数理解析研究所講究録
第 1995 巻 2016 年 42-47

A Riccati approach to computing PageRank

Takafumi Miyata
Fukuoka Institute of Technology

Abstract

PageRank is for ranking Web pages and plays an important role in the Google search engine. In recent years, large-scale computations on PageRank are attractive research topics of numerical linear algebra and its application. The existing approaches include Krylov subspace methods such as the Arnoldi-type method, which is the efficient variant of the Arnoldi method for eigenvalue problems. The Arnoldi method requires complex arithmetic, whereas the Arnoldi-type method does not, and thus, can compute PageRank efficiently. We focus on the non-Krylov subspace method called the Riccati method and present its variants for computing PageRank. The proposed Riccati-type methods can efficiently compute PageRank without complex arithmetic, while keeping the attractive convergence behavior.

1 Introduction

PageRank is for ranking Web pages and large-scale computations on it has attracted much attention in recent years [5, 8, 10, 11]. The PageRank computation results in an eigenvalue problem of an $n \times n$ real nonsymmetric matrix A. Here, A is determined by link structures between Web pages. The number of Web pages n can be large, while each Web page often has a few hyperlinks. Thus, A is large and sparse. In addition, A is positive and column stochastic. From the Perron–Frobenius theorem, it follows that the largest eigenvalue of A in magnitude is simple and equal to 1 and its corresponding positive eigenvector exists [5]. This eigenpair is denoted by $(\lambda, \boldsymbol{x})$, where $\lambda = 1$ and $x_j > 0$. When $\|\boldsymbol{x}\|_1 = 1$, the eigenvector $\boldsymbol{x}$ defines a PageRank vector, where x_j shows the ranking of the jth page [5, 10].

The important point to note here is the influence of a parameter α $(0 < \alpha < 1)$ in A. As reported in [6], rankings of Web pages x_j change with α. Higher values of α will give true PageRank vectors [8]. Meanwhile, α determines the difficulty of computing $\boldsymbol{x}$. It is shown that $|\widetilde{\lambda}| \leq \alpha$, where $\widetilde{\lambda}$ is the second largest eigenvalue of A in magnitude [5]. When α is close to 1, λ will not be well separated from other eigenvalues. Consequently, more iterations are taken to compute $\boldsymbol{x}$ (corresponding to λ) by simple means like the power method, and thus faster methods are required.

2 The Arnoldi variants for computing PageRank

The Arnoldi method [1, 2] shows faster convergence than the power method. However, the Arnoldi method computes Ritz values of A, which will be complex numbers. To avoid complex arithmetic, the Arnoldi-type method was proposed [6]. In this method, the known eigenvalue λ is utilized as a fixed shift to compute a refined Ritz vector [7]. This vector denoted by $\boldsymbol{u}$ is generated so that it minimizes a relative residual norm

$$\frac{\|A\boldsymbol{u}-\boldsymbol{u}\|_2}{\|\boldsymbol{u}\|_2}, \qquad \boldsymbol{u} \in \mathcal{K}_k(A, \boldsymbol{u}_0), \tag{1}$$

where $\mathcal{K}_k(A, \boldsymbol{u}_0)$ is the k ($\ll n$) dimensional Krylov subspace with respect to A and an initial vector $\boldsymbol{u}_0 \in \mathbb{R}^n$. Since $\boldsymbol{u}$ can be computed via a singular value decomposition of a real matrix [6], the Arnoldi-type method requires no complex arithmetic. The method will stop if a norm of the residual $\boldsymbol{r} = A\boldsymbol{u} - \boldsymbol{u}$ is enough small. Otherwise, $\boldsymbol{u}$ is set to $\boldsymbol{u}_0$ as a new initial vector, and then the next iteration will start. As reported in [6], the Arnoldi-type method shows faster convergence than the power method, in particular, when α is close to 1.

