenvectors by iteration on dominant eigenvalues λ on an equation $\mathrm{Ax} = \lambda\mathrm{x} = \mathrm{b}$ with A is Markov matrix, x, b is an eigenvector, the value of x and b and its approximation x^* and b^* then $\frac{\|x-x^*\|}{\|x\|}$ and $\frac{\|b-b^*\|}{\|b\|}$ is the relative error. It is using the equation to find the residual, $r = b - Ax^*$, resulting in the approximate value of the ratio $\frac{\|b\|}{\|x\|} = \frac{\|Ax\|}{\|x\|}$, which causes that inserting noise into the formula does not affect the result of the eigenvector.

2.2. Convergence Criteria

The Bolzano method guarantees that convergence will be achieved because it works on a linear model that convergence to a number, but not with Rayleigh. Therefore, it is necessary to prove that this convergence will hold when the modified Bolzano is included in Rayleigh, resulting in a convergent character approach. From the formula $Ax = \lambda x = b$ then it can be reduced to (1).

$$\lambda = \frac{x^T Ax}{x^T x} = \frac{scalar}{scalar} = scalar \tag{1}$$

The equation explains that the dominant eigenvalue will converge to a scalar value. This is under the Bolzano nature, making the function converge to a scalar from a point in a specific range.

3. Results and Discussion

Bolzano helps Rayleigh algorithm to achieve better convergence by taking a scalar approximation. The noise included in the algorithm does not seem to affect the results of the eigenvectors, so the convergence of Rayleigh and Bolzano to scalars is proven to be linearly sensitive. After the formula is combined by making some modifications to the Bolzano method, the next step is to test the algorithm on a computer with 32GB RAM, and each dataset is run at least 1000 trials or under particular conditions.

3.1. The Bolzano Booster

Implementing an algorithm with another algorithm cannot be done directly. This requires proving that the algorithm can work well, especially if the algorithm requires modification to give good results when combined. In this research, the Bolzano algorithm is modified so that the convergence of the dominant eigenvalues of the Rayleigh quotient can be obtained more quickly as an effect of the Bolzano modification combined with Rayleigh using a scalar as in (2) and psudocode (Fig. 4).

$$\mathrm{RQI} = \lambda = \frac{\mathrm{x^T A x}}{\mathrm{x^T x}} \tag{2}$$

```
1. function RQI_Bolzano
2. v^(0) = some vector with ||v^(0)|| = 1
3. λ^(0) = (v^(0))^T A v^(0) = correspoding Rayleigh quotient
4. While ε < 10^-8
5.          Solve (A − λ^(k−1) I) w = v^(k−1), for w
6.          v^(k) = w/||w||
7.          λ^(k) = (v^(k))^T A v^(k)  (Rayleigh Quotient)
8.          λ_bolzano = λ^(k) + ||λ^(k) − λ^(k−1)|| α
9.          ε = ||λ^(k) − λ_bolzano||
```

Fig. 4. RQI Bolzaono function

If x is a vector in vector space V, then x can be said to be a linear combination of vectors x_1, x_2, x_3, ..., x_n in V can be written in the form $x = c_1 x_1 + c_2 x_2 + c_3 x_3 + \dots + c_n x_n$ Where c_1, c_2, c_3, ..., c_n is a scalar of the linear combination. Then the equation is linear with degree one.

$$\lambda_{k+1} \approx 2\lambda_k + \lambda_{k-1} \tag{3}$$

$$\lambda_{k+1} \approx \lambda_k + (\lambda_k - \lambda_{k-1}) \tag{4}$$

$$\lambda_{k+1} \approx \lambda_k + |\lambda_k - \lambda_{k-1}| \tag{5}$$

$$\lambda_{k+1} = \lambda_k + |\lambda_k - \lambda_{k-1}|\alpha \tag{6}$$

From the equation, it appears that the next dominant eigenvalue (6) is a value that is influenced by the last eigenvalue and the current value. From (6), the resulting eigenvalue, according to the Bolzano method's character, is an equation with a value of $\alpha=0.85$ following previous research.

