Method for efficiently orthogonalizing the eigenvectors of the Laplacian matrix to estimate social network structure

Naoki Hirakura\textsuperscript{1a)}, Chisa Takano\textsuperscript{2b)}, and Masaki Aida\textsuperscript{1c)}

\textsuperscript{1} Graduate School of Systems Design, Tokyo Metropolitan University
6-6 Asahigaoka, Hino, Tokyo 191-0065, Japan

\textsuperscript{2} Graduate School of Information Sciences, Hiroshima City University
3-4-1 Ozuka-Higashi, Hiroshima 731-3194, Japan

\textsuperscript{a)} hirakura-naoki@ed.tmu.ac.jp
\textsuperscript{b)} takano@hiroshima-cu.ac.jp
\textsuperscript{c)} aida@tmu.ac.jp

Abstract: The structure of social networks or human relationships is difficult to understand since we cannot observe their links and link weights directly. The network resonance method was proposed to obtain information on the unknown Laplacian matrix representing the social network structure. This method extracts information on the eigenvalues and eigenvectors of the Laplacian matrix by observing user dynamics on social networks. The original Laplacian matrix can be reconstructed if all eigenvalues and eigenvectors are known. However, the network resonance method has a problem: the information available about eigenvectors is limited to the absolute value of each element. Therefore, to determine the Laplacian matrix, it is necessary to determine the signs of each element of all eigenvectors. However, sign determination incurs the computation cost of the order of \(O(2^n)\) for each of \(n\) eigenvectors. This paper proposes a method to determine the signs of each eigenvector element efficiently. The main idea of the method is to generate \(n^2 - n\) different sign determination problems for \(n\) eigenvectors and to solve them in parallel. All that is required is to obtain \(n\) different eigenvectors determined in the shortest time from the \(n^2 - n\) different sign determination problems. Since the ratio of the number of sign determinations completed in the method is \(1/(n - 1)\), its efficiency rises with the number of network users. In addition, simulations on networks generated by the BA model show that proposed method offers sign determination in polynomial time.

Key Words: social network, combinatorial optimization, branch and bound

1. Introduction
Social networks represent human relationships, but understanding their structure is difficult since we cannot observe their links and link weights directly. An alternative approach is structure estimation by
indirect measurements of social networks. One interesting candidate is intended to obtain information on the eigenvalues and eigenvectors of the Laplacian matrix of the social networks from observations of user dynamics on the social networks. The Laplacian matrix of the social network structure can be estimated if all the eigenvalues and eigenvectors of the Laplacian matrix are determined. This method is based on a theoretical framework that represents user dynamics on social networks [1, 2].

The network resonance method [3, 4] has been proposed to allow observations of user dynamics to be used to estimate the information of eigenvalues and eigenvectors of the Laplacian matrix that represents the social networks. If we know all the eigenvalues and eigenvectors, the original Laplacian matrix can be determined by spectral decomposition [5]. Unfortunately, the network resonance method provides only a part of eigenvector information, that is, it gives only the absolute values of elements of the eigenvalues. That is, the network resonance method cannot determine the signs of each element of the eigenvectors making a new method necessary.

The network resonance method can be applied to network topologies called symmetrizable directed graphs [2]. A symmetrizable directed graph has a graph structure represented by a Laplacian matrix that can be converted into a symmetric matrix by similarity transformation, and a symmetrizable directed graph can be treated as a symmetric matrix like an undirected graph. Since social network structures described by symmetrizable directed graph gives stable user dynamics observed in usual [2], we assume social networks to be symmetrizable directed graphs. In addition, we assume the Laplacian matrix does not have duplicated eigenvalues. This assumption is justified by taking the engineering approach to social network analysis [2]. Therefore, the orthogonality of eigenvectors is guaranteed.

It is possible to determine the signs from the orthogonality of eigenvectors, but this process incurs a lot of computation cost. For a network with \( n \) nodes, the corresponding Laplacian matrix is an \( n \times n \) square matrix, so we need to determine the combination of signs for the \( n \) elements of the \( n \) different eigenvectors. There are \( O(2^n) \) combinations for each eigenvector of \( n \) eigenvectors, so determining the signs requires the exponential time of \( O(n \times 2^n) \) in the worst case. Since social networks are large-scale and \( n \) takes large values, a new approach is needed that can avoid the computational complexity of \( O(n \times 2^n) \).

