NOLTA, IEICE

Paper

Reconstructing the Laplacian matrix to estimate social network structure by using compressed sensing

Shun Sugimoto [1 a)] *and Masaki Aida* [1 b)]

[1] *Graduate School of Systems Design, Tokyo Metropolitan University Hino-shi 191-0065, Japan*

[a)] *sugimoto-shun@ed.tmu.ac.jp*
[b)] *aida@tmu.ac.jp*

Received January 22, 2019; Revised May 12, 2019; Published October 1, 2019

Abstract: For complex large scale networks, like social networks, it is usually impossible to observe complete information about their topology structure or link weight directly. A recent proposal, the network resonance method, can estimate the eigenvalues and eigenvectors of the Laplacian matrix for representing network structure, by using the resonance phenomena of oscillation dynamics on networks. However, it is generally not possible to observe all the eigenvalues and eigenvectors. This paper uses compressed sensing to create a new method of reconstructing the original Laplacian matrix from a partial set of its eigenvalues and eigenvectors. Since very few node pairs in social networks have links, we can expect that compressed sensing will be effective. The estimated Laplacian matrix of a social network enables to us to determine its structure and link weights.

Key Words: compressed sensing, Laplacian matrix, eigenvalue, eigenvector

1. Introduction

Due to the spread of the Internet, information exchange among users through on-line social networks is becoming active. One significant feature of social networks is their scale-free topology structure, that is, the degree distribution of nodes follows a power law [1]. Accordingly, just a few nodes have a very large number of links while most nodes have only a few links. For example, in Social Networking Service (SNS), while most users have hundreds of followers at most, there are celebrities who have millions of followers. If we consider the relationship of follow as a link, we can understand the structure of scale-free networks in social networks from SNS data [2, 3]. It is known that scale-free networks appear frequently in the topologies of complex networks in various fields other than social networks [4].

Networks topology structure can be represented by a square matrix by setting the link information from node i to node j as the (i, j) component of the matrix. Spectral graph theory is a method of analyzing the characteristics of networks by eigenvalues and eigenvectors of the matrix representing the network structure, and it is applicable to many problems including clustering of networks, graph drawing, graph cut, node coloring, and image segmentation [5, 6]. In graph signal processing, the

Nonlinear Theory and Its Applications, IEICE, vol. 10, no. 4, pp. 496–506 ©IEICE 2019 DOI: 10.1587/nolta.10.496

graph Fourier transformation plays importance role, and it is introduced based on the spectral graph theory [7, 8].

Since the eigenvalues and eigenvectors include information of the corresponding matrix, the original matrix can be reproduced if all the eigenvalues and eigenvectors are given. This means that if all the eigenvalues and eigenvectors are given, we can acquire complete information of the topology structure of the network.

In social networks, the strength of the influence of the human relationship between two users is represented as the weight of link between the corresponding user nodes. If we can measure such quantities directly, we can determine the topology structure of the network, and at the same time, determine the square matrix representing the network structure. For example, influence strength could be taken to be proportional to the volume of traffic exchanged between the user nodes, which can be easily measured. Unfortunately, the strength of influence in human relations is different in concept from the amount of traffic exchanged between user nodes. As this is also true for other measurable quantities, we have to determine link weights in a different way.

We introduced the oscillation model on networks as a model that can describe network dynamics generated from the influences among user nodes [9, 10]. The importance of the oscillation model lies in the relationship between the oscillation energy of each node and node centrality. The oscillation energy of each node gives the degree centrality or the betweenness centrality in simple cases, and gives a generalized notion of node centrality [11, 12]. Based on the oscillation model, we proposed a method to estimate the eigenvalues and eigenvectors of the Laplacian matrix that represents network structure called the network resonance method [13, 14]. This method is based on the resonance phenomena of network behavior with respect to an external input that varies at a certain frequency. Even if we do not know the Laplacian matrix, knowing all the eigenvalues and eigenvectors of the Laplacian matrix allows us to reproduce the original Laplacian matrix. To know all the eigenvalues and eigenvectors of the Laplacian matrix means knowing link presence and weights between user nodes for the corresponding social network.

