2018 International Symposium on Nonlinear Theory and Its Applications,
NOLTA2018, Tarragona, Spain, September 2-6, 2018

Efficient Orthogonalizing the Eigenvectors of the Laplacian Matrix to Estimate Social Network Structure

Naoki Hirakura† Chisa Takano‡ and Masaki Aida†

†Graduate School of Systems Design, Tokyo Metropolitan University
Hino, Tokyo 191-0065, Japan
‡Graduate School of Information Sciences, Hiroshima City University
Hiroshima 731-3194, Japan
Email: hirakura-naoki@ed.tmu.ac.jp, takano@hiroshima-cu.ac.jp, aida@tmu.ac.jp

Abstract—It is inherent difficult to directly quantify the structure of the social networks that describe human relations. The network resonance method was proposed to elucidate the unknown Laplacian matrix representing social network structure. This method gives information on the eigenvalues and eigenvectors of the Laplacian matrix from observations of the dynamics of a social network. If all the eigenvalues and eigenvectors are known, the original Laplacian matrix can be determined. One problem with the network resonance method is that only limited information about eigenvectors can be acquired, and only the absolute values of the vector elements are available. Therefore, to determine the Laplacian matrix, it is necessary to determine the signs of each element of the eigenvectors; this task has order of $O(2^n)$ given the combinations of n users for every eigenvector. This paper proposes a method that determines eigenvector element signs efficiently by running a sign determination algorithm in parallel and uses only those with fewer calculation amount. The proposal executes sign determination in polynomial time. We also reduce the calculation overhead by applying compressed sensing; the computational complexity of sign determination is reduced to almost $O(n^2)$.

1. Introduction

The social network is intended to represent human relationships but it is inherently difficult to directly observe a complete network because it is large and complicated. A more practical approach is to indirectly estimate the structure of the network. One interesting candidate, the network resonance method, derives information on eigenvalues and eigenvectors of the Laplacian matrix from observations of dynamics on the social network for an unknown Laplacian matrix representing the social network structure [1, 2, 3]. This approach is valid because if all eigenvalues and eigenvectors are known, the original matrix can be determined. Social networks have the characteristic being scale-free [4, 5]. This means that there are few large-degree nodes (here, nodes having a large the number of connected links), while most nodes have few links. Therefore, the link structure of social networks is sparse. Utilizing this feature, applying the compressed sensing to a method for indirectly estimating the structure of the social network is an attractive approach. We can expect to estimate the sparse link structure of social networks from incomplete set of eigenvalues and eigenvectors.

The network resonance method can determine the structure of a social network indirectly; for this, it obtains information on eigenvalues and eigenvectors of the unknown Laplacian matrix describing the structure of a social network from observations of the network's dynamics [1, 2, 3]. Once all eigenvalues and eigenvectors are known, the original Laplacian matrix can be determined. One issue with the network resonance method is that only limited information can be acquired about eigenvectors, and only the absolute values of the vector elements can be known. Therefore, to determine the Laplacian matrix, it is necessary to determine signs of each element of the eigenvectors. It is possible to determine signs from the orthogonality of eigenvectors, but this incurs significant computation overhead. For a network with n nodes, the corresponding Laplacian matrix is a square matrix of $n \times n$, so we need to determine the combination of signs of the n elements of the n different eigenvectors. There are $O(2^n)$ combinations of n users for every eigenvector. Therefore, calculating the signs requires exponential time of $O(n\,2^n)$ in the worst case.

In this paper, we propose a method that can efficiently determine the signs of elements of the eigenvectors by running multiple sign determination algorithms in parallel, and by stopping the process at the time when the desired number of vectors are obtained. We show that the proposed method completes the process of sign determination in polynomial time to n. In addition, since Laplacian matrixes that describe social networks are sparse in general, we also reduce the calculation amount by applying compressed sensing; this reduces the computational complexity of the sign determination to almost $O(n^2)$.

The rest of this paper is organized as follows. In Sec. 2, we introduce a method that can efficiently determine the sign of the absolute value of the elements of eigenvectors. In Sec. 3, we evaluate the computational complexity of our sign determination algorithm. Sec. 4 draws the conclusions of this work.

