Lanczos type method for computing PageRank

by

Kazuma Teramoto
Takashi Nodera

Department of Mathematics
Faculty of Science and Technology
Keio University

©2015 KSTS
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 223-8522 Japan
Lanczos type method for computing PageRank

Kazuma Teramoto\textsuperscript{1} \hspace{1cm} Takashi Nodera\textsuperscript{2}

March 1, 2015

Abstract

Computing eigenvalues and eigenvectors of a large matrix is one of the most important tasks in numerical analysis. PageRank is a link analysis algorithm used by the internet search engine, Google, which ranks each document in the order of its relative importance in its database. In this paper, a new algorithm for computing the PageRank vector is proposed, using a combination of the Lanczos bi-orthogonalization algorithm with a semi-orthogonality and a SVD (singular-value decomposition). This method converges faster than the Arnoldi method requiring less computation time. The results of some numerical experiments have been documented to evaluate the effectiveness of our proposed Lanczos algorithm.

\textbf{key words.} PageRank, Lanczos method, semi-orthogonality, eigenvalue and eigenvector

\textbf{AMS(MOS) subject classifications.} 65F10, 65M12

Contents

1 Introduction \hspace{1cm} 1

2 Arnoldi Method \hspace{1cm} 3

3 Lanczos Method with Semi-Orthogonality 3

\hspace{1cm} 3.1 Computing PageRank with Semi-Orthogonality \hspace{1cm} 5

4 Numerical Experiments \hspace{1cm} 6

\hspace{1cm} 4.1 Example 1 \hspace{1cm} 7

\hspace{1cm} 4.2 Example 2 \hspace{1cm} 7

5 Conclusions \hspace{1cm} 7

References \hspace{1cm} 8

1 Introduction

PageRank is the essential approach for ranking a Web page where the status of a page is determined according to its link structure on the Web. This model has been used by Google as a part of its contemporary search engine equipment. Nowadays, the precise ranking procedures and computation schemes used by Google are no longer public, but the
1 Introduction

Algorithm 1: Arnoldi Method

1 Compute \( q_1 = q_0 / \|q_0\|_2; \)
2 for \( j = 1, 2, ..., \) to \( m \) do
3 \hspace{1cm} Compute \( w_j = Aq_j; \)
4 \hspace{1cm} for \( k = 1, 2, ..., j \) do
5 \hspace{2cm} \( h_{kj} = q_k^T w_j; \)
6 \hspace{2cm} \( w_j = w_j - h_{kj}q_k; \)
7 \hspace{1cm} end
8 \hspace{1cm} Compute \( h_{j+1,j} = \|w_j\|_2; \)
9 \hspace{1cm} if \( h_{j+1,j} = 0 \) then
10 \hspace{2cm} stop and exit;
11 \hspace{1cm} else
12 \hspace{2cm} Set \( q_{j+1} = w_j / h_{j+1,j}; \)
13 \hspace{1cm} end
14 end

PageRank model has taken on a life of its own and has received important consideration in the science and technology communities in the past ten years. PageRank is essentially a fixed distribution vector of the Markov chain wherein the transition matrix is a convex combination of the Web link graph and a precise rank-1 matrix. A major parameter in the model is a ‘damping factor’, a scalar that determines the weight given to the Web link graph in the model. The weighted PageRank constitutes the elements of the dominant eigenvector of the modified adjacency matrix as follows:

\[
A = \alpha P + (1 - \alpha) E,
\]

where \( P \) is a column stochastic matrix, \( \alpha \) is a ‘damping factor’, and \( E \) is a rank one matrix. The specified derivation is detailed in a paper by Kanvar et al. [8].

More recently, the computation of the eigenpair (eigenvalue and eigenvector) of non-symmetric matrices have become one of the most important tasks in many science and technology applications. A typical example, nowadays, is the computation of PageRank based on the link structure of the Web. Due to the great size and sparsity of the matrix, factorization schemes are considered impractical, and iterative schemes are used, where the computation is dominated by matrix-vector products. Detailed descriptions of this problem are available, and the algorithms can be found in numerous references, e.g. [2, 7, 6, 8, 9, 10, 13].