When estimating the eigenvalues of the Laplacian matrix by the network resonance method, it is not always possible to estimate all eigenvalues due to factors such as the duplication of eigenvalues and the unknown strength of oscillation damping. Of course, it is also not always possible to estimate all the eigenvectors. Therefore, it is desirable to be able to reproduce the original Laplacian matrix from just some of the eigenvalues/eigenvectors of the Laplacian matrix. In addition, since estimation of eigenvalues/eigenvectors involves errors, the influence of these errors on networks structure estimation must be offset.

Almost all components of the Laplacian matrix are occupied by zero components, because social networks are scale-free. This property, zero components are dominant, is called sparsity. We can expect that applying compressed sensing exerts a good effect in this situation.

The compressed sensing is a technique that leverages sparsity to attain reasonable estimation accuracy from the fewest observations possible; it has been studied actively in various fields in recent years [15], including information systems [16], signal processing [17], and communications systems [18]. In particular in communication networks, the compressed sensing is used in efficient observations for data gathering in wireless sensor networks [19] and for classification of link quality of service (QoS) in network tomography [20].

In this paper, we investigate the method that uses compressed sensing to reconstruct the original Laplacian matrix from just some of the eigenvalues and eigenvectors. This is basic research on estimating social network structure from some partial information observed through the network resonance method.

This paper is organized as follows. Section 2 explains the fundamental properties of spectral graph theory and compressed sensing to aid the reader's understanding. Section 3 proposes a method for reconstructing the Laplacian matrix that uses compressed sensing. Section 4 discusses the efficiency of the proposed method.

2. Preliminaries

2.1 Laplacian matrix

Let us consider undirected graph $G = G(V, E)$ with n nodes, where $V = \{1, 2, \ldots, n\}$ is a set of nodes and E is a set of links. Let w_{ij} (> 0) be the link weight between node i and j. The (weighted) adjacency matrix $\boldsymbol{A} = [A_{ij}]$ for undirected graph $G(V, E)$ is the $n \times n$ matrix defined as

$$A_{ij} := \begin{cases} w_{ij} & ((i, j) \in E), \\ 0 & ((i, j) \notin E). \end{cases} \tag{1}$$

Since $w_{ij} = w_{ji}$, adjacency matrix $\boldsymbol{A}$ is real and symmetric. Next, node i has degree d_i defined as

$$d_i := \sum_{j \in \partial i} w_{ij}, \tag{2}$$

where ∂i is the set of nodes adjacent to node i. The degree matrix is defined as $\boldsymbol{D} := \mathrm{diag}(\{d_i\})$, and the Laplacian matrix of $G(V, E)$ is defined as

$$\boldsymbol{L} := \boldsymbol{D} - \boldsymbol{A}. \tag{3}$$

Note that the Laplacian matrix is real and symmetric, so the eigenvalues of $\boldsymbol{L}$ are real numbers and eigenvectors can be chosen as the eigenbasis. In addition, $\boldsymbol{L}$ is a positive-semidefinite matrix, and it is known that the minimum eigenvalue is 0 [6]. Here, we write the n eigenvalues of $\boldsymbol{L}$ in ascending order as

$$0 = \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{n-1}.$$

The eigenvectors $\boldsymbol{v}_\mu$ associated with eigenvalues λ_μ $(\mu = 0, 1, \ldots, n-1)$ are chosen as follows.

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

where $\delta_{\mu\nu}$ denotes the Kronecker delta.