A variant of the Arnoldi-type method was recently proposed [12]. This variant has a weight of inner products as a parameter, while keeping the advantage of the Arnoldi-type method. Let $\tilde{\boldsymbol{u}}_0 \in \mathbb{R}^n$ be an approximate eigenvector and $\tilde{\boldsymbol{r}}$ be its residual vector $\tilde{\boldsymbol{r}} = A\tilde{\boldsymbol{u}}_0 - \tilde{\boldsymbol{u}}_0$. Instead of (1), the variant computes a new approximate eigenvector $\tilde{\boldsymbol{u}}$ so that it minimizes a weighted residual norm

$$\frac{\|A\tilde{\boldsymbol{u}}-\tilde{\boldsymbol{u}}\|_W}{\|\tilde{\boldsymbol{u}}\|_W}, \qquad \tilde{\boldsymbol{u}} \in \mathcal{K}_k(A, \tilde{\boldsymbol{u}}_0), \tag{2}$$

where W is an $n \times n$ positive diagonal matrix whose jth diagonal element is

$$w_{jj} = \frac{|\tilde{r}_j|}{\|\tilde{\boldsymbol{r}}\|_1}. \tag{3}$$

Since each element of $\tilde{\boldsymbol{u}}_0$ will converge to each ranking of Web pages, a larger jth element $\tilde{r}_j$ means the slower convergence of the jth ranking $\tilde{u}_j$. Through the minimization of (2), W determined in (3) leads to update $\tilde{u}_j$ much more. Since W changes per iteration, the method is called the adaptively accelerated Arnoldi method. When $W = I$, the method results in the Arnoldi-type method. Compared to the Arnoldi-type method, the adaptively accelerated Arnoldi method requires to store one more n-vector for diagonal elements of W and more computations on W-weighted norms per iteration.

3 New Riccati variants for computing PageRank

We consider an alternative approach to computing PageRank using the Riccati method [3], because the method often shows faster convergence than the Arnoldi method. The Riccati method iteratively computes approximate eigenvectors. In the method, a Riccati equation needs to be solved to generate subspaces. Usually, the equation is approximately solved. To be more precise, a Riccati equation is reduced to an eigenvalue problem of a small non-symmetric matrix so that an approximate solution of the Riccati equation can be easily generated from an eigenvector. The reduction step gives not only a way to solve a Riccati equation approximately, but also the relationship between the Riccati method and the Arnoldi method. Implicitly, the Riccati method uses approximate eigenvectors computed by the Arnoldi method to expand subspaces. This relationship motivates to consider a Riccati-type method for computing PageRank efficiently. To avoid complex arithmetic in the Riccati method, we use the Arnoldi-type method instead of the Arnoldi method as follows. Let us assume that $(m-1)$ dimensional subspace $\mathcal{Z}_{m-1}$ is given. Here, the Arnoldi-type method is used to expand $\mathcal{Z}_{m-1}$ to

$$\mathcal{Z}_m = \mathcal{Z}_{m-1} + \mathrm{span}\{\boldsymbol{u}\}, \qquad \boldsymbol{u} \in \mathcal{K}_k(A, \boldsymbol{z}_0), \tag{4}$$

where $\boldsymbol{z}_0 \in \mathbb{R}^n$ is a starting vector for the k-dimensional Krylov subspace and $\boldsymbol{u} \in \mathbb{R}^n$ is an approximate eigenvector computed by the Arnoldi-type method. In this subspace $\mathcal{Z}_m$, a new approximate eigenvector $\boldsymbol{z} \in \mathbb{R}^n$ is generated so that it minimizes a relative residual 2-norm

$$\frac{\|A\boldsymbol{z} - \boldsymbol{z}\|_2}{\|\boldsymbol{z}\|_2}, \qquad \boldsymbol{z} \in \mathcal{Z}_m. \tag{5}$$

Reusing computational results in the Arnoldi-type method enables us to compute $\boldsymbol{z}$, which minimizes (5), without matrix-vector multiplications [9]. This Riccati-type method can compute PageRank without complex arithmetic, while keeping the attractive convergence behavior, as reported in the next section.

Through the above approach, we consider a variant of the Riccati-type method. Instead of the Arnoldi-type method, we use the adaptively accelerated Arnoldi method to generate subspaces. To be more precise, we replace $\boldsymbol{u}$ in (4) by an approximate eigenvector $\tilde{\boldsymbol{u}}$ computed by the adaptively accelerated Arnoldi method. In addition, we update the weight W for the adaptively accelerated Arnoldi method per iteration. These simple modifications enables us to derive another Riccati-type method. The derivation of this variant is straightforward and indicates that our approach can be flexibly incorporated with several approaches based on the Arnoldi-type method.