2. Proposed Method to Determine Signs of Eigenvector Elements

In this section, we detail an algorithm that can determine the signs of the elements of vector $\boldsymbol{v}_\mu^+ = (|v_\mu(0)|, |v_\mu(1)|, \ldots, |v_\mu(n-1)|)$, which consists of the absolute value of the eigenvector's element estimated by the network resonance method. First of all, we choose two eigenvectors from among all eigenvectors and compute the product for each pair of elements of the chosen two vectors and sort the products in descending order, as follows

$$\boldsymbol{r}_{\mu\nu} := (\,|\bar{v}_\mu(0)\,\bar{v}_\nu(0)|,\ |\bar{v}_\mu(1)\,\bar{v}_\nu(1)|,\ \ldots,\ |\bar{v}_\mu(n-1)\,\bar{v}_\nu(n-1)|\,). \tag{1}$$

The relative sign of the elements can be determined by using the orthogonality of the eigenvectors $\boldsymbol{v}_\mu^+$ and $\boldsymbol{v}_\nu^+$, as in

$$\boldsymbol{v}_\mu \cdot \boldsymbol{v}_\nu = \delta_{\mu\nu},$$

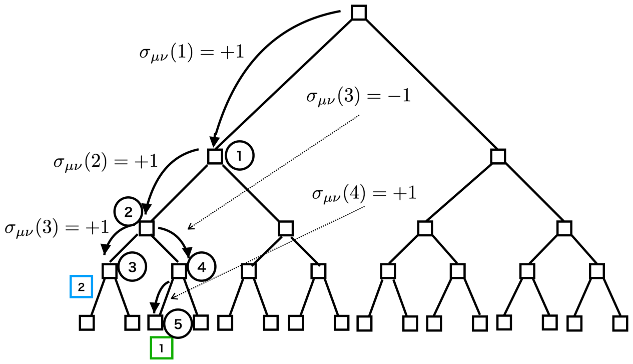
where $\delta_{\mu\nu}$ is the Kronecker delta. The relative sign is positive if the corresponding elements of the two vectors are the same sign, or negative otherwise. Let $\boldsymbol{\sigma}_{\mu\nu}$ be the vector of relative signs associated with vector (1). Then, the sign determination problem can be expressed as the problem of finding $\sigma_{\mu\nu}$ that minimizes $\phi_{\mu\nu}$, as

$$\begin{aligned}\phi_{\mu\nu} := \left|\boldsymbol{r}_{\mu\nu} \cdot \boldsymbol{\sigma}_{\mu\nu}\right| &= \left|\sum_{i=0}^{n-1} r_{\mu\nu}(i)\,\sigma_{\mu\nu}(i)\right| \\ &= \left|\sum_{i=0}^{n-1} \bar{v}_\mu(i)\,\bar{v}_\nu(i)\,\sigma_{\mu\nu}(i)\right|, \quad \sigma_{\mu\nu}(i) = \pm 1.\end{aligned} \tag{2}$$

In the sign determination process, we initially set $\sigma_{\mu\nu}(0)$ to +1 or −1, and then determine the signs from $\sigma_{\mu\nu}(1)$ until $\sigma_{\mu\nu}(n-1)$. 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 combination of elements of the $\sigma_{\mu\nu}$. In the search, we introduce the following pruning and stopping rule to reduce the calculation load.

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 the $\sigma_{\mu\nu}$ corresponding to $r_{\mu\nu} = (6, 4, 3, 2, 1)$. Let $\sigma_{\mu\nu}(0) = -1$ and $\boldsymbol{\sigma}_{\mu\nu} = (-1, -1, -1, -1, -1)$ be the initial state. Figure 1 shows the binary search tree that represents the combinations of $\sigma_{\mu\nu}$. In this search, choosing a left branch means that the corresponding element of $\boldsymbol{\sigma}_{\mu\nu}$ is +1, while choosing a right branch means that the corresponding element of $\boldsymbol{\sigma}_{\mu\nu}$ is −1. At first, we choose the left 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 the stopping rule is triggered. This search determines that $\boldsymbol{\sigma}_{\mu\nu} = (-1, +1, +1, -1, +1)$.