If all eigenvalues and eigenvectors are known, using diagonal matrix $\boldsymbol{\Lambda} := \mathrm{diag}(\lambda_0, \lambda_1, \cdots, \lambda_{n-1})$ and orthogonal matrix $\boldsymbol{P} := [\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n-1}]$, Laplacian matrix $\boldsymbol{L}$ is expressed as

$$\boldsymbol{L} = \boldsymbol{P}\,\boldsymbol{\Lambda}\,{}^t\boldsymbol{P}. \tag{4}$$

Therefore, the Laplacian matrix can be reproduced from the information expressed by eigenvalues and eigenvectors. In addition, once the Laplacian matrix is known, we can also know the topology structure of the network.

2.2 Compressed sensing

Compressed sensing can estimate unknown vectors from linear observations. Let $\boldsymbol{x}$ $(\in \mathbb{R}^n)$ be an n-dimensional unknown vector. At this time, linear observations for $\boldsymbol{x}$ can be written as inner product $\boldsymbol{a} \cdot \boldsymbol{x}$, where $\boldsymbol{a}$ $(\in \mathbb{R}^n)$ is an observation vector. When this linear observation is performed m times, the observation matrix $\boldsymbol{F} \in \mathbb{R}^{m \times n}$ is expressed as

$$\boldsymbol{F} = \begin{bmatrix} {}^t\boldsymbol{a}_1 \\ {}^t\boldsymbol{a}_2 \\ \vdots \\ {}^t\boldsymbol{a}_m \end{bmatrix} \tag{5}$$

by setting each observation vector as $\boldsymbol{a}_i$ $(i = 1, 2, \ldots, m)$. The m linear observations for the unknown vector $\boldsymbol{x}$ can be defined as

$$\boldsymbol{y} = \boldsymbol{F}\,\boldsymbol{x}, \tag{6}$$

where the m-dimensional vector $\boldsymbol{y}$ $(\in \mathbb{R}^m)$ is a vector having $\boldsymbol{a}_i \cdot \boldsymbol{x}$ $(i = 1, 2, \ldots, m)$ in each component as a result of the m linear observations.

When the observation matrix $\boldsymbol{F}$ is known, (6) is equivalent to a set of simultaneous linear equations in which each component of unknown vector $\boldsymbol{x}$ is a variable. Since the number of observations, m, corresponds to the number of equations, and $\boldsymbol{x}$ is an n-dimensional vector, it is necessary to satisfy at least $m \geq n$ in order to solve the equations. Even if unknown vector $\boldsymbol{x}$ is sparse, the use of compressed sensing allows us to obtain a solution even if $m < n$. In other words, compressed sensing is a method for solving underdetermined linear equations, and regularizing $\|\boldsymbol{x}\|_1$ as a cost function as follows.

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \quad \text{subj. to} \quad \boldsymbol{y} = \boldsymbol{F}\,\boldsymbol{x}, \tag{7}$$

where $\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i|$. Regularization via (7) is called ℓ_1 reconstruction and is known to be very effective in compressed sensing. When there are k non-zero components in unknown vector $\boldsymbol{x}$, $\boldsymbol{x}$ is said to be k-sparse. In general, smaller values of k make estimation possible with fewer observations, m. Extensive studies have been conducted on compressed sensing, for details please refer to [16, 17].

3. Estimation of the Laplacian matrix by compressed sensing

In this section we propose a method that uses compressed sensing in estimating a Laplacian matrix from just a few eigenvalues and eigenvectors. The eigenvalues and eigenvectors are estimated by the network resonance method [13, 14]. Eigenvalue estimation by using the network resonance method is outlined in Appendix A. In this section, matrices and vectors are denoted by [ ] and ( ), respectively.