In this paper, we propose a method that can efficiently determine the signs of the elements of the eigenvectors. Basically, we use the branch and bound method to determine the sign of each eigenvector. This is not a special and does not reduce computational complexity itself. Our main idea is as follows. As we need to solve the sign determination problem for \( n \) different eigenvectors, we generate \( n^2 - n \) different sign determination problems from the original \( n \) different problems. We solve the \( n^2 - n \) different sign determination problems simultaneously. The time taken to solve each sign determination problem depends on cases. Exponential time will be needed in the worst case, but some cases can be solved rapidly. Our idea is that only \( n \) different eigenvectors need be determined from the \( n^2 - n \) different sign determination problems. Accordingly, we stop the computation process when we have the signs of \( n \) different eigenvectors. This means we can select the results of cases that have extremely short computation times. We show that the proposed method completes the sign determination process in time polynomial to \( n \).

The rest of this paper is organized as follows. In Sec. 2, after showing the sign determination problem and how to solve it, we propose a method that can efficiently determine the signs of the absolute values of the eigenvector elements, by generating \( n^2 - n \) different sign determination problems and solving them in parallel. In Sec. 3, simulations of network models generated by the BA model show that proposed method completes sign determination in polynomial time. Sec. 4 concludes this work.

2. Proposed method to determine signs of eigenvector elements

2.1 Formulation and algorithm example

In this section, we detail an algorithm that can determine the signs of the elements of given vectors \( v^+ = (|v_0(0)|, |v_0(1)|, \ldots, |v_0(n−1)|) \) \((\mu = 0, 1, \ldots, n−1)\), which consist of the absolute values of the eigenvector elements obtained by the network resonance method. First of all, we choose two vectors \( v^+_{\mu} \) and \( v^+_{\nu} \) from all \( n \) given vectors and calculate the product for each pair of elements for the chosen
two vectors and sort the products in descending order, as follows
\[ r_{\mu\nu} := (|\tilde{v}_\mu(0)\tilde{v}_\nu(0)|, \ldots, |\tilde{v}_\mu(n-1)\tilde{v}_\nu(n-1)|). \] (1)
The relative signs of the elements can be determined by the orthogonality of the eigenvectors \( v_\mu \) and \( v_\nu \), as in
\[ v_\mu \cdot v_\nu = \delta_{\mu\nu}, \]
where \( \delta_{\mu\nu} \) is the Kronecker delta. The relative sign is positive if the corresponding elements of the two vectors have the same sign, and negative otherwise. Let \( \sigma_{\mu\nu} \) be the vector of relative signs corresponding to vector (1). Then, the sign determination problem can be expressed as the problem of finding \( \sigma_{\mu\nu} \) that minimizes \( \phi_{\mu\nu} \), as
\[ \phi_{\mu\nu} := r_{\mu\nu} \cdot \sigma_{\mu\nu} = \sum_{i=0}^{n-1} r_{\mu\nu}(i) \sigma_{\mu\nu}(i) \]
\[ = \sum_{i=0}^{n-1} |\tilde{v}_\mu(i)\tilde{v}_\nu(i)| \sigma_{\mu\nu}(i), \] (2)
where \( \sigma_{\mu\nu}(i) = \pm 1 \).

It can be said that (2) is a number partitioning problem (NPP) for real numbers. NPP is the combinatorial optimization problem to partition a set of positive integers into two sets whose sums of elements in each subset are as close as possible. For more detail, see the Appendix.

In the sign determination process, we initially set \( \sigma_{\mu\nu}(0) \) to +1 or −1, and then determine the signs from \( \sigma_{\mu\nu}(1) \) to \( \sigma_{\mu\nu}(n-1) \). To find the optimal solution of the combination of signs, we search depth-first by applying the branch and bound method to the binary search tree that represents the combinatorial optimization problem to partition a set of positive integers into two sets whose sums of elements are as close as possible. For more detail, see the Appendix.

1. If \( \phi_{\mu\nu} = 0 \), the combination of signs has been found whose inner product of two eigenvectors is 0; so the algorithm is stopped.
2. If the sign of \( \phi_{\mu\nu} \) changes, \( |\phi_{\mu\nu}| \) will never be smaller in that subtree, so all subsequent combinations are pruned.