3.2. Testing of Algorithm

After obtaining the results of the modification of the Bolzano method for the Rayleigh quotient, the next step is to test that the formula can be used properly. The test is carried out in two ways: (1) using mathematical induction, which is checked by mathematicians consisting of 2 people, (2) using matlab to run the new formula with an existing dataset. Table 1 shows the calculation results using Matlab. The column of *nnz* is for non zero and *nodes* column is for the number of web pages or elements of matrix markov.

Table 1. Preparation Dataset

Dataset	preparation dataset			sparsity		
	size (MB)	*nodes*	*load mem(s)*	*nodes*	*nnz*	*zero cols*
wiki-talk-temporal	169	1,140,149	49.282649	1,140,149	3,309,592	888,995
wiki-topcats	412	1,791,489	157.050748	1,791,489	28,511,807	0
wiki-Talk	69	2,394,385	33.825236	2,394,385	5,021,410	2,246,783
sx-stackoverflow	1,599	2,601,977	370.078845	6,024,271	36,233,450	3,798,028
cit-Patents	273	3,774,768	124.334527	6,009,555	16,518,948	3,920,210
soc-LiveJournal1	1,000	4,847,571	434.833407	4,847,571	68,993,772	539,119
sinaweibo	2,000	21,000,000	1768.333232	58,655,850	261,321,071	15,445

This dataset has been gradually loaded into Matlab memory. Each loaded dataset is recorded attributes that appear to perform further analysis. The “sinaweibo” dataset with the most significant number of nodes requires more computation time and vice versa.

The dataset in the Table 1 shows that the wikitalk temporal dataset has 128 iterations to achieve convergence with *nnz* = 3,309,592. The number of iterations in the algorithm gives the same calculation results. The Table 2 provides comparative information between calculations using the Rayleigh and Bolzano modification.

Table 2. Result of Wikitalk-temporal dataset computation

Attributes	Rayleigh	RB
Σ trials	1000	1000
Best case (s)	3.374253	3.135263
λ	0.99999666	0.99999666
error rate	0.0000001	0.0000001
best on iteration	446	875

Table 2 shows that from 1000 experiments carried out, with each experiment going through 128 iterations, the Bolzano method's calculation time is faster than using the Rayleigh method alone. In contrast, the value of and error rate have the same value. The computation result uses the same attributes to determine whether the behavior of the formula can be compared. To better understand the condition of the eigenvectors after iterations between the two methods, Table 3 provides a statistical description.

Table 3. Statistics of Wikitalk-temporal dataset computation

Attributes	Rayleigh	RB
min	3.37425	3.13526
max	3.98574	4.01611
median	3.70394	3.72968
mean	3.69934	3.69234
modus	3.37425	3.13526
std	0.09596	0.15408
var	0.00921	0.02374
regression	0.005554916	0.005470106
gradient	0.005554859	0.005470051

In the Table 3, the most crucial information is on the gradient, where the slope angle of the regression curve later turns out that Bolzano has a smaller angle than Rayleigh. It can be concluded that the more nodes in the Markov matrix, the Bolzano angle value will undoubtedly be smaller than Rayleigh, who provides knowledge new that RB is faster than Rayleigh original formula. The calculation time of the ranking is also influenced by *nnz*. Fig. 5 explains the non-zero value in the Markov matrix.

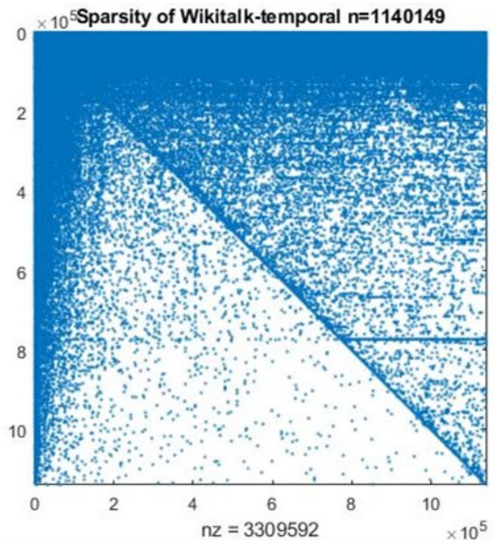


Fig. 5. Sparsity of Wikitalk-temporal matrix.