3.1 Proposed method

First, we denote the Laplacian matrix as

$$\boldsymbol{L} = \begin{bmatrix} {}^t\boldsymbol{l}_0 \\ {}^t\boldsymbol{l}_1 \\ \vdots \\ {}^t\boldsymbol{l}_{n-1} \end{bmatrix}, \tag{8}$$

where ${}^t\boldsymbol{l}_i$ ($i = 0, 1, \ldots, n-1$) is a row vector of Laplacian matrix $\boldsymbol{L}$. The characteristic equation of the Laplacian matrix is given as

$$\boldsymbol{L}\,\boldsymbol{v}_i = \lambda_i\,\boldsymbol{v}_i. \tag{9}$$

Using (8), the left-hand side of (9) is expressed as

$$\boldsymbol{L}\,\boldsymbol{v}_i = \begin{bmatrix} {}^t\boldsymbol{l}_0 \\ {}^t\boldsymbol{l}_1 \\ \vdots \\ {}^t\boldsymbol{l}_{n-1} \end{bmatrix} \boldsymbol{v}_i = \begin{pmatrix} \boldsymbol{l}_0 \cdot \boldsymbol{v}_i \\ \boldsymbol{l}_1 \cdot \boldsymbol{v}_i \\ \vdots \\ \boldsymbol{l}_{n-1} \cdot \boldsymbol{v}_i \end{pmatrix}. \tag{10}$$

We denote each component of eigenvector $\boldsymbol{v}_i$ as $v_i(j)$ ($j = 0, 1, \ldots, n-1$), and extract the j-th component from (9) and (10) as

$$\lambda_i\, v_i(j) = \boldsymbol{l}_j \cdot \boldsymbol{v}_i. \tag{11}$$

Since the right-hand side of (11) is an inner product, we can reverse the order as

$$\lambda_i\, v_i(j) = \boldsymbol{v}_i \cdot \boldsymbol{l}_j. \tag{12}$$

Since there are n eigenvalues ($i = 0, 1, \ldots, n-1$), it can be considered that there are n relational expressions for each j. Here, we select m equations from among these n relational expressions and write the eigenvalues and eigenvectors as $\bar{\lambda}_k$ and $\bar{\boldsymbol{v}}_k = (\bar{v}_k(0), \bar{v}_k(1), \ldots, \bar{v}_k(n-1))$ ($k = 1, 2, \ldots, m$), respectively. By using the selected eigenvalues and eigenvectors, we can cast the relationship of (12) as a vector as follows.

$$\begin{pmatrix} \bar{\lambda}_1\,\bar{v}_1(j) \\ \bar{\lambda}_2\,\bar{v}_2(j) \\ \vdots \\ \bar{\lambda}_m\,\bar{v}_m(j) \end{pmatrix} = \begin{bmatrix} \bar{\boldsymbol{v}}_1 \\ \bar{\boldsymbol{v}}_2 \\ \vdots \\ \bar{\boldsymbol{v}}_m \end{bmatrix} \boldsymbol{l}_j. \tag{13}$$

Now, if we regard the left-hand side of (13) as $\boldsymbol{y}$ and the right-hand side as $\boldsymbol{F} = {}^t[\bar{\boldsymbol{v}}_1, \bar{\boldsymbol{v}}_2, \ldots, \bar{\boldsymbol{v}}_m]$ and $\boldsymbol{x} = \boldsymbol{l}_j$, (13) has the same form as (6). Thus, when $\boldsymbol{l}_j$ is sparse, we can estimate the j-th row vector of the unknown Laplacian matrix from m ($< n$) sets of eigenvalues/eigenvectors. In addition, if this estimation is extended to all row vectors ($\boldsymbol{l}_0, \boldsymbol{l}_1, \ldots, \boldsymbol{l}_{n-1}$), we can reproduce the entire Laplacian matrix.

This method can be applied to not only Laplacian matrices but also other real symmetric matrices. Unfortunately, it is difficult to apply this method to asymmetric Laplacian matrices, e.g. general directed graphs, because eigenvalues are generally complex numbers. However, this method can be applied to the symmetrizable directed graphs [10].