As an example, we search for the optimal combination of \( \sigma_{\mu\nu} \) corresponding to \( r_{\mu\nu} = (6, 4, 3, 2, 1) \). Let \( \sigma_{\mu\nu}(0) = -1 \) and \( \sigma_{\mu\nu} = (-1, -1, -1, -1, -1) \) be the initial state. Figure 1 shows the binary search tree that represents the combinations of \( \sigma_{\mu\nu} \). In this figure, \( \sigma_{\mu\nu}(i) \) means the \( i \)-th element, numbers in circles mean the step number in the search. In this search, choosing a left branch means that the corresponding element of \( \sigma_{\mu\nu} \) changes to +1, while choosing a right branch means that the corresponding element of \( \sigma_{\mu\nu} \) changes to −1. At first, we choose the right branch means that the corresponding element of \( \sigma_{\mu\nu} \) changes to +1, while choosing a right branch means that the corresponding element of \( \sigma_{\mu\nu} \) changes to −1. At first, we choose the right branch, which means \( \sigma_{\mu\nu}(1) = +1 \), and so \( \phi_{\mu\nu} = -8 \). Similarly, if we again choose the left branch, which means \( \sigma_{\mu\nu}(2) = +1 \), we have \( \phi_{\mu\nu} = -2 \). We again choose the left branch, which means \( \sigma_{\mu\nu}(3) = +1 \), the value of \( \phi_{\mu\nu} \) changes from negative to positive \( \phi_{\mu\nu} = +2 \). At this point, the pruning rule stops branching and we backtrack and choose the right branch, which means \( \sigma_{\mu\nu}(3) = -1 \). Next we choose the left branch which means \( \sigma_{\mu\nu}(4) = +1 \) and \( \phi_{\mu\nu} = 0 \), so again the stopping rule is applied. This search determines that \( \sigma_{\mu\nu} = (-1, +1, +1, -1, +1) \).

2.2 Efficient orthogonalization
We propose here the method that can efficiently determine the sign of the elements of the \( n \) eigenvectors. Our method executes multiple sign determination algorithms in parallel, and outputs only the combinations whose calculations are found to be completed more rapidly. There are \( n(n-1)/2 \) pairs of different vectors from \( n \) vectors \( v_\mu^+ \) (\( \mu = 0, 1, \ldots, n-1 \)). Given that are two initial settings (the first element has the same or different sign), there are \( n(n-1) \) combinations. We run these \( n(n-1) \) sign determination algorithms in parallel, and stop all of them when the signs of the first \( n \) different eigenvectors are determined.

Figure 2 shows an image of the relationship between the calculation time taken by the full sign de-
termination algorithms and by our proposal with early termination. In the worst case, the calculation time is exponential, $O(2^n)$, with respect to the number of nodes $n$. Also, the average calculation time increases exponentially. Since estimating the Laplacian matrix requires only $n$ different eigenvectors, by using the most rapidly obtained $n$ eigenvectors, we might be able to stop the algorithms in polynomial time, as indicated by the red curve in Fig. 2. In Fig. 2, the total area of each distribution is proportional to $n(n-1)$, while the red area is proportional to $n$. Thus the ratio of the red area to the whole area of the distribution is $1/(n-1)$, it decreases as $n$ increases. Since the calculation amounts of the sign decision problems conducted in parallel are diverse and the number of sign determination problems to be solved is only $n$, we can use only the $n$ different results of eigenvectors with the determined signs with smaller calculation amount. If the calculation amount of such sign determination problems increase in polynomial order with respect to $n$, like the red curve shown in Fig. 2, the eigenvectors required to estimate the structure of the social network can be obtained in polynomial time.

To implement our method, we execute $n(n-1)$ problems in separate threads, one step at a time, in parallel. We calculate the first step of all problems and then move to the second step. We repeat this until we have solved $n$ different sign determination problems. Here, step number equals the number of search nodes of the binary search tree.

Finally, after calculating the relative signs of $n$ distinct eigenvectors, we determine the sign of each eigenvector.

We show the procedure and an example of obtaining the signs of each eigenvector from the relative signs. In this procedure, we use the following two facts. One is that the elements of the eigenvector associated with the zero eigenvalue have the same sign and we can set them as positive. Incidentally, the Laplacian matrix always has the eigenvalue of zero because its row sum is zero, so the elements of the eigenvector associated with zero eigenvalue are the same value. The second is that we can choose the sign of at least one element of the eigenvector. Accordingly, we choose the first element of each eigenvector as positive. Since the sign of $v_\mu(1)$ is positive for all $\mu$, the relative sign $\sigma_{0\mu}$ represents the sign of each element of $v_\mu$. The sign of $v_\nu (\nu \neq \mu)$ can be obtained by using the relative sign $\sigma_{\mu\nu}$ and $v_\mu$. By appropriately repeating this procedure, it is possible to determine the sign of the
elements of all \( n \) eigenvectors.

