HermitianStatus: Toward the Calculation cost efficient Alternative to Google PageRank

Keita Sugihara

Department of Mechatronics, Faculty of Science and Engineering, Nanzan University

Abstract

This study is focused on a proposed alternative algorithm to Google PageRank, named HermitianStatus, which employs Hermitian adjacency matrix to score a node of the network to overcome the high cost calculation issues of the Google’s algorithm. The problem with Google is associated with the damping factor of the algorithm. The algorithm for HermitianStatus is designed to be free from the damping factor, and it reproduces PageRank results well. Moreover, the proposed algorithm can mathematically and systematically change the point of the node of the network.

Key words: Google PageRank, damping factor, link analysis, Hermitian adjacency matrix

I. Introduction

Google PageRank is a link analysis algorithm developed for search engines. In general, it is known that the link analysis approach is associated with the high cost of calculations. In this paper, a new algorithm is proposed as an alternative to PageRank to address this problem.

II. Related Studies

PageRank [1] [2] of the Google search engine has been a widely investigated algorithm [3]. The algorithm’s high cost of calculations is caused by the size of the Google matrix, is approximately 8.1 billion [4].

There have been two approaches to overcoming the problem of PageRank and its damping factor. In the first approach, Avrachenkov and Litvak have shown that the global PageRank may be calculated from the local PageRanks of weakly connected components [5]. However, Andrei, Kumar, and Maghoul have suggested that 91% of the web sites on the Internet are in a weakly connected network [6]. In the second approach, Brinkmeier has focused on strong components [7]. PageRank and both of the abovementioned approaches use the Adjacency matrix to describe the net.

This paper proposes HermitianStatus, HStatus, or HS, which is a new algorithm for cost efficient node ranking calculations. HermitianStatus uses the Hermitian adjacency matrix to describe the Internet. Hermitian matrix is an idea in graph theory, which has been explored by Guo [8]. Sugihara [9] [10] was the first to use the Hermitian adjacency matrix to score a node of a directed graph.

III. PageRank

a) Definitions

Definition 1: A semi path is a collection of distinct nodes, \( v_1, v_2, \ldots, v_n \) together with \( n - 1 \) links, one from each \( v_1v_2 \) or \( v_2v_1 \), \( v_2v_3 \) or \( v_3v_2 \), \( \ldots, v_{n-1}v_n \) or \( v_nv_{n-1} \).

Definition 2: A path is a collection of distinct nodes, \( v_1, v_2, \ldots, v_n \), together with the links, \( v_1v_2, v_2v_3, \ldots, v_{n-1}v_n \).

Definition 3: A directed graph \( G = (V, E) \) is called weakly connected if, for all nodes \( v_1, v_2 \in V \) there exists a semi path between \( v_1 \) and \( v_2 \).

Definition 4: A directed graph \( G = (V, E) \) is called unilaterally connected if, for all nodes \( v_1, v_2 \in V \) there exists a path from \( v_1 \) to \( v_2 \).
there exists a path from \( v_1 \) to \( v_2 \) or from \( v_2 \) to \( v_1 \).

**Definition 5:** A directed graph \( G = (V, E) \) is called strongly connected if for all nodes \( v_1, v_2 \in V \) there exists a path from \( v_1 \) to \( v_2 \).

\[ \] 

\( b) \) PageRank algorithm

PageRank has three characteristics. First, the page receives a high score when it has an in-link from a node with a high score. Second, the page catches a high score when it has many in-links. Third, a page receives a high score when it has an in-link from a node with few out-links. Thus, the selected out-links to the page is important to obtain a high score. Page Rank considers the score of a node in a directed graph on the basis of the nodes that have an out-link to the node without taking into account a node that has an in-link from the node. PageRank represents the network \( G = (V, E) \) as the matrix (1).

\[
A_{uv} = \begin{cases} 
1 & \text{if } uv \in E; \\
0 & \text{otherwise.} 
\end{cases} 
\tag{1}
\]

PageRank employs the damping factor \( d(0 \sim 1) \) to transform the matrix to the Google matrix \( G \) to apply the power method to solve the eigenvalue problem because the method is applicable only when the multiplicity of the absolute maximal eigenvalue of the matrix equals 1 [4]. For the network, the condition is strongly connected: for each pair of nodes there is a sequence of directed links leading from one node to another. For example, the network in Fig. 1 is not strongly connected, but its \( G \) is assured to be strongly connected.

![Network Comprising Four Components](image)

**c) Problems**

PageRank currently faces the following problems.