4 Numerical experiments

We report numerical experiments to compare the Riccati-type method **(R1)** and its variant **(R2)** with the Arnoldi-type method **(A1)** and the adaptively accelerated Arnoldi method **(A2)**. Computational environment is Fortran 77 double precision arithmetic run under Linux with Intel Xeon E5-1680 v3 (3.2 GHz). Our codes are compiled by Intel Fortran compiler (ver. 15.0.3) with an option -fast. Test matrix A comes from *Stanford* ($n = 281,903$) obtained from the matrix collection [4]. The parameter of A is set to $\alpha = 1 - 10^{-j}$ ($j = 1, \ldots, 4$). In all the methods, initial vectors are given by $[1, \ldots, 1]^{\mathrm{T}}$. These methods stop if a relative residual 1-norm is less than 10^{-7}, since the PageRank vector is the eigenvector normalized by 1-norm. To compare the methods under the same condition, the major part of memory usage is common to all the methods and kept to be $10n$ as follows. In A1, the dimension of Krylov subspace is set to $k = 10$. In A2, $k = 9$ due to store the weight W. In R1 and R2, two n-vectors need to be stored additionally per iteration [9]; one is for a basis vector to expand the subspace $\mathcal{Z}_m$, and the other is a workspace. To keep memory requirements under constant ($10n$ in this experiment), the dimension of Krylov subspace k is decreased by 2 per iteration. In R1, $k = 10, 8, 6, 4$ at each iteration, and after the 4 steps R1 restarts. In R2, $k = 9, 7, 5, 3$ at each iteration due to store the weight W. Table 1 shows computational results.

Table 1: Computational results of A1 (Arnoldi-type), A2 (adaptively accelerated Arnoldi), R1 (Riccati-type), and R2 (variant of Riccati-type). MV and Time show the number of matrix-vector multiplications taken for convergence and computational time (sec.).

	$\alpha = 0.9$		$\alpha = 0.99$		$\alpha = 0.999$		$\alpha = 0.9999$	
Method	MV	Time	MV	Time	MV	Time	MV	Time
A1	90	1.11	470	5.00	1550	15.38	5940	58.09
A2	81	1.07	324	3.72	1026	11.10	2700	28.86
R1	74	0.93	290	3.23	682	7.09	2138	21.40
R2	81	1.16	312	3.67	717	7.78	1912	20.25

Table 1 shows better performances of the Riccati-type method and its variant, in particular, when α is close to 1. The convergence behaviors are illustrated in Figure 1. For higher values of α, the Riccati-type method and its variant show faster convergence.