We provide an example of the above procedure. We consider the case of \( n = 3 \). We assume that the absolute values of the eigenvector elements are as follows

\[
\begin{align*}
 v_0^+ &= (1, 1, 1), \\
 v_1^+ &= (3, 2, 1), \\
 v_2^+ &= (1, 4, 5).
\end{align*}
\]

From these three vectors, six sign determination problems can be generated. These are problems of determining relative signs \( \sigma_{01}, \sigma_{02}, \sigma_{10}, \sigma_{12}, \sigma_{20}, \text{ and } \sigma_{21} \). We assume that the calculations of relative signs \( \sigma_{01} \) and \( \sigma_{12} \) are completed earlier, and the results are

\[
\begin{align*}
 \sigma_{01} &= (-1, +1, +1), \\
 \sigma_{12} &= (+1, -1, +1).
\end{align*}
\]

Here, if the first element of \( v_0 \) is +1, it can be said that \( v_0 = (+1, +1, +1) \). We can determine the sign of \( v_1 \) from the relative sign \( \sigma_{01} \) and eigenvector \( v_0 \). From \( \sigma_{01} \), we can see that \( v_1(0) \) has
different sign from $v_0(0)$, $v_1(1)$ has same sign as $v_0(1)$ and $v_1(2)$ has the same sign as $v_0(2)$. Therefore, $v_1 = (-3, +2, +1)$. Since the sign of $v_1$ has been determined, we can determine the sign of $v_2$ from the relative sign $\sigma_{12}$ and eigenvector $v_1$. From $\sigma_{12}$, we can see that $v_2(0)$ has the same sign as $v_1(0)$, $v_2(1)$ has different sign from $v_1(1)$, and $v_2(2)$ has the same sign as $v_1(2)$. Therefore, $v_2 = (-1, -4, +5)$. We can determine the signs for all eigenvectors in this manner from the relative signs of $n$ distinct eigenvectors.

3. Experimental evaluation of the proposed method

In this section, we evaluate the calculation cost of the proposed method using a network model generated by the Barabási-Albert model (BA model) [6, 7]; it well replicates the scale-free nature of social networks [8]. Note that there are three initial nodes in the BA model and each additional node has three links; all link weights are 1. The number of computation steps to be counted is the total number of branch selections in the binary search tree representing the combination of relative signs of each element. We generate $n$ $(n-1)$ sign determination problems, and the first step of searching for the optimal solution for each problem is executed in order. When the first steps of all problems have been determined, the processing of the second step is executed. We repeat this process until the optimal solution is found. When the sign determination problem of $n$ sets is completed, the process is terminated. The total number of steps in this process is taken as the calculation cost metric.

We compare the calculation costs of the proposed method and the simple pruning and stopping method. The first is calculation cost incurred by sign determination of $n$ different eigenvectors using the proposed method. The second is the simple approach of solving $n$ different sign determination problems sequentially for all pairs $(v_0, v_\mu)$ $(\mu = 0, 1, \ldots, n-1)$. For the proposed method, the calculation amount is roughly $O(n^3)$ when $n$ is sufficiently large. On the other hand, in the simple sequential calculation approach, it is impossible to avoid the huge number of calculations, and in some cases the number of steps exceeded $2^{31} - 1$. From these comparisons, we can recognize that the proposed method takes polynomial time with respect to $n$, and so is the more efficient method for reducing the calculation cost than the simple branch and bound method.

To verify the ability of the proposed method to reduce computational complexity, we investigate
the calculation amount of the orthogonalization process of eigenvectors for various network models obtained by different BA model parameters. Let \( m_0 \) be the number of the initial nodes and \( m \) be the number of additional links for the BA model parameters. The experiment was conducted with \( m_0 = m = 1, \ldots, 9 \), with 10 to 200 nodes in the network.

According to Fig. 4, when the number of nodes is large enough, the computational complexity is \( O(n^3) \) regardless of the values of \( m_0 \) and \( m \).

4. Conclusion
This paper introduced an efficient method for determining the signs of eigenvector elements when the absolute values of the elements of the eigenvectors are known. The number of calculation steps taken by the method were determined. This paper proposed an efficient method for determining the signs of eigenvector elements when the absolute values of the elements of the eigenvectors are given. For eigenvectors with \( n \) elements, there are \( 2^n \) combinations of signs for each of \( n \) eigenvectors. Therefore, the simple sequential calculation approach takes exponential time of \( n \) in the worst case for determining the correct combinations of signs. Our method executes sign determination calculations for \( n(n-1) \) different pairs for \( n \) eigenvectors in parallel. By using only the unique \( n \) pairs whose calculation is completed first, the solution is achieved in the polynomial time of \( O(n^3) \).