$\sigma_{\mu\nu}(1) = +1$
$\sigma_{\mu\nu}(3) = -1$
$\sigma_{\mu\nu}(2) = +1$
$\sigma_{\mu\nu}(4) = +1$
$\sigma_{\mu\nu}(3) = +1$
1
2
3
4
5
2
1

Figure 1: An example of the binary search tree in the sign determination algorithm

Next, we explain two efficient methods for determining the sign of the elements of the n eigenvectors. The first method executes multiple sign determination algorithms in parallel, and use only the combinations that can be calculated more rapidly. There are $n(n-1)/2$ pairs of different vectors from n vectors $\boldsymbol{v}_\mu^+$ ($\mu = 0, 1, \ldots, n-1$). Considering two kinds of initial settings, that is, the case where the first element has the same sign and the case where it has different sign, there are $n(n-1)$ combinations. We run $n(n-1)$ of sign determination algorithms in parallel, and stop all of them when the signs of the first n different eigenvectors are determined.

Fig.2 shows the relationship between the calculation amount distribution of the sign determination algorithms and the partial calculation amount of n different eigenvectors that can be calculation earlier. In the worst case, a sign determination calculation requires exponential time of $O(2^n)$ with respect to the number of nodes n, but since the actual calculation amount is varied, some will be completed more rapidly. Since Laplacian matrix estimation requires only n different eigenvectors, by using most rapidly obtained n eigenvectors, we might be able to stop the algorithms in polynomial time, as indicated by the red line in 2. Note that ratio of n to $n(n-1)$ decreases as n increases. To achieve this approach, we execute $n(n-1)$ sign determination algorithms step by step in parallel.

The second method is to apply compressed sensing to the estimation of the Laplacian matrix. Since the Laplacian matrix of a social network is sparse, applying compressed

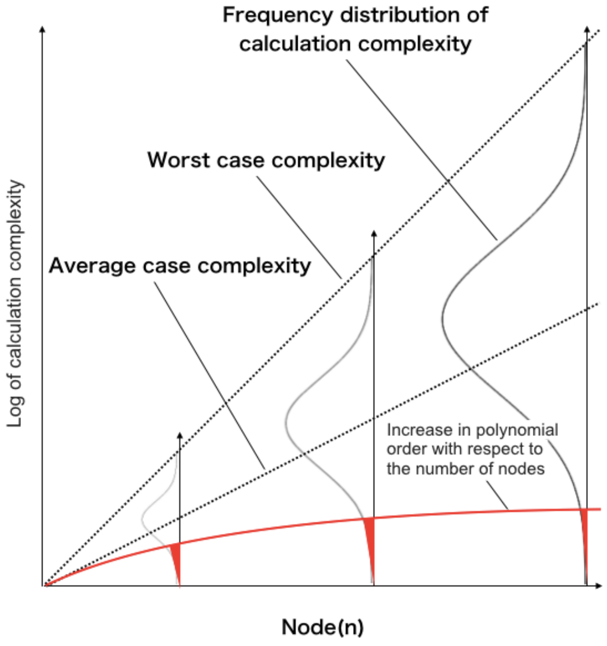

Frequency distribution of calculation complexity
Worst case complexity
Average case complexity
Log of calculation complexity
Increase in polynomial order with respect to the number of nodes
Node(n)

Figure 2: Intuitive image of calculation load reduction

sensing can reduce the number of eigenvectors required in estimating the Laplacian matrix [7]. By using this method, the original Laplacian matrix can be estimated from a subset of the eigenvalues and eigenvectors. The larger the number of nodes is, the greater the sparseness of the Laplacian matrix is, and thus the estimation efficiency increases. To evaluate the efficiency of this method, we experimentally examine how many eigenvalues and eigenvectors are required to estimate a Laplacian matrix. Let α be the ratio of the number of eigenvectors required for estimating a Laplacian matrix with compressed sensing to the number of all eigenvectors, n.