The power method was firstly considered for computing PageRank. For detailed properties of PageRank using the power method, please refer to Kanvar et al. [8]. However, the power method has its disadvantages, i.e. for some given matrices, the power method converges very slowly. Although different methods have been suggested for accelerating convergence, there have been no significant improvements so far. For example, a procedure using orthogonalization such as the Arnoldi method has been suggested [7]. Teratomo et al. [13] have published studies which suggest the Lanczos method is an effective procedure to compute the PageRank vector. In this paper, we have studied a new algorithm for computing the PageRank vector, using a combined Lanczos bi-orthogonalization algorithm and SVD (Singular Value Decomposition).
Algorithm 2: Arnoldi-Type Method

1. Choose $q_0$ with $\|q_0\|_2 = 1$;
2. for $l = 1, 2, \ldots$, until convergence do
   3. Compute $[Q_m, H_{m+1,m}] = \text{Arnoldi}(A, q_0, m)$;
   4. Compute singular value decomposition $H_{m+1,m} - [I; 0] = U \Sigma V^T$;
   5. Compute $q_0 = Q_m v_m$;
   6. Compute $r = \sigma_m Q_m + u_m$;
   7. if $\|r\|_1 < \text{TOL}$ then
      8. stop and exit;
   9. end
10 end

The remainder of the paper is organized as follows: Section 2 includes a brief description of the Arnoldi method of the PageRank vector. In section 3, a new Lanczos algorithm with a SVD scheme is proposed. In section 4, the results of numerical experiments obtained through running MATLAB codes are shown. The conclusion follows.

2 Arnoldi Method

In this section, a brief introduction of the Arnoldi method [1] for computing the PageRank vector is given. The Arnoldi method, which uses Algorithm 1, builds an orthonormal basis for the Krylov subspace given by:

$$\mathcal{K}_m(A, q_0) = \text{span}\{q_0, Aq_0, \ldots, A^{m-1}q_0\},$$

where the Krylov subspace is restricted to a fixed dimension $m$ and $q_0$ is an initial vector which satisfies $\|q_0\| = 1$. From Algorithm 1, the following relations hold:

$$AQ_m = Q_m H_m + h_{m+1,m} q_{m+1} e_m^T,$$

$$Q_m^T A Q_m = H_m,$$

where $Q_m = [q_1, q_2, \ldots, q_m] \in \mathbb{R}^{n \times m}$ is a column-orthogonal matrix, and $H_m = \{h_{i,j}\} \in \mathbb{R}_{m \times m}$ is a Hessenberg matrix [7]. Since $H_m$ is an orthogonal projection from $A$ to $\mathcal{K}_m$, the eigenvalue of $H_m$ can be used as an approximate eigenvalue of $A$. If $y$ is the eigenvector of $H_m$, then $Q_m y$ is the approximate eigenvector of $A$. This is true because it has been established that the largest eigenvalue of a PageRank matrix is 1. The Arnoldi-type method was proposed by Golub and Greif [5] and it will be referred to as Algorithm 2. In Algorithm 2, a singular value decomposition is computed normally, instead of the eigenvalue of $H_m$ [7]. When $m$ increases, the total computation cost of this method continuously increases with every cycle, while the total iterations decrease. It can be very difficult to choose $m$ a priori and if too small a value is chosen, the convergence may stall. Consequently, it is difficult to choose the optimal value of $m$ to minimize total computation cost (CPU time).