\( 1) \) High Cost of Calculations

The algorithm is known for its high cost of calculations. The size of the matrix is the number of nodes in the network. Approximately 8.1 billion web-sites exist on the internet [4]. When we use the adjacency matrix to describe the internet, we may employ \( A_1, A_2, \ldots, A_N \), which correspond to
weakly connected components in the net. Thus, the size of $A_t$ may not be large. However, the size of the Google matrix $G$ must be approximately 8.1 billion because of the use of the damping factor $G$ to make the network strongly connected, which results in the high cost of PageRank calculation.

2) Empirical Labor

The selection of the damping factor value is eminently empirical, and in most cases, the value of 0.85 proposed by Brian and Page is used [11].

3) Inconsistent Rankings

The network has inconsistent rankings when using different damping factor values [12]. An example of this case for the network in Fig. 2 is shown in Fig. 3. As stated in the abovementioned empirical labor problem, we do not know how the ranking of the nodes will be changed before we increase the damping factor from 0 to 1.

4) Fixed Top-Ranking Node

This problem means that the top-ranking node of the directed graph is fixed for all damping factor values from 0 to 1 even though we would like another node to be recognized as the top ranking [10].

5) Possible Use for Spam

A specific damping factor value can be used to create spam against a search engine [13].
IV. HermitianStatus

a) Definitions

Definition 5: A node $v_1$ is reachable to a node $v_2$ if there is a path from the former to the latter.

Definition 6: For a directed graph $G = (V, E)$, the Hermitian adjacency matrix $H$ is defined in the following equation (2), using $i$ as the imaginary unit [4]. This matrix is a Hermitian matrix because for all $u$ and $v$, $h_{uv}$ and $h_{vu}$ are complex conjugates each other.

$$H_{uv} = \begin{cases} 1 & \text{if } uv \in E \text{ and } vu \in E; \\ i & \text{if } uv \in E \text{ and } vu \notin E; \\ -i & \text{if } uv \notin E \text{ and } vu \in E; \\ 0 & \text{otherwise}. \end{cases} \quad (2)$$

b) Advantage of the Hermitian adjacency matrix

The proposed algorithm represents a network by the Hermitian adjacency matrix in graph theory [8]. It is applied to each at least weakly connected component of the network for the power method because trials show that the multiplicity of the maximal eigenvalue of the matrix is 1 if the network is weakly connected. HermitianStatus is based on eigenvector centrality[14] in social network analysis[15].

As shown in the table 1, an advantage of using the Hermitian adjacency matrix is that its eigenvalues of are always real numbers, because it is a Hermitian matrix. Moreover, the results of trials suggest that, if the directed graph is weakly connected, the absolute dominant eigenvalue, $|\lambda|_1$, of the graph’s Hermitian adjacency matrix, $H$, is a positive number with a multiplicity of 1, a negative number with a multiplicity of 1, or a positive number with a multiplicity of 1 and a negative number with a multiplicity of 1. According to the results of the trials, these conditions are satisfied when we derive the Hermitian matrix $H'$ from $H$ using the method described below and when we create the Hermitian matrix $H''$ from $H$ using the procedure introduced in this paper. We select the positive eigenvalue, if the dominant eigenvalues include one positive and one negative real values.

| Table 1: Advantages of the Hermitian Adjacency Matrix over the adjacency matrix |
|---------------------------------|---------------------------------|
| Eigenvalues                    | Condition for the Absolute Maximal Eigenvalue Multiplicity 1. |
| Adjacency Matrix               | Not Always Real Numbers         |
| Hermitian adjacency Matrix     | Always Real Numbers             |
|                                | **Strongly Connected:** mathematically proven |
|                                | **Weakly Connected:** proven with examples |
c) Correspondences between HermitianStatus and PageRank

HermitianStatus has three characteristic that are shown in table 2. These characteristics correspond to the three characteristic of Google PageRank, as indicated in the table 2. In this study, we evaluated the algorithms for prototype HermitianStatus I, II, III before we evaluated HermitianStatus.

Table 2: Correspondences between HermitianStatus and PageRank

<table>
<thead>
<tr>
<th>Hermitian Status</th>
<th>PageRank</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>I. High score is given to a page that has an in-link with a high score.</td>
</tr>
<tr>
<td></td>
<td>II. High score is given to a page that has many in-links.</td>
</tr>
<tr>
<td></td>
<td>III. High score is given to a page that has an in-link from a site with few out-links.</td>
</tr>
<tr>
<td></td>
<td>Corresponding to PageRank’s I by combination with C</td>
</tr>
<tr>
<td></td>
<td>Identical to PageRank’s II</td>
</tr>
<tr>
<td></td>
<td>Corresponding to PageRank’s I by combination with A</td>
</tr>
<tr>
<td></td>
<td>Identical to PageRank’s III</td>
</tr>
</tbody>
</table>
d) Algorithms for Prototype HermitianStatus I, II and III

The algorithms for the prototype HermitianStatus methods for the directed graph nodes are as follows. We use \( N \) to designate the number of all the nodes of the entire graph. Those algorithms are designed to be used for each weakly connected directed graph in the entire graph. Once we derive a score of the node of a weakly connected graph from the algorithms, we can compare it to the score of another node in a different weakly connected graph, which is also derived by those algorithms.