The experimental conditions are as follows. We define the estimation as having succeed if the values of all elements of the Laplacian matrix estimated by compressed sensing are less than 0.01 from the true values. Ratio α is obtained from the smallest number of eigenvectors when the estimation is successful, and we investigate the dependency of α on n. In the experiment, we use a network model generated by the Barabási-Albert model (BA model) [8]; it well reflects the scale-free nature of social networks. In addition, there are three initial nodes in BA model and each additional node has three links; all link weights are 1. Figure 3 shows the experiment's result. The n dependency of α is given by

$$\alpha = -0.047\ \log n + 0.73 \qquad (n \geq 50) \qquad (3)$$

Therefore, the Laplacian matrix can be estimated more efficiently as the number of nodes increases. Note that we do not need to determine the sign of all n eigenvectors. Since the determination process of the sign of elements of eigenvectors is terminated when the computation for the first

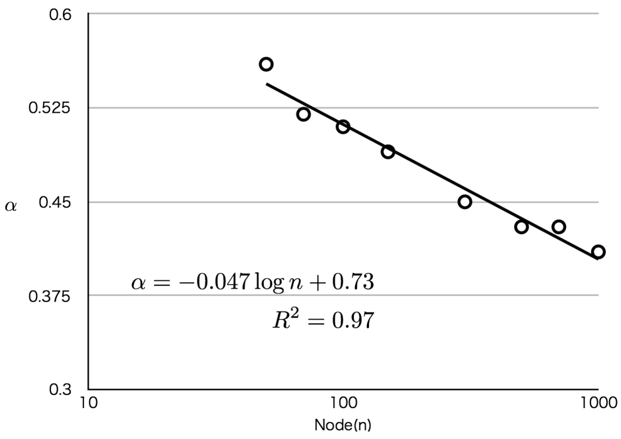


Figure 3: n dependency of the ratio α

$[\alpha\, n]$ ($\alpha < 1$) of the eigenvectors is completed, the computational complexity can be further reduced.

Finally, after calculating relative signs of distinct n (or $[\alpha\, n]$) eigenvectors, we determine the sign of each eigenvector. The first element, $v_\mu(0)$ ($\mu = 0, 1, \ldots, n-1$), of each eigenvector can be set positive. Since it is known that the signs of the element of the eigenvector associated with the zero eigenvalue $\boldsymbol{v}_0$ is the same, the sign of the elements of $\boldsymbol{v}_\mu$ can be determined from the relative sign $\boldsymbol{\sigma}_{0\mu}$ of the product of elements ($\boldsymbol{v}_0$, $\boldsymbol{v}_\mu$). By repeating the procedure, we can determine the sign of the n (or $[\alpha\, n]$) eigenvectors.

3. Experimental Evaluation of the Proposed Method

In this section, we evaluate the computational complexity of the proposed method. In this experiment, we use a network model generated by the BA model; the details are the same as shown in the previous section. The number of computation steps counted is the total number of search nodes of the binary search tree that represents the combinations of each element of relative sign. The number of computation steps is the number of steps in the process of sign determination of $n(n-1)$ combinations of vectors until the time of terminate the process.

The comparison uses three different sign determination algorithms, and evaluates their computational complexity. The first is to determine the signs for the n eigenvectors using the proposed method. The second is to determine the signs only for the $[\alpha\, n]$ eigenvectors using the proposed method based on compressed sensing. The third is the simple approach of applying the sign determination algorithm sequentially to the pairs ($\boldsymbol{v}_0, \boldsymbol{v}_\mu$) ($\mu = 0, 1, \ldots, n-1$) of the eigenvectors.