3 Lanczos Method with Semi-Orthogonality

In this section, a modified Lanczos method with semi-orthogonality is proposed. This algorithm is derived from using the result of Day [3]. This can be used when a non-
3 Lanczos Method with Semi-Orthogonality

Hermitian matrix is converted to a tridiagonal matrix and one of the Krylov subspace methods. Given two starting vectors, this algorithm computes simultaneously, a basis for the following two subspaces, where, matrix $A$ is a $n \times n$ PageRank matrix, $p^*$ and $q$ are initial vectors and the conjugate transpose is denoted by the superscript $^{\ast \ast}$. Unlike the classical Lanczos method [13], this algorithm has an advantage because it can perform computations without having to set the value of $m$:

$$
K_m(A, q) = \text{span}\{ q, Aq, ..., A^{m-1}q \},
$$

$$
K_m(A, p^*) = \text{span}\{ p^*, p^*A, ..., p^*A^{m-1} \}.
$$

Set $p = p_1^*$, $q = q_1$. After $j$ successful Lanczos steps, the matrices $P$ and $Q$ are produced. The rows of $P_j^*$ span $K_j^m(A, p^*)$ and $n$ the column of $Q_j$ span $K_j^m(A, q)$. The matrix $T_j = P_j^*AQ_j$ is tridiagonal. This means that $\Omega_j = P_j^*AQ_j$ is diagonal. Let $P_j^*$ and $Q_j$ have full rank. If $P_j^*Q_j$ is invertible then,

$$
\Pi_j = Q_j\Omega_j^{-1}P_j^* \quad (3)
$$

and is an oblique projector ($\Pi_j^2 = \Pi_j$) onto $\text{Range}(Q_j)$. Generally, it is not orthogonal ($\Pi_j \neq \Pi_j$), and

$$
\{ u^*\Pi_j, u \in C^j \} = \text{Range}(P_j^*) \quad (4)
$$

is also an oblique projector onto the dual space, where $C^j$ is a $j$ dimensional complex space. From equation (3) and (4), $\Pi_jAP_j$ is a projection from $A$ onto the dual $\text{Range}(Q_j)$ and $\text{Range}(P_j)^*$. Assuming that $P_j^*$ and $Q_j$ exist, the description of $A$ with regard to the basis $\{ q_1, ..., q_n \}$ is $Q_n^{-1}AQ_n$. $\Pi_n = I$ implies that $Q_n^{-1} = \Omega^{-1}P_n^*$ and $Q_n^{-1}AQ_n = \Omega^{-1}T_n$. The tridiagonal matrix $\Omega_j^{-1}T_j$ represents $\Pi_jAP_j$ in the bases $(q_1, ..., q_j)$ and $(\omega_1^{-1}p_1^*, ..., \omega_j^{-1}p_j^*)$. The next step is to consider how to generate Lanczos vectors. Lanczos vectors can compute the following equations:

$$
\beta_2p_2^* = p_1^*A - \frac{\alpha_1}{\omega_1}p_1^*, \quad (5)
$$

$$
\beta_{m+1}p_{m+1}^* = p_m^*A - \frac{\alpha_m}{\omega_m}p_m^* - \frac{\gamma_m\omega_m}{\omega_{m-1}}p_{m-1}^*, \quad (6)
$$

$$
\gamma_2q_2 = Aq_1 - \frac{\alpha_1}{\omega_1}q_1, \quad (7)
$$

$$
\gamma_{m+1}q_{m+1} = Aq_m - \frac{\alpha_m}{\omega_m}q_m - \frac{\beta_m\omega_m}{\omega_{m-1}}q_{m-1}. \quad (8)
$$

The initial condition is $\omega_m = p_m^*q_m, \alpha_m = p_m^*Aq_m$. With the intention of making the size of the Lanczos vector $1$, $\beta$ and $\gamma$ are scaled as follows:

$$
\beta_{m+1} = \left\| p_m^*A - \frac{\alpha_m}{\omega_m}p_m^* - \frac{\gamma_m\omega_m}{\omega_{m-1}}p_{m-1}^* \right\|_2, \quad (9)
$$

$$
\gamma_{m+1} = \left\| Aq_m - \frac{\alpha_m}{\omega_m}q_m - \frac{\beta_m\omega_m}{\omega_{m-1}}q_{m-1} \right\|_2. \quad (10)
$$