1) Prototype HermitianStatus I

This method expresses a network with the Hermitian adjacency matrix. The algorithm solves the eigenvalue problem of the matrix and plots the elements of the eigenvector on the complex plane.

Stage 1: 1. Decompose the entire network into the weakly connected components.
2. Create the Hermitian adjacency matrix \( H(3) \) of each component (Fig.3) for the eigenvalue problem for the absolute maximal eigenvalue \( \lambda_1 \).

Stage 2: Solve the eigenvalue problem(4) using the power method so that the element corresponding to the node with 0 in-link equals 1.

Stage 3: Locate each element on the complex plane (Fig. 4). Define the prototype Hermitian centrality score of the node \( i \) as \( (2\pi - \arg(x_i)) \times |x_i| \).

This algorithm captures the link relations between the nodes in Fig. 3 as the relative locations of their vectors on the complex plane. The score of node 1 is \( (2\pi - \arg(x_1)) \times |x_1| = 0 \) because its vector on the plan is set on the real axis in Stage 2. The score of node2 is \( (2\pi - \arg(x_2)) \times |x_2| = (1/4)\pi \times |x_2| \), as defined in Stage 3. The angle and length of the vector of node 2 are the composition of the following: \( \pi/2 \) clockwise rotation of the vector of node 1, \( \pi/2 \) counter clockwise rotation of the vector of node 3, \( \pi/2 \) counter clockwise rotation of the vector of node 4, as indicated in (4). For node 3, its score is \( (2\pi - \arg(x_3)) \times |x_3| \), as set in Stage 3. For the vector of node 3, its angle and length are the composition of the following: \( \pi/2 \) clockwise rotation of the vector of node 2, \( \pi/2 \) counter clockwise rotation of the vector of node 4, as defined in (4).

The score of node 4 is \( (2\pi - \arg(x_4)) \times |x_4| \), as define in Stage 3. The angle and the length of the vector of node 4 is the composition of the following: \( \pi/2 \) clockwise rotation of the vector of node 2, \( \pi/2 \) clockwise rotation of the vector of node 3, as set in (4). The ranking is 4, 3, 2, 1, from the highest to the lowest.

This algorithm is not applicable to the network with a long path because in this case, there would be a vector the angle of which exceeds \( 2\pi \). Prototype HermitianStatus II overcomes this problem.

\[
\begin{align*}
    X &= \frac{1}{H}X \\
    C_1 &= \frac{1}{|H|} (0C_1 + iC_2 + 0C_3 + 0C_4) = 1 + 0i \\
    C_2 &= \frac{1}{|H|} (-iC_1 + 0C_2 + 0C_3 + iC_4) \\
    C_3 &= \frac{1}{|H|} (0C_1 - iC_2 + 0C_3 + iC_4) \\
    C_4 &= \frac{1}{|H|} (0C_1 - iC_2 - iC_3 + 0C_4) \\
\end{align*}
\]

Fig. 3: Weakly(Unilaterally) Connected Network

\[
H = \begin{bmatrix}
0 & i & 0 & 0 \\
-i & 0 & i & i \\
0 & -i & 0 & i \\
0 & -i & -i & 0
\end{bmatrix}
\]

(3)

Fig 4: Complex Plane Plotting of 2-dimensional vectors of the nodes shown in Fig. 3 focused on the fourth quadrant
2) **Algorithm for Prototype HermitianStatus II**

**Stage 1:**
1. Decompose the entire network into the weakly connected components.
2. Create the Hermitian adjacency matrix $H$ of each component (Fig. 3).
3. For (a) in $H$, convert each $i$ element to $s(t + i)$ and each $-i$ element to $s(t - i)$ (5), which derives $H$ (6): $N$ stands for the number of nodes of the entire network.

**Stage 2:**
Solve the eigenvalue problem using the power method so that the element corresponding to the node with 0 in-links equals 1 (7).