3.2 Related method

First, we summarize the requirements of the proposed method proposed in the previous subsection. In reproducing the matrix by using the diagonal matrix and the orthogonal matrix as in (4), all the eigenvalues and eigenvectors are used. However, not all the eigenvalues/eigenvectors can be estimated by the network resonance method, and it is likely that some estimations will be somewhat inaccurate. Thus it is difficult to reconstruct Laplacian matrixes by using (4).

We now consider the low-rank approximation of real symmetric matrixes as another estimation method. (4) can be transformed into

$$\boldsymbol{L} = \boldsymbol{P}\,\boldsymbol{\Lambda}\,{}^t\boldsymbol{P} = \sum_{\mu=0}^{n-1} \lambda_\mu\,\boldsymbol{v}_\mu\,{}^t\boldsymbol{v}_\mu. \tag{14}$$

Since eigenvalue λ_μ with low value contributes little to $\boldsymbol{L}$, it is possible to approximate the real symmetric matrix by considering only terms with large eigenvalues. Unfortunately, with this method, the estimation accuracy decreases, when large eigenvalues are missing or are quite noisy. The proposed method, on the other hand, does not require all of the large eigenvalues, and makes it possible to select eigenvalues flexibly.

4. Evaluations

In this section, we demonstrate the performance of the proposed method. In order to clarify basic features for reconstruction of the Laplacian matrix by using compressed sensing, we generate network models as pseudo social networks and evaluate the proposed method by using the network models. Many complex networks including social networks are known to exhibit the scale-free characteristic. Based on this, our evaluations use network models with scale-free property. In particular, we estimate the Laplacian matrix for scale-free networks generated by the BA model [1].

When creating a network with the BA model, the initial state is a complete graph with 3 nodes, and every time one node is added, links are established to three nodes selected by preferential attachment. The number of nodes in our network model is $n = 100$.

The following evaluation results are the averages of the experimental results gained from ten independent scale-free networks generated independently.

4.1 Estimation accuracy in each eigenvalues patterns

In the proposed method, estimation accuracy greatly differs depending on the eigenvalues and the eigenvectors used for reconstruction. Therefore, in this section, we classify combinations of the eigenvalues used for the reconstruction into several typical categories and investigate which eigenvalues patterns improve the estimation performance.

All the link weights of our network model are set to 1; the Laplacian matrix is unknown. In addition, we consider the situation where m pairs of eigenvalues and eigenvectors of the Laplacian matrix can be selected; they are assumed to be noise-free.

Here, we introduce the Vector Estimation Success Rate as the performance index. The success of vector estimation is defined as the case where the difference between the actual value and the estimated value of all components of the vector is less than 0.01. Figure 1 plots the Vector Estimation Success Rate versus the number of pairs of eigenvalues and eigenvectors, m. We use different methods to select m pairs and the results are plotted using different colors. Selection methods of eigenvalues are listed in Table I. The mixing ratio of the eigenvalues in Mixture is based on the combination that yielded the highest estimated performance as determined in the experiments.