When representing the recurrence formula in matrix form, the following equation holds:

$$
P_m^*A = T_m\Omega^{-1}P_m + e_m\beta_{m+1}p_{m+1}^*, \quad (11)$$
Algorithm 3: Lanczos Method with Local Orthogonality

1. Start: $p_1 = p / \|p\|_2, q_1 = q / \|q\|_2, \omega_0 = 1, \beta_1 = \gamma_1 = 0, \omega_1 = p^*_1 q_1$;
2. for $l = 1, 2, ..., \text{MaxStep}$ do
3. \hspace{1em} $r_i^* = p_i^* B - \frac{\omega_i}{\omega_{i-1}} p_{i-1}^*$,
4. \hspace{1em} $s_i = A q_i - q_{i-1} \frac{\beta_i}{\omega_{i-1}}$;
5. \hspace{1em} $\alpha_i = r_i^* s_i$;
6. \hspace{1em} $r_i^* := r_i^* - \frac{\alpha_i}{s_i} p_i^*$, $s_i := s_i - q_i \frac{\alpha_i}{s_i}$;
7. \hspace{1em} $\alpha_i^* = r_i^* q_i$, $\alpha_i^* = p_i^* s_i$;
8. \hspace{1em} $r_i^* := r_i^* - \frac{\alpha_i^*}{s_i} p_i^*$, $s_i := s_i - q_i \frac{\alpha_i^*}{s_i}$;
9. \hspace{1em} $\beta_{i+1} = \|r_i^*\|_2$, $\gamma_{i+1} = \|s_i\|_2$;
10. \hspace{1em} $p_{i+1}^* = r_i^*/\beta_{i+1}$, $q_{i+1} = s_i/\gamma_{i+1}$;
11. \hspace{1em} $\omega_{i+1} = p_{i+1}^* q_{i+1}$;
12. check for breakdown: $|\omega_{m+1}| < (n + 10(m + 1)) \epsilon$;
13. check for convergence;
14. end

$$AQ_m = Q_m \Omega^{-1} T_m + q_{m+1} \gamma_{m+1} e_m^*,$$ \hspace{1em} (12)

The tridiagonal matrix can compute as follows:

$$T_m = \begin{pmatrix}
\alpha_1 & \beta_2 \omega_2 & & & \\
\gamma_2 \omega_2 & \alpha_2 & \beta_3 \omega_3 & & \\
& \ddots & \ddots & \ddots & \\
& & \gamma_{m-1} \omega_{m-1} & \alpha_{m-1} & \beta_m \omega_m \\
& & & \gamma_m \omega_m & \alpha_m
\end{pmatrix}.$$

Using these equations, we can set the Lanczos algorithm as Algorithm 3. The local orthogonality is defined as follows:

Definition 3.1 (Day [3]) Lanczos vectors $p_1^*, \ldots, p_i^*$ and $q_1, \ldots, q_i$ satisfy the local orthogonality of the following equations in $i$ steps:

$$|\cos \angle p_i^* q_{i-1}| \leq 4 \epsilon, \hspace{1em} (13)$$

$$|\cos \angle p_i^* q_i| \leq 4 \epsilon. \hspace{1em} (14)$$

In order to satisfy equations (13) and (14), it is necessary to maintain the bi-orthogonality between $p_{i+1}$ and $q_i$, and between $q_{i+1}$ and $p_i$, in line 6 and 7 of Algorithm 6. When the closely bi-orthogonal vectors are improved, then the cosine of the angles between the new vectors is detached from the dimension. These bi-orthogonalities are generally referred to as a local orthogonality.

The right Ritz vector which can compute this algorithm is the PageRank vector. The convergence criterion will be discussed at length in the next section.