**Stage 3:** Locate each element of $X$ on a complex plane (Fig. 5). Define the Hermitian centrality score value of the node as $\{2\pi - arg(x_j)\} \times |x_j|$.

In this algorithm, the third step of Stage 1 solves the problem of the previous algorithm. The step confines all vectors of the nodes into the fourth quadrant by keeping the relative positions between the vectors realized in the previous algorithm. The ranking is identical to the result of Prototype HermitianStatus I.

This algorithm has the following problems. First, it can be applied to the unilaterally connected network and cannot handle only a weakly connected network such as the graph in Fig. 6. Second, the third characteristic of PageRank is not realized with the algorithm. Third, the algorithm is different from PageRank in the point that prototype HermitianStatus II determines the score of the node in a network by considering both of in-links and out-links, while the Google’s algorithm uses only in-links. Prototype HermitianStatus III overcomes those problems.

$$s(t \pm i) = \sin \left( \frac{\pi}{2} \times \frac{1}{N} \right) \left\{ \frac{1}{\sin \left( \frac{\pi}{2} \times \frac{1}{N} \right)} \cos \left( \frac{\pi}{2} \times \frac{1}{N} \pm 1 \right) \right\}$$

$$= \sin \left( \frac{\pi}{2} \times \frac{1}{N} \right) \left\{ \frac{1}{\sin \left( \frac{\pi}{2} \times \frac{1}{N} \right)} \cos \left( \frac{\pi}{2} \times \frac{1}{N} \pm \sin \left( \frac{\pi}{2} \times \frac{1}{N} \right) \right\} \quad (5)$$

$$H' = \begin{bmatrix}
0 & s(t + i) & 0 & 0 \\
0 & s(t - i) & s(t + i) & s(t + i) \\
0 & s(t - i) & 0 & s(t + i) \\
0 & s(t - i) & s(t - i) & 0
\end{bmatrix} \quad (6)$$

$$X = \frac{1}{|A|} HX$$

$$C_1 = \frac{1}{|A|} \left\{ 0 \times C_1 + s(t + i) \times C_1 + 0 \times C_3 + 0 \times C_4 \right\} = 1 + 0i$$

$$C_2 = \frac{1}{|A|} \left\{ s(t - i) \times C_1 + 0 \times C_1 + s(t + i) \times C_3 + s(t + i) \times C_4 \right\}$$

$$C_3 = \frac{1}{|A|} \left\{ 0 \times C_1 + s(t - i) \times C_1 + 0 \times C_3 + s(t + i) \times C_4 \right\}$$

$$C_4 = \frac{1}{|A|} \left\{ 0 \times C_1 + s(t - i) \times C_1 + s(t - i) \times C_3 + 0 \times C_4 \right\}$$

$$\text{Fig. 5: Complex plane plotting of 2-dimensional vectors of the nodes shown in Fig. 3 with a focus on the fourth quadrant}$$
3) **Prototype HermitianStatus III**

i) **Algorithm**

**Stage 1:** For each weakly connected graph (Fig. 6), label the nodes with zero in-links as $o_1, o_2, \ldots, o_p, \ldots, o_q$ (nodes 1 and 8).

**Stage 2:** If the weakly connected graph does not have a node with zero in-links, add a dummy node to the entire graph and add links from the dummy node to all the nodes in the weakly connected graphs. The same dummy node is used for another weakly connected graph if this weakly connected graph also does not have a node with zero in-links.

**Stage 3:** For each weakly connected graph, induce subgraphs using the nodes that are reachable from $o_1$ (Fig. 7) and create the Hermitian adjacency matrix $H$ from the subgraph of $o_1$. The same processes are conducted for the remaining $o_2, \ldots, o_i, \ldots, o_q$ (Fig. 8).

**Stage 4:** In $H$, convert each $i$ element to $s(t + i)$ and each $-i$ element to $s(t - i)$, which derives $H^*: N$ stands for the number of nodes of the whole network.

**Stage 5:** In $H^*$ each $s(t+i)$ is divided by the number of appearances of $s(t+i)$ in a row. Each diagonally corresponding $s(t-i)$ is divided by the same number, which creates $H''$.

**Stage 6:** Solve the eigenequation $X = \frac{1}{|\lambda|} H'' X$

with the power method and designate the solution $X = [x_1, x_2, \ldots, x_{o_1}(=1), \ldots, x_n]^T$.

where $x_{o_1}$ is the element corresponding to node $o_1$ in Stage 1, and chose the solution so that $x_o$ equals 1. [(8) and Fig. 9 for the network in Fig. 7].