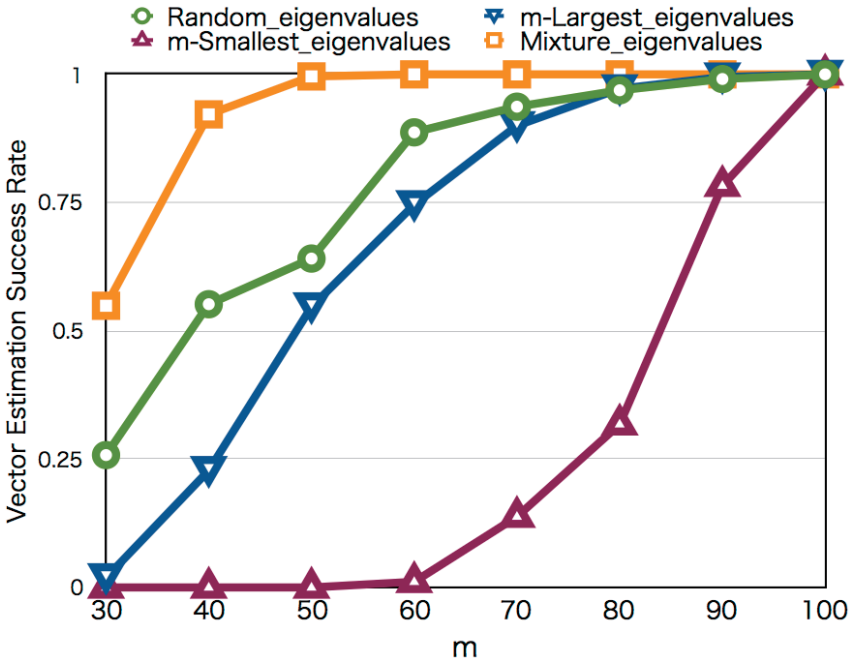

Fig. 1. Vector estimation success rate of each pattern.

Table I. Selection methods of eigenvalues.

Random	m eigenvalues at random
m-Largest	m eigenvalues in ascending order
m-Smallest	m eigenvalues in descending order
Mixture	$(m \times 1/8)$ in ascending, $(m \times 7/8)$ in descending

From Fig. 1, m-Smallest ($\triangle$) yields the smallest Vector Estimation Success Rate regardless of m. This result implies that estimation accuracy does not improve even if more small eigenvalues used. In addition, m-Largest ($\bigtriangledown$) always higher Vector Estimation Success Rate than m-Smallest. However, Random ($\bigcirc$) offers better performance, suggesting that larger eigenvalues and smaller eigenvalues are needed for higher performance. Mixture ($\square$) achieves the best results; even if only half of the eigenvalues and eigenvectors are used, $(m = 50)$, almost all vectors can be accurately estimated. The optimal mixing ratio was $7 : 1$, but ratios of $6 : 1$ or $5 : 1$ yielded very similar performance.

The reason why Mixture achieves best results is considered as follows. The eigenvalues of the Laplacian matrix gives hierarchical structure with respect to spatial scales. Larger eigenvalues and their associated eigenvectors have information of fine structure of the network. On the other hand, smaller eigenvalues and their associated eigenvectors have information of coarse grained structure of the network. Since reproducing the structure of entire network requires both fine structure and coarse grained information, Mixture gives best results.

4.2 Estimation by the eigenvalues and the eigenvectors that contain noise

In this subsection, we consider the situation where in the given eigenvalues and eigenvectors contain noise. We generated a scale-free network by the BA model and assigned uniform random numbers in the range of $[0.1, 1.1]$ as link weights. As noise, we added a random number following the normal distribution ($\mu = 0$, $\sigma = 0.01$) to all m eigenvalues and all components of the m eigenvectors. Since the eigenvalues and eigenvectors contain noise, it is impossible to obtain a complete solution. Here we evaluate performance by using the absolute error between the actual value and the estimated value of each component of the Laplacian matrix.

Figure 2 plot the average error and the maximum error versus m; Mixture was used to select the eigenvalues. Average Error denotes the average of absolute error of all components of the estimated matrix. Max Error denotes the largest error among all components. From these figures, we can recognize that the maximum error decreases with observation number m, but the average error is minimum around $m = 70$. Therefore, using many eigenvalues for estimation does not necessarily lead to an improvement in accuracy if noise is present.