Figure 4 shows the results. In determining the signs of the n ($\alpha = 1$) eigenvectors, the computational complexity is roughly $\boldsymbol{O}(n^3)$ when n is sufficiently large. When using the n dependency of α in the compressed sensing based method, the calculation complexity is almost $\boldsymbol{O}(n^2)$ when n

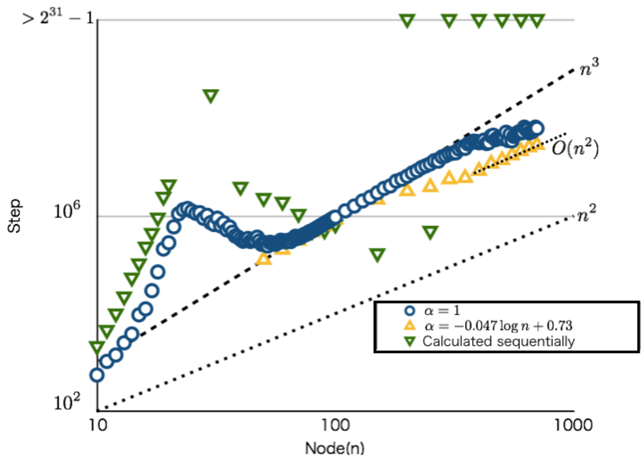


Figure 4: The calculation complexity of the sign determination algorithm

is sufficiently large ($n \geq 450$). Finally, we explain the result of calculating the sign determination algorithm of the n pairs without using the proposed method. In this case, it is impossible to avoid the huge numbers of calculations, some values were greater than $2^{31} - 1$. From these comparisons, we can recognize that as the proposed method takes polynomial time with respect to n it is the preferred choice.

4. Conclusion

In this paper, in the situation that the absolute values of the elements of the eigenvectors are known, we proposed an efficient method for determining their signs and evaluated its computational complexity. For eigenvectors with n elements, there are 2^n combinations of signs. Therefore, it takes exponential time in n in the worst case to identify the correct combinations of signs. To ease this complexity, we proposed the following two methods. The first method conducts sign determination calculations for $n(n-1)$ different pairs for n eigenvectors in parallel. By using only the n pair that were calculated first, the solution could be achieved in the polynomial time of $O(n^3)$. The second method applies compressed sensing to Laplacian matrix estimation because the Laplacian matrix for social networks is, in general, sparse. By using this method, in the case that n is sufficiently large, the computational complexity of the sign determination algorithm is reduced to almost $\boldsymbol{O}(n^2)$. Our results show that we the structure of social networks can be efficiently estimated from the eigenvalues and absolute values of elements of the eigenvectors that can be observed by applying the network resonance method.

Acknowledgment

This research is supported by a Grant-in-Aid for Scientific Research (B) No. 17H01737 (2017–2019) from the Japan Society for the Promotion of Science (JSPS).

References

[1] Masaki Aida, Chisa Takano, and Masayuki Murata, "Oscillation model for describing network dynamics caused by asymmetric node interaction," *IEICE Transactions on Communications*, vol. E101-B, no. 1, pp. 123–136, 2018.

[2] Satoshi Furutani, Chisa Takano, and Masaki Aida, "Proposal of the network resonance method for estimating eigenvalues of the scaled laplacian matrix," In *Intelligent Networking and Collaborative Systems (INCoS), 2016 International Conference on*, pp. 451–456, 2016.

[3] Satoshi Furutani, Chisa Takano, and Masaki Aida, "Method for estimating the eigenvectors of a scaled Laplacian matrix using the resonance of oscillation dynamics on networks," In proc. *IEEE/ACM ASONAM 2017*, 2017.

[4] Lada A. Adamic, Rajan M. Lukose, Amit R. Puniyani, and Bernardo A. Huberman, "Search in power-law networks," *Physical review E*, vol. 64, no. 4, 046135, 2001.

[5] Réka Albert and Albert-László Barabási, "Statistical mechanics of complex networks," *Rev. Mod. Phys.*, vol. 74, pp. 47–97, Jan 2002.

[6] Yoshiyuki Kabashima, "Statistical-mechanical approach to compressed sensing," *The Brain & Neural Networks*, vol. 17, no. 2, pp. 70–78, 2010.

[7] Shun Sugimoto and Masaki Aida, "Estimating the structure of social networks from incomplete sets of observed information by using compressed sensing," In proc. *IEEE Latin-Americal Conference on Communications (IEEE LATINCOM 2017)*, 2017.

[8] Albert-László Barabási and Réka Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.