**Stage 7:** First, the tentative score of node $i$ in the subgraph including $o_1$ is defined as $[k_2 + (2\pi - arg(x_i))] \times (k_1 + |x'_i|)$ (Tables 3 and 4). Here, vector $x'_i$ is derived by subtracting from $x_i$ the effects of the vectors corresponding to the nodes that have in-links from node $i$. Second, for node $i$ in the weakly connected graph, the final score is the sum of its scores from each tentative score in each subgraph in the weakly connected graph (Table 5).

In this algorithm, Stage 3 solves the first problem of the previous algorithm. The networks in Figs. 7 and 8 derived by the stage are weakly (unilaterally connected). Using this algorithm, Stage 5
overcomes the second problem of the previous algorithm. Moreover, Stage 7 of this algorithm solves the third problem of the previous algorithm.

Of note $k_1$ is the parameter for the distance from the node with 0 in-links. This distance is defined in terms of the angle from the real axis on the complex plane. With an increase in the value of $k_1$ from 0, the score of the node increases depending on how far away the node is from the node with 0 in-links. Of note $k_2$ is the parameter for the selected in-links to the node. With an increase in the value of $k_2$ from 0, the score of the node increases depending on how small the number of out-links of the nodes on the path from the node with zero in-links to the node, excluding the node itself.

$$X \overset{1}{\rightarrow} \frac{H'X}{|k_1|}$$

$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$

$H = \begin{bmatrix} 0 & s(t+i) & 0 & 0 & 0 & 0 & 0 & 0 \\ s(-i) & 0 & s(t+i) & 0 & 0 & 0 & 0 & 0 \\ 0 & s(-i) & 0 & \frac{s(t+i)}{3} & 0 & \frac{s(t+i)}{3} & 0 & \frac{s(t+i)}{3} \\ 0 & 0 & \frac{s(t+i)}{3} & 0 & s(t+i) & 0 & 0 & 0 \\ 0 & 0 & 0 & s(-i) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & s(t-i) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{s(t-i)}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{s(t-i)}{3} \end{bmatrix}$

$= \begin{bmatrix} 1.0000000 + 0.0000000i \\ 1.5000000 + 0.000000i \\ 2.1556467 + 0.3800987i \\ 3.5626452 + 1.2966968i \\ 0.7660444 + 0.6427876i \\ 1.8956439 + 1.0944505i \\ 0.0000000 + 0.0000000i \\ 1.5000000 + 0.8660254i \end{bmatrix}$

Fig. 9: Complex plane plotting of the 2-dimensional vectors of the nodes shown in Fig. 7 focused on the fourth quadrant
Table 3: Tentative Scores for the node in Fig. 7

<table>
<thead>
<tr>
<th>Node</th>
<th>Score</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000</td>
<td>$k_1$ and $k_2$ are set to 1 and 0, respectively.</td>
</tr>
<tr>
<td>2</td>
<td>0.3490659</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.6981317</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.6981317</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.9308423</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.6981317</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.9308423</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.6981317</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Tentative Scores of the nodes shown in Fig. 8

<table>
<thead>
<tr>
<th>Node</th>
<th>Score</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000</td>
<td>$k_1$ and $k_2$ are set to 1 and 0, respectively.</td>
</tr>
<tr>
<td>2</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.3490659</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.4654211</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.6981317</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.4654211</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.6981317</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.4654211</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Final Scores of the nodes shown in Fig. 6

<table>
<thead>
<tr>
<th>Node</th>
<th>Score</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000</td>
<td>$k_1$ and $k_2$ are set to 1 and 0, respectively.</td>
</tr>
<tr>
<td>2</td>
<td>0.3490659</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.0471976</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.1635528</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.6289740</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1.6289740</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1.1635528</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1.1635528</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Final Scores of the nodes shown in Fig. 6

<table>
<thead>
<tr>
<th>Node</th>
<th>Score</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.520000</td>
<td>$k_1$ and $k_2$ are set to 0.9 and 0.4, respectively.</td>
</tr>
<tr>
<td>2</td>
<td>1.851613</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2.514838</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2.062953</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2.493468</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2.062953</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2.493468</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1.520000</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2.062953</td>
<td></td>
</tr>
</tbody>
</table>
For the directed graph in Fig. 6, we changed \( k_1 \) and \( k_2 \) from 0 to 0.5 with an interval of 0.05. The prototype HermitianStatus III score values of nodes 3 and 5 (the score of node 7) are shown in Fig. 10. Fig. 10 shows that when \( k_2 = 0 \), the score of node 5 is always higher than that of node 3; and if \( k_1 = 0 \), the score of node 3 is higher than that of node 5. The results are obtained because \( k_1 \) is the parameter for the distance from the node with 0 zero in-links, and, \( k_2 \) is the parameter for selected in-links to the node. The fixed top ranking node problem of PageRank with the directed graph in Fig. 1 has been solved by the algorithm.