Acknowledgments
This research was supported by Grant-in-Aid for Scientific Research (B) No. 17H01737 (2017–2019) and Grant-in-Aid for Scientific Research (C) No. 18K11271 (2018–2020) from the Japan Society for the Promotion of Science (JSPS).

Appendix

A. Number Partitioning Problem
The number partitioning problem (NPP) is a combinatorial optimization problem that finds a partition that divides a set of positive integers into two sets while minimizing the difference between the element sums of each set [9–11]. This can be defined as follows. Given a set of \( N \) positive integers as \( S = \{n_1, n_2, \ldots, n_N\} \), find the partition \( R \) of the subset of \( S \), that minimize the discrepancy \( E \),
\[ E = \left| \sum_{i \in R} n_i - \sum_{i \notin R} n_i \right|, \quad (A-1) \]

is minimized. (A-1) can be converted by \( \sigma_i \)

\[ E = \left| \sum_{i=1}^{N} n_i \sigma_i \right|, \quad \sigma_i = \begin{cases} +1 & i \in R \\ -1 & i \notin R \end{cases} \quad (A-2) \]

Comparing (A-2) and (2), we can see that (2) is an extension of (A-2). NPP divides the positive integer set, while the sign determination problem divides the real set. In NPP terminology, a perfect partition occurs when \( E = 0 \) or \( E = 1 \). It can be said that the optimum division was found at this time. Although NPP is known to be NP-complete, some polynomial-time approximations are known. Using an algorithm called complete Karmarkar-Karp (CKK) to solve NPP, it is shown that the number of search nodes increases in proportion to \( N \) if \( N \) is sufficiently large [11].

On the other hand, we need to solve \( n \) sign determination problems of \( n(n-1) \) problems and proposed method take calculation time proportional to \( O(n^3) \). The stopping and pruning rules in our sign determination algorithm have the same computational complexity reduction effect close to CKK. However, the significant idea of the proposed method is not in the stopping and pruning rules, but in to generate \( n(n-1) \) different problems and to select \( n \) sign determination problems completed earlier.

Incidentally, in the sign determination problem, there is always a way to decide signs such that the value of (2) becomes 0 due to the orthogonality of eigenvectors. In the sign determination problem, it is necessary to set a threshold value because the elements are real numbers. In this work, we assume that the optimal sign allocation is found when \( \phi_{\mu \nu} < 10^{-7} \).

References

[1] M. Aida, C. Takano, and M. Murata, “Oscillation model for network dynamics caused by asymmetric node interaction based on the symmetric scaled Laplacian matrix,” The 12th International Conference on Foundations of Computer Science (FCS 2016), pp. 38–44, 2016.
[2] M. Aida, C. Takano, and M. Murata, “Oscillation model for describing network dynamics caused by asymmetric node interaction,” IEICE Transactions on Communications, vol. 101-B, no. 1, pp. 123–136, 2018.
[3] S. Furutani, C. Takano, and M. Aida, “Network resonance method: Estimating network structure from the resonance of oscillation dynamics,” IEICE Transactions on Communications, vol. 102-B, no. 4, pp. 799–809, 2019.
[4] S. Furutani, C. Takano, and M. Aida, “Method for estimating the eigenvectors of a scaled Laplacian matrix using the resonance of oscillation dynamics on networks,” Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, pp. 615–618, ACM, 2017.
[5] F.R. Chung and F.C. Graham, “Spectral graph theory,” American Mathematical Soc., 1997.
[6] R. Albert and A.L. Barabási, “Statistical mechanics of complex networks,” Reviews of Modern Physics, vol. 74, no. 1, p. 47, 2002.
[7] A.L. Barabási and R. Albert, “Emergence of scaling in random networks,” Science, vol. 286, no. 5439, pp. 509–512, 1999.
[8] A.L. Barabási and E. Bonabeau, “Scale-free networks,” Scientific American, vol. 288, no. 5, pp. 60–69, 2003.
[9] I.P. Gent and T. Walsh, “Phase transitions and annealed theories: Number partitioning as a case study,” ECAI, pp. 170–174, PITMAN, 1996.
[10] J.P. Pedroso and M. Kubo, “Heuristics and exact methods for number partitioning,” European Journal of Operational Research, vol. 202, no. 1, pp. 73–81, 2010.
[11] S. Mertens, “The easiest hard problem: Number partitioning,” Computational Complexity and Statistical Physics, vol. 125, no. 2, pp. 125–139, 2006.