3.1 Computing PageRank with Semi-Orthogonality

In this subsection, a modified Lanczos method for computing PageRank with a semi-orthogonality will be introduced. The eigenvalues of $A$ are approximated using the eigenvalues of the pair $(T_m, \Omega_m)$. The pair $(T_m, \Omega_m)$ will be used to compute PageRank. The
Algorithm 4: Lanczos Method for Computing PageRank

1. Choose initial vectors $p, q$ such that $(p, q) = 1$;
2. $[T, P, Q] = \text{Lanczos}(A, p, q)$;
3. Compute singular value decomposition $T_m - I = U\Sigma V^T$;
4. if Convergence condition satisfies then
5. Compute PageRank vector $Q_m v_T$;
6. end
7. continue to Lanczos method;

The eigentriplet of matrix $A$ is approximately as follows:

$$ u^*_T T_m = \theta_T u^*_T \Omega_m, $$

$$ T_m v_T = \theta_T \Omega_m v_T. $$

where, $\theta_T$ is a Ritz value, $u^*_T P_m^*$ is a left Ritz vector, and $Q_m v_T$ is a right Ritz vector. In this paper, when a Lanczos algorithm converges, the right Ritz vector is regarded as the PageRank vector. The following equation has been used as the convergence criterion:

$$ \frac{||Ax - \lambda x||_2}{||A||_1} < \epsilon. $$

(15)

This definition of the Ritz value and Ritz vector, $Ax - \lambda x$ can be represented as follows:

$$ Ax - \lambda x \approx AQ_m v_T - \theta_T Q_m v_T. $$

(16)

Substitute equation (16) for equation (12), and the following equation is satisfied:

$$ Ax - \lambda x \approx q_{m+1} e_{m+1} v_T. $$

(17)

So even if only the value of $v_T$ is known, it is possible to compute the convergence criterion easily. From the above, the modified Lanczos method with semi-orthogonality with SVD is Algorithm 4.

4 Numerical Experiments

In this section, the numerical results of the two methods described in the previous sections on the test problems were compared. All computing of the numerical experiments were run on a PC with 3.6 GHz and an eight-gigabyte memory using MATLAB R2012b. These results are shown to demonstrate the efficiency of the Lanczos algorithm with a semi-orthogonality. The test matrices, Death_Penalty and E-mail Enron, were obtained from Web pages: Stanford University Large Network Data Set Collection and University of Toronto Data sets for Link Analysis Ranking Experiments.
Table 1: Example 1, Iterations and Computation Time (4298 × 4298)

<table>
<thead>
<tr>
<th>c</th>
<th>Power Time</th>
<th>Arnoldi IT</th>
<th>Arnoldi Time</th>
<th>Lanczos IT</th>
<th>Lanczos Time</th>
<th>LanLD IT</th>
<th>LanLD Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.85</td>
<td>55</td>
<td>1.06</td>
<td>8</td>
<td>0.453</td>
<td>10</td>
<td>0.677</td>
<td>5</td>
</tr>
<tr>
<td>0.90</td>
<td>76</td>
<td>1.48</td>
<td>9</td>
<td>0.516</td>
<td>10</td>
<td>0.677</td>
<td>6</td>
</tr>
<tr>
<td>0.95</td>
<td>119</td>
<td>2.28</td>
<td>11</td>
<td>0.625</td>
<td>10</td>
<td>0.677</td>
<td>5</td>
</tr>
<tr>
<td>0.99</td>
<td>226</td>
<td>4.33</td>
<td>13</td>
<td>0.765</td>
<td>10</td>
<td>0.677</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 2: Example 2, Iterations and Computation Time (15000 × 15000)

<table>
<thead>
<tr>
<th>c</th>
<th>Arnoldi (m=10) IT</th>
<th>Arnoldi (m=10) Time</th>
<th>Lanczos IT</th>
<th>Lanczos Time</th>
<th>LanLD IT</th>
<th>LanLD Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.85</td>
<td>4</td>
<td>4.16</td>
<td>10</td>
<td>5.23</td>
<td>6</td>
<td>2.78</td>
</tr>
<tr>
<td>0.90</td>
<td>6</td>
<td>5.02</td>
<td>10</td>
<td>5.23</td>
<td>7</td>
<td>3.07</td>
</tr>
<tr>
<td>0.95</td>
<td>9</td>
<td>7.32</td>
<td>10</td>
<td>5.23</td>
<td>7</td>
<td>3.07</td>
</tr>
<tr>
<td>0.99</td>
<td>17</td>
<td>14.09</td>
<td>10</td>
<td>5.23</td>
<td>7</td>
<td>3.07</td>
</tr>
</tbody>
</table>