The rankings by the algorithm score of the nodes from Fig. 6 become the same as those of PageRank at the damping factor value 0.85 when \( k_1 = 0.9 \) and \( k_1 = 0.4 \), as shown in Table 6.

The problems with this algorithm are as follows. First, the second characteristic of PageRank is not realized. Second, when there are two paths from node \( A \) to node \( B \) and the lengths of those paths are different, the two nodes that have an in-link from node \( A \) only take different scores. The result is not equal to PageRank. The final HermitianStatus algorithm solves problems.
4) HermitianStatus

i) Algorithm

Stage 1: Label the nodes in it with zero in-links as $o_1$, $o_2$, ..., $o_i$, ..., $o_q$. If the weakly connected graph does not have a node with zero in-links, add a dummy node to the entire graph and add links from the dummy node to all the nodes in the weakly connected graphs. The same dummy node is used for another weakly connected graph if this weakly connected graph also does not have a node with zero in-links.

Stage 2: Label $e_1$, $e_2$, ..., $e_i$, ..., $e_r$ as the nodes with zero out-links and the nodes with the sum of its out-link and in-link numbers that is not equal to the number of its links in the corresponding undirected graph. If the weakly connected graph does not have a node that satisfies those conditions, add a dummy node to the weakly connected graph and add links to the dummy node from all the nodes in the weakly connected graph.

Stage 3: For each weakly connected graph, induce the path using the nodes that are reachable from $o_1$ and reachable to $e_i$, and create the Hermitian adjacency matrix $H$ for the path of $o_1$ and $e_i$. The same processes are conducted for the remaining $o_2$, ..., $o_i$, ..., $o_q$ and $e_1$, $e_2$, ..., $e_r$.

Stage 4: In $H$, convert each $i$ element to $s(t + i)$ and each $-i$ element to $s(t - i)$, which derives $H'$. Here, $s = \sin\left(\frac{\pi}{2} \times \frac{1}{N}\right)$ and $t = \frac{1}{s} \times \cos\left(\frac{\pi}{2} \times \frac{1}{N}\right)$.

Stage 5: Solve the eigenequation $X = \frac{1}{|\lambda|_1} H' X$ and designate the solution $X = [x_1, x_2, ..., x_{o_1}( = 1), ..., x_n]^T$, where $x_{o_1}$ is the element corresponding to node $o_1$ in Stage 1, and chose the solution so that $x_{o_1}$ equals 1. Then, each element of $x$ is located on the complex plane, and is considered as a two-dimensional vector on the plane. The score of node $i$ in the path including $o_1$ and $e_i$ is defined as $\left\{\frac{2\pi - \arg(x'_i)}{\pi/2N}\right\}^{k_1}$. Here, $x'_i$ is derived by subtracting from $x_i$ the effects of the vectors of the nodes that have in-links from node $i$.

Stage 6: The first tentative score for the node $i$ in the weakly connected graph is the sum of its scores from its every score in every path induced by $o_1$, $o_2$, ..., $o_i$, ..., $o_q$ and $e_1$, $e_2$, ..., $e_i$, ..., $e_r$ containing the node. The second tentative score of the node is derived by dividing the second tentative score by the number of paths from the node to every zero out-link (or highest score node) node that is reachable from the node. The third tentative score of the node $i$ is derived by multiplying the $k_2$ power of the number of in-links to node $i$ to the second tentative score of the node. The fourth score of the node $i$ is the third tentative score divided by $M^{k_3}$. Here, $M$ is the multiplication of the number of out-links of each node, which precede the node, excluding the node itself. The fifth score of the node $i$ is calculated by adding scores to the nodes on the cycles in the weakly connected graph. Those added scores are determined by the maximal score of the fourth tentative scores of the nodes on the circle. The final score of the nodes $i$ is derived by dividing the sixth score by the maximal score among all the sixth scores of all the nodes of the entire network.
**ii) Experimental Evaluation of the HermitianStatus Algorithm**

- HermitianStatus scores dependency on one parameter

Fig. 11 shows the behavior of scores of the nodes depending on $k_1(0 - 1)$ with fixed $k_2 = 0$ and $k_3 = 0$. $k_1$ is the parameter for the distance from the node with no-links. This figure shows that with an increase in the value of $k_1$ from 0 to 1, all scores of all node increase or remain constant, as defined in Stage 5. This result is assured for any node owing to Stage 5 of the algorithm in which $\left( \frac{2\pi - \arg(x_i)}{2N} \right)^{k_1}$ is the $k_1$th power of 0,1,2,3,...and $k_1$ is equal to or greater than 0.