In Fig. 5, a white or unshaded region indicates that the Markov A matrix has a value of zero. The area with the shaded dots indicates a non-zero value, or the website is linked to another website or webpage. This sparsity matrix influences the calculation time. Hence, this study employs the same data set to compare the old (Rayleigh) method to the new one (RB).

From the explanation of one temporal wikitalk dataset in Table 3, a complete description of the results of the other running datasets is shown in Table 4.

Table 4. Result of dataset computation

Dataset	trials on	nodes	nnz	zero cols	Rayleigh(s)	RB(s)
wiki-talk-temporal	1000	1,140,149	3,309,592	888,995	3.374253	3.135263
wiki-topcats	1000	1,791,489	28,511,807	0	0.301992	0.28826
wiki-Talk	1000	2,394,385	5,021,410	2,246,783	2.204261	2.194955
sx-stackoverflow	25	2,601,977	36,233,450	3,798,028	149.5025	147.5919
cit-Patents	1000	3,774,768	16,518,948	3,920,210	5.481925	5.470497
soc-LiveJournal1	50	4,847,571	68,993,772	539,119	290.0422	289.4445
sinaweibo	10	21,000,000	261,321,071	15,445	196.5554	195.9187

Thus, it is clear that a dataset with a large number of *nnz* takes a long time to load into memory so the number of trials cannot reach 1000 experiments. In this experiment, one trial takes the time indicated by the Rayleigh column and the Bolzano column within the Rayleigh column, with the outcome expressed in seconds.

As an example of sinaweibo dataset, in the results of Rayleigh's calculation, the iteration time is 196.5554 seconds, while the modified RB produces a calculation time of 195.9187. There is a time difference of 0.6367 seconds. Roughly speaking, if the number of nodes crawled by search engines is currently 100 Billion pages, then the difference in the calculation time of the two methods is about 3.032 seconds or 50 minutes 40 seconds. The wiki-talk-temporal dataset shows the highest difference in calculations of the two algorithms. The difference between the RB and the Rayleigh is 0.23899 seconds (7.08% acceleration) for the dataset size is 1,140,149.

4. Conclusion

Rank calculations using the PageRank algorithm have developed, and recent research has optimized the Rayleigh quotient in the power method to accelerate computation time. The Bolzano method, similar to Rayleigh's in finding a convergence point solution, can provide insight into how the two functions can be combined. The properties of Rayleigh and Bolzano with the same linearity sensitivity provide confidence that they can work well even though there is noise in the dataset with the formula. After the characters of the two functions are known to be similar, then the proof of the formula is done by induction so that the formula is valid, the new Bolzano method inside Rayleigh as $\lambda_{k+1} = \lambda_k + |\lambda_k - \lambda_{k-1}|\alpha$. The study results provide a new formula that can speed up ranking calculations up to 7.08%, although further research is still needed on the percentage of calculation acceleration.

Acknowledgment

Researchers received assistance in suggestions and advice from several other researchers. Thanks to Prof David F. Gleich from Purdue University, Prof Michael Saunders from Stanford University, Prof Dr Marjono, Saiful Anam Ph.D from Brawijaya University, and Prof Nanna Suryana from Universiti Teknikal Malaysia Melaka.

Declarations

Author contribution. The formula design and mathematical proof were done by M. Zainal Arifin and Prof. Naim Che Pee, while the algorithm analysis and testing was designed by Dr. Sarni Suhaila Rahim and Dr. Aji Prasetya Wibawa.

Funding statement. This research received publication support from the Universitas Negeri Malang and Research Facilities from the Universiti Teknik Malaysia Melaka.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] V. Kuleshov, "Fast algorithms for sparse principal component analysis based on Rayleigh quotient iteration," *30th Int. Conf. Mach. Learn. ICML 2013*, vol. 28, no. 3, pp. 2468–2475, 2013. Available: Google Scholar.