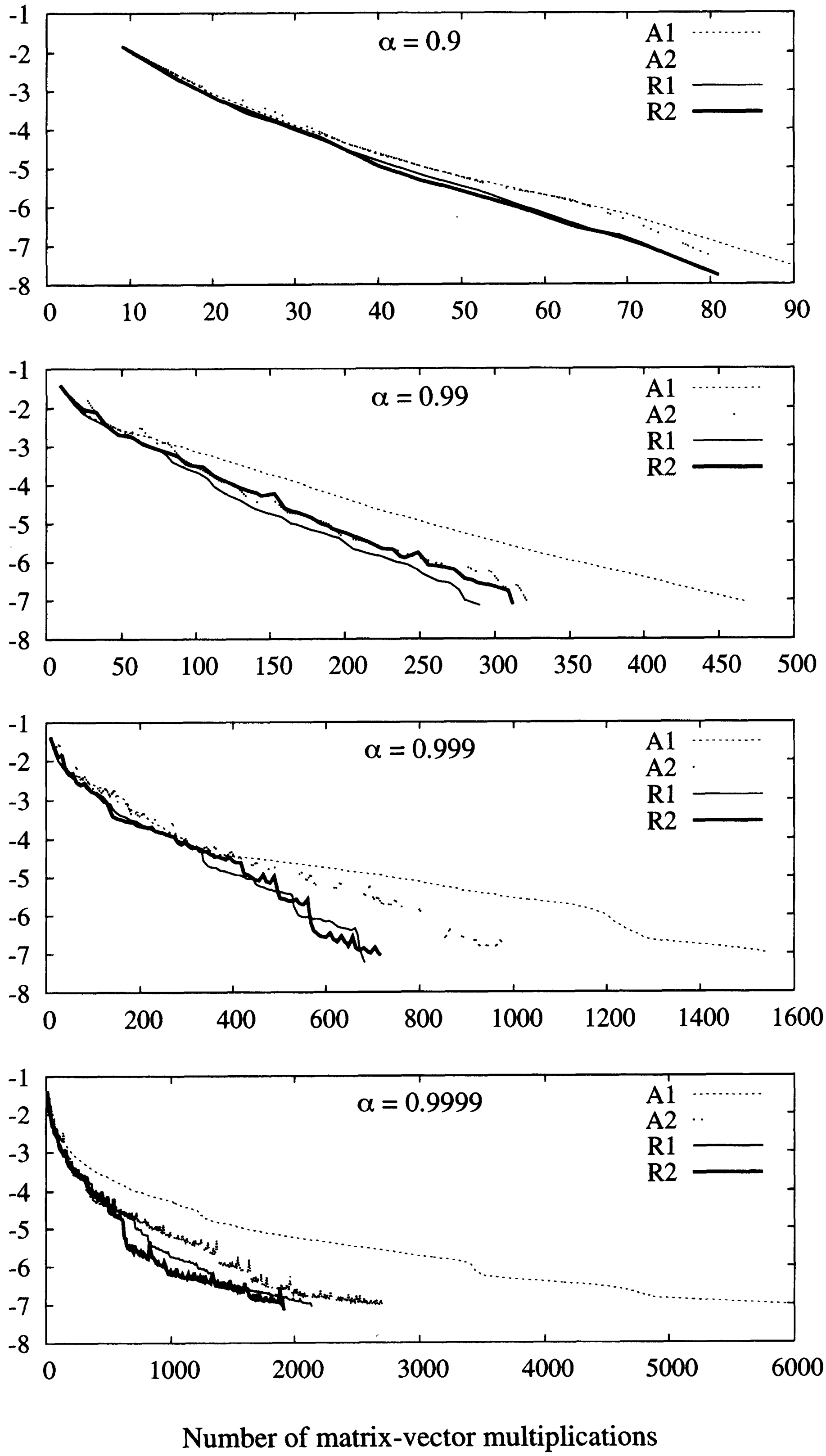

Figure 1: Convergence behaviors of A1 (Arnoldi-type), A2 (adaptively accelerated Arnoldi), R1 (Riccati-type), and R2 (variant of Riccati-type).

5 Conclusions

We have explored a variant of the Riccati method to compute PageRank efficiently. The relationship between the Riccati method and the Arnoldi method motivates us to incorporate the advantage of the Arnoldi-type method for computing PageRank into the Riccati method. The proposed Riccati-type method can compute PageRank without using complex Ritz values and additional matrix-vector multiplications. Through this approach, we have derived another Riccati-type method utilizing the adaptively accelerated Arnoldi method. Numerical experiments illustrate the better performances of the Riccati-type methods, in particular, when α is close to 1.

References

[1] W. E. ARNOLDI, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.

[2] Z. BAI, J. DEMMEL, J. DONGARRA, A. RUHE, AND H. VAN DER VORST, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, 2000.

[3] J. H. BRANDTS, *The Riccati algorithm for eigenvalues and invariant subspaces of matrices with inexpensive action*, Linear Algebra Appl., 358 (2003), pp. 335–365.

[4] T. A. DAVIS AND Y. HU, *The university of Florida sparse matrix collection*, ACM Trans. Math. Software, 38 (2011), pp. 1–25. (`http://www.cise.ufl.edu/research/sparse/matrices/`).

[5] L. ELDÉN, *Matrix Methods in Data Mining and Pattern Recognition*, SIAM, Philadelphia, 2007.

[6] G. H. GOLUB AND C. GREIF, *An Arnoldi-type algorithm for computing page rank*, BIT, 46 (2006), pp. 759–771.

[7] Z. JIA, *Refined iterative algorithms based on Arnoldi's process for large unsymmetric eigenproblems*, Linear Algebra Appl., 259 (1997), pp. 1–23.

[8] A. N. LANGVILLE AND C. D. MEYER, *Deeper inside PageRank*, Internet Math., 1 (2003), pp. 335–380.

[9] T. MIYATA, *Riccati method for PageRank*, in NCTS Workshop on Numerical Linear Algebra and High Performance Computing, 2013.

[10] C. MOLER, *Experiments with MATLAB*, Electronic edition published by MathWorks, 2011.

[11] L. PAGE, S. BRIN, R. MOTWANI, AND T. WINOGRAD, *The PageRank citation ranking: bringing order to the Web*, Technical Report, Stanford InfoLab. 1999-66, 1999.

[12] J.-F. YIN, G.-J. YIN, AND M. NG, *On adaptively accelerated Arnoldi method for computing PageRank*, Numer. Linear Algebra Appl., 19 (2012), pp. 73–85.