Fig. 12 shows the behavior of scores of the nodes depending $k_2(0 - 1)$ with fixed $k_1 = 0$ and $k_3 = 0$. $k_2$ is the parameter for the number of the in-links of a node. This figure shows that with an increase in the value of $k_2$ from 0 to 1, all scores of all node increase or remain constant, as defined in Stage 6. This result is assured for any node of any graph because of Stage 6 of the algorithm in which the third tentative score of the node $i$ is the $k_2$ power of the second tentative score, and $k_2$ is equal to or greater than 0.

Fig. 13 shows the behavior of scores of the nodes depending $k_3(0 - 1)$ with fixed $k_1 = 0$ and $k_2 = 0$. $k_3$ is the parameter for the selected out-links to the node. This figure shows that with an increase in the value of $k_3$ from 0 to 1, all scores of all nodes decrease or remain constant, as defined in Stage 5. This result is assured for any node of any graph because of Stage 5 of the algorithm in which the fourth score of the node $i$ is the third tentative score divided by $M^{k_3}$. Here, $M$ is the multiplication of the number of out-links of each node and is $k_3$ is equal or greater than 0.
- HermitianStatus scores dependency on two parameters

Fig. 14 shows the behavior of scores of the nodes depending on $k_1(0 - 1)$ and $k_2(0 - 1)$ with fixed $k_3 = 0$. We observe that with an increase in $k_1$, the scores of the nodes increase depending on their distance from the node with 0 on-links. In addition, we see that $k_2$ increases the score of node 3 depending on its in-link number. These results are assured for any node of any graph because of Stages 5 and 6.

Horizontal Direction (to right): $k_1(0.00, 0.25, 0.50, 0.75, 1.000)$,
Vertical Direction (to bottom): $k_2(0.00, 0.25, 0.50, 0.75, 1.000)$,
with fixed $k_3=0.00$, Blue (low score) to Orange (high score)

Fig. 14: Effects of $k_1$ and $k_2$ on the scores of the nodes of the network $k_1$ and $k_2$ with fixed $k_3$
Fig. 15 shows the behavior of scores of the nodes depending $k_1(0 - 1)$ and $k_3(0 - 1)$ with fixed $k_2 = 0$. We see that with an increase in $k_1$, the scores of the nodes increase depending on their distance from the node with 0 on-links. In addition, we observe that $k_3$ increases the score of the node (e.g., Nodes 4 and 5) decrease depending on the number of out-links of the nodes on the paths from 0 in-link nodes to these nodes. Those results are assured for any node of any graph because of Stages 5 and 6.

**Fig. 15**: Effects of $k_1$ and $k_3$ on the scores of the nodes of the network $k_1$ and $k_3$ with fixed $k_2$
Fig. 16 shows the behavior of scores of the nodes depending $k_2(0 - 1)$ and $k_3(0 - 1)$ with fixed $k_1 = 0$. We observe that with an increase in $k_2$, the score of node 3 increases depending on its in-link number (this may be hard to recognize with human eyes). In addition, we observe that $k_3$ increases the score of the node (e.g., Nodes 4 and 5) decrease depending on the number of out-links of the nodes on the paths from the 0 in-link nodes to these nodes. Those results are assured for any node of any graph because of Stages 5 and 6.

Horizontal Direction (to right): $k_2(0.00, 0.25, 0.50, 0.75, 1.000)$, Vertical Direction (to bottom): $k_3(0.00, 0.25, 0.50, 0.75, 1.000)$, with fixed $k_1 = 0.00$, Blue (low score) to Orange (high score)

Fig. 16: Effects of $k_2$ and $k_3$ on the scores of the nodes of the network $k_2$ and $k_3$ with fixed $k_1$
- HermitianStatus scores for two specific nodes depending on two parameters

Fig. 17 shows the behavior of the scores of nodes 3 (red) and 5 (blue) of the graph in Fig. 2 depending on $k_1(0 \sim 1)$ and $k_2(0 \sim 1)$ with fixed $k_3 = 0$. We observed that with an increase in $k_1$, the score of node 5 is higher than that of node 3. This occurs because the sum of distances from nodes 1 and 8 to 5 is 8, and this is larger than that of node 3, which is 6. In addition, we observe that with an increase in $k_2$, the score of node 3 is higher than that of node 5. This occurs because node 3 has three in-links, and node 5 has only one in-link. These results are assured for any nodes of any graphs because of Stages 5 and 6.