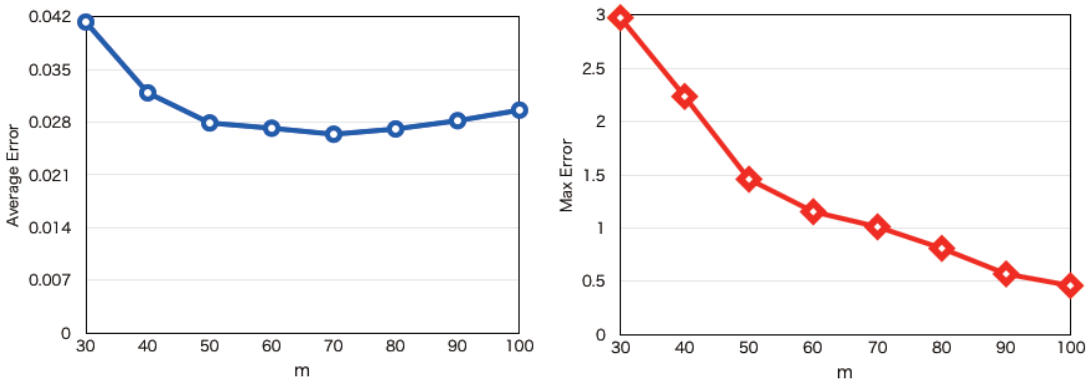

Fig. 2. Average and maximum error of the estimated Laplacian matrix.

Figures 3 and 4 are the heat maps showing the error of all components of the estimated matrix for $m = 40$ and $m = 80$, respectively. Note that they have different scales. The order of components is the order of node addition in the BA model. Therefore, the nodes corresponding to the upper left components are hub nodes.

These figures show that the errors fall as the observation number is increased from $m = 40$ to $m = 80$. In addition, for $m = 40$, the errors concentrate on diagonal components, but for $m = 80$, many errors are found in non-diagonal components of hub nodes. In other words, it is considered that

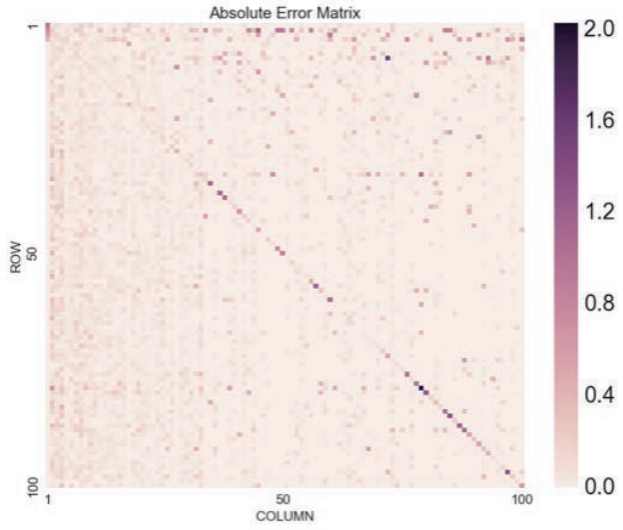

Fig. 3. Error matrix between actual matrix and estimated matrix ($m = 40$).

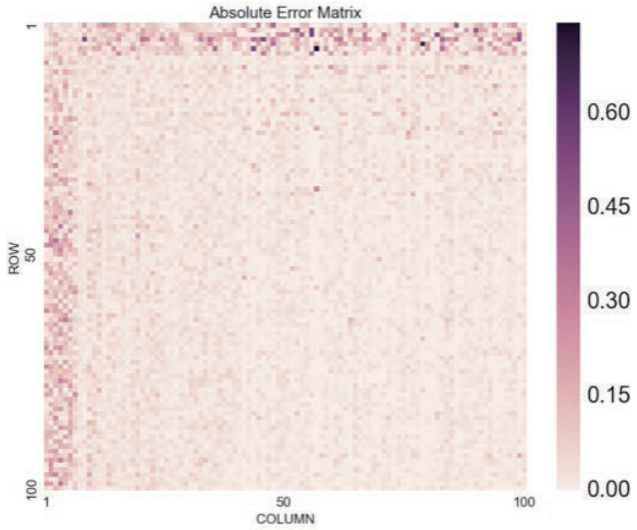

Fig. 4. Error matrix between actual matrix and estimated matrix ($m