4.1 Example 1

The Power method, Arnoldi method, Lanczos method and the proposed method were applied to the 4298 × 4298 matrix which was downloaded from the following URL: http://www.cs.toronto.edu/~tsap/experiments/download/download.html. This matrix was made by converting the directed graph to the adjacency matrix. Eigenvalues of this matrix are dense, and the computation time of the power method is time consuming. There was no significant difference between the Lanczos method and the Arnoldi method, but when the value of c was large, the proposed Lanczos method converged faster than other methods.

4.2 Example 2

The Power method, the Arnoldi method, the Lanczos method and the proposed method were applied to the 15000 × 15000 E-mail Enron matrix which was downloaded from the following URL: http://snap.stanford.edu/data/index.html. This matrix was made by converting the directed graph to the adjacency matrix. This directed graph refers to e-mail server links. The numerical results are tabulated in Table 3. With the exception of when the Matrix size is larger than the previous one and when the value of c is large, convergence requires significantly more time. Otherwise, the proposed method is not significantly influenced by the value of c, and its convergence speed is faster.

5 Conclusions

In this paper, we proposed a new algorithm to compute PageRank, using a combination of the Lanczos method and SVD. Computation times were dependent on the number of the tridiagonal matrix’s degree. Our numerical results showed that computation times were
constant. The proposed method has advantages which are not critically dependent on $\alpha$. According to our numerical experiments, the power method requires more time than other methods, so it is not a practical choice for computing PageRank. On the other hand, we must note that the Arnoldi method with SVD satisfies the stopping criterion faster than the power method. In terms of speed, however, our proposed Lanczos method was faster than the Arnoldi method.

References


References

http://dx.doi.org/10.1016/j.amc.2005.08.035

http://dx.doi.org/10.5176/2251-1911_CMCGS14.15

Author addresses

1. Kazuma Teramoto, School of Fundamental Science and Technology, Graduate School of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohoku, Yokohama, Kanagawa, 223-8522, JAPAN.
mailto:trmt@keio.jp

2. Takashi Nodera, Department of mathematics, Keio University, 3-14-1 Hiyoshi, Kohoku, Yokohama, Kanagawa, 223-8522, JAPAN.
mailto:nodara@math.keio.ac.jp
Department of Mathematics
Faculty of Science and Technology
Keio University

Research Report

2014

[14/001] A. Larraín-Hubach, Y. Maeda, S. Rosenberg, F. Torres-Ardila,
*Equivariant, strong and leading order characteristic classes associated to fibrations*,
KSTS/RR-14/001, January 6, 2014

[14/002] Ryoichi Suzuki,
*A Clark-Ocone type formula under change of measure for canonical Lévy processes*,
KSTS/RR-14/002, March 12, 2014

2015

[15/001] Shiro Ishikawa,
*Linguistic interpretation of quantum mechanics: Quantum Language*,
KSTS/RR-15/001, January 22, 2015

[15/002] Takuji Arai, Ryoichi Suzuki
*Local risk-minimization for Lévy markets*,
KSTS/RR-15/002, February 2, 2015

[15/003] Kazuma Teramoto, Takashi Nodera
*Lanczos type method for computing PageRank*,
KSTS/RR-15/003, March 9, 2015

[15/004] Yoichi Matsuo, Takashi Nodera
*Block symplectic Gram-Schmidt method*,
KSTS/RR-15/004, March 9, 2015

[15/005] Yuto Yokota, Takashi Nodera
*The L-BFGS method for nonlinear GMRES acceleration*,
KSTS/RR-15/005, March 9, 2015