[2] D. Siahaan and Z. Arifin, "Implementing quotient rayleigh in power method to improve the computation speed of Pagerank," *Asian J. Inf. Technol.*, vol. 8, no. 4, pp. 104–111, 2009. Available: Google Scholar.

[3] J. H. Kim, "Efficient Node Proximity and Node Significance Computations in Graphs," *ProQuest Diss. Theses*, p. 198, 2017, [Online]. Available: https://search.proquest.com/docview/1957948842?accountid=188395.

[4] D. Silvestre, J. Hespanha, and C. Silvestre, "A PageRank Algorithm based on Asynchronous Gauss-Seidel Iterations," *2018 Annu. Am. Control Conf.*, vol. 2018-June, pp. 484–489, Jun. 2018, doi: 10.23919/ACC.2018.8431212.

[5] Z.-L. Luo, W.-D. Cai, Y.-J. Li, and D. Peng, "A PageRank-Based Heuristic Algorithm for Influence Maximization in the Social Network," in *Lecture Notes in Electrical Engineering*, vol. 157, pp. 485–490, 2012, doi: 10.1007/978-3-642-28798-5_65.

[6] P. Berkhin, "A Survey on PageRank Computing," *Internet Math.*, vol. 2, no. 1, pp. 73–120, Jan. 2005, doi: 10.1080/15427951.2005.10129098.

[7] K. Sugihara, "Beyond Google's PageRank: Complex Number-based Calculations for Node Ranking," *Glob. J. Comput. Sci. Technol.*, vol. 19, no. 3E, pp. 1–12, Oct. 2019, doi: 10.34257/GJCSTEVOL19IS3PG1.

[8] D. F. Gleich, L.-H. Lim, and Y. Yu, "Multilinear PageRank," *SIAM J. Matrix Anal. Appl.*, vol. 36, no. 4, pp. 1507–1541, Jan. 2015, doi: 10.1137/140985160.

[9] X. Wang, J. Ma, K. Bi, and Z. Li, "A Improved PageRank Algorithm Based on Page Link Weight," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8630, no. 1, pp. 720–727, 2014, doi: 10.1007/978-3-319-11197-1_57.

[10] S. Luo, "Distributed PageRank Computation: An Improved Theoretical Study," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, pp. 4496–4503, Jul. 2019, doi: 10.1609/aaai.v33i01.33014496.

[11] A. N. Langville and C. D. Meyer, *Google's PageRank and beyond: The science of search engine rankings*. 2011. Available: Google Scholar.

[12] H.-F. Zhang, T.-Z. Huang, C. Wen, and Z.-L. Shen, "FOM accelerated by an extrapolation method for solving PageRank problems," *J. Comput. Appl. Math.*, vol. 296, no. C, pp. 397–409, Apr. 2016, doi: 10.1016/j.cam.2015.09.027.

[13] A. K. Srivastava, M. Srivastava, R. Garg, and P. K. Mishra, "An Aitken-extrapolated Gauss-Seidel method for computing PageRank," *J. Stat. Manag. Syst.*, vol. 22, no. 2, pp. 199–222, Feb. 2019, doi: 10.1080/09720510.2019.1580901.

[14] G. Liu, "An adaptive improvement on PageRank algorithm," *Appl. Math. J. Chinese Univ.*, vol. 28, no. 1, pp. 17–26, Mar. 2013, doi: 10.1007/s11766-013-3123-9.

[15] J. Lei and H.-F. Chen, "Distributed Randomized PageRank Algorithm Based on Stochastic Approximation," *IEEE Trans. Automat. Contr.*, vol. 60, no. 6, pp. 1641–1646, Jun. 2015, doi: 10.1109/TAC.2014.2359311.

[16] N. Chen, N. Litvak, and M. Olvera-Cravioto, "Generalized PageRank on directed configuration networks," *Random Struct. Algorithms*, vol. 51, no. 2, pp. 237–274, Sep. 2017, doi: 10.1002/rsa.20700.