Fig. 18 shows the behavior of the scores of nodes 3 (red) and 5 (blue) of the graph in Fig. 2 depending on $k_1(0 \sim 1)$ and $k_3(0 \sim 1)$ with fixed $k_2 = 0$. We observe that with an increase in $k_1$, the score of node 5 is higher than that of node 3. This occurs because the sum of distances from nodes 1 and 8 to 5 is 8, which is larger than that of node 3, which is 6. In addition, we observe that with an increase in $k_3$, the score of node 3 is higher than that of node 5. This occurs because while all nodes on the path from nodes 1 and 8 to 3 have one out-link only, the nodes on the paths from nodes 1 and 8 to node 5 include node 3 that have three out-links. Those results are assured for any nodes of any graphs because of Stages 5 and 6.

Fig. 18 shows the behavior of the scores of nodes 3 (red) and 5 (blue) of the graph in Fig. 2 depending on $k_2(0 \sim 1)$ and $k_3(0 \sim 1)$ with fixed $k_1 = 0$. We observe that with an increase in $k_2$, the score of node 3 increases, and the score of node 5 is constant. This occurs because node 3 has two in-links and node 5 has only one in-link. In addition, we observe that with an increase in $k_3$, the score of node 5 decreases and the score of node 3 is constant. This occurs because while all the nodes on the paths from nodes 1 and 8 to 3 have only one out-link, the nodes on the paths from nodes 1 and 8 to node 5 include node 3 that has three out-links. These results are assured for any nodes of any graphs because of Stages 5 and 6.
iii) Case of a non-weakly connected network

Fig. 1 shows the network is a non-weakly connected graph and has four weakly connected components. For the network nodes, the maximal Spearman’s correlation coefficient between PageRank scores at the damping factor 0.85 and HermitianStatus scores is 0.9474212, which is realized by ninth $k_1$, ninth $k_2$, and first $k_3$; each of them has ten values between 0 and 1 with the equal intervals. Fig. 19 shows the plot of the PageRank scores and the HermitianStatus scores.

![Fig. 19: 2-Dimensional Plot of the HermitianStatus Scores and PageRank Scores of the Nodes Shown in Fig. 1](image)

V. Discussion

For the high cost calculation problem of PageRank, two approaches were used. In the first approach, we focus on weakly connected components in the net. However, according [6], 91% web sites on the Internet are in the weakly connected network. However, we can consider that in the large weakly connected component, not all hyperlinks among the web sites can be used because of blocked routers and other issues [16]. Therefore, we need to consider the large component as the number of weakly connected components. Therefore, by decomposing the Internet into weakly connected networks and by the applying HermitianStatus algorithm to each of them, we can decrease the calculation cost of ranking web sites, compared to that of PageRank. Moreover, when only one of weakly connected components in the net changes the number of nodes and link relations among them, PageRank has to create a new $G$, which corresponds to the updated network aiming to obtain the new ranking of the nodes in the network. In turn, the HermitianStatus algorithm requires only a new $H$ of the component to determine the ranking of the nodes in the network, because the updated scores of the nodes in the component are comparable to those of the nodes in the remaining unchanged weakly connected components. Compared to PageRank, the advantage of the proposed algorithm that is based on the Hermitian method allows decreasing the calculation cost for large-sized networks. Moreover, the scores of the nodes in the entire network can be systematically arranged using the three parameters. When we compare HermitianStatus and the approach of K. Avrachenkov and N. Litvak[5], the former algorithm have the advantage because it does not need to apply the damping factor and avoids the problems caused by this factor. Moreover, using HermitianStatus, we can systematically change the scores of the nodes of the network by changing the parameters $k_1$, $k_2$, and $k_3$. 
Regarding the second approach, when we focus strong components [7], we notice the problem that in the Internet, there must be web-sites that do not belong to strong components. In the second approach, this issue needs to be elaborated. Therefore, we should focus on weakly connected components.

VI. Conclusion

In this study, the author proposes the HermitianStatus algorithm for ranking nodes in the network. This algorithm has the advantage over Google PageRank because the Hermitian approach does not need a damping factor. HermitianStatus can score a note in a weakly connected component without transforming into the strongly connected network with a damping factor. The proposed algorithm can reproduce the result of PageRank well. Moreover, the three parameters of HermitianStatus can systematically change the scores of the nodes of the network. When we apply the proposed method to the search engine the benefits are high.

Acknowledgment

This research was supported by Nanzan University Pache Research Subsidy I-A-2 for the 2019 academic year.

Patent application: 2019-230822

References