[17] C. Gu, F. Xie, and K. Zhang, "A two-step matrix splitting iteration for computing PageRank," *J. Comput. Appl. Math.*, vol. 278, pp. 19–28, Apr. 2015, doi: 10.1016/j.cam.2014.09.022.

[18] D. F. Gleich, "PageRank Beyond the Web," *SIAM Rev.*, vol. 57, no. 3, pp. 321–363, Jan. 2015, doi: 10.1137/140976649.

[19] V. Dinh, L. S. T. Ho, D. Nguyen, and B. T. Nguyen, "Fast learning rates with heavy-tailed losses," *Adv. Neural Inf. Process. Syst.*, pp. 505–513, Sep. 2016, [Online]. Available: http://arxiv.org/abs/1609.09481.

[20] M. I. Ortega, R. N. Slaybaugh, P. N. Brown, T. S. Bailey, and B. Chang, "A Rayleigh quotient method for criticality eigenvalue problems in neutron transport," *Ann. Nucl. Energy*, vol. 138, p. 107120, Apr. 2020, doi: 10.1016/j.anucene.2019.107120.

[21] B. J. Yang, "Dominant eigenvector and eigenvalue algorithm in sparse network spectral clustering," *Lat. Am. Appl. Res. - An Int. J.*, vol. 48, no. 4, pp. 323–328, Oct. 2018, doi: 10.52292/j.laar.2018.248.

[22] P. I. Frazier, S. G. Henderson, and R. Waeber, "Probabilistic Bisection Converges Almost as Quickly as Stochastic Approximation," *Math. Oper. Res.*, vol. 44, no. 2, pp. 651–667, May 2019, doi: 10.1287/moor.2018.0938.

[23] M. Mahrudinda, D. Munandar, and S. Purwani, "Efficiency and Convergence of Bisection, Secant, and Newton Raphson Methods in Estimating Implied Volatility," *World Sci. News*, vol. 153, no. 2, pp. 157–168, 2021, [Online]. Available: http://psjd.icm.edu.pl/psjd/element/bwmeta1.element.psjd-a6a50cb9-4815-4ac7-b8ea-c69cf6790fbb.

[24] Q. Yang and X. Peng, "An Exact Method for Calculating the Eigenvector Sensitivities," *Appl. Sci.*, vol. 10, no. 7, p. 2577, Apr. 2020, doi: 10.3390/app10072577.

[25] D. Ruiz, J. C. Bellido, and A. Donoso, "Eigenvector sensitivity when tracking modes with repeated eigenvalues," *Comput. Methods Appl. Mech. Eng.*, vol. 326, pp. 338–357, Nov. 2017, doi: 10.1016/j.cma.2017.07.031.

[26] G. H. Yoon, A. Donoso, J. Carlos Bellido, and D. Ruiz, "Highly efficient general method for sensitivity analysis of eigenvectors with repeated eigenvalues without passing through adjacent eigenvectors," *Int. J. Numer. Methods Eng.*, vol. 121, no. 20, p. nme.6442, Jun. 2020, doi: 10.1002/nme.6442.

[27] S. Noschese and L. Reichel, "Eigenvector sensitivity under general and structured perturbations of tridiagonal Toeplitz-type matrices," *Numer. Linear Algebr. with Appl.*, vol. 26, no. 3, May 2019, doi: 10.1002/nla.2232.

[28] A. H.-D. Cheng, "Multiquadric and its shape parameter—A numerical investigation of error estimate, condition number, and round-off error by arbitrary precision computation," *Eng. Anal. Bound. Elem.*, vol. 36, no. 2, pp. 220–239, Feb. 2012, doi: 10.1016/j.enganabound.2011.07.008.

[29] J. H. Wilkinson, "Error analysis of floating-point computation," *Numer. Math.*, vol. 2, no. 1, pp. 319–340, Dec. 1960, doi: 10.1007/BF01386233.

[30] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," *World Wide Web Internet Web Inf. Syst.*, vol. 54, no. 1999–66, pp. 1–17, 1998, doi: 10.1.1.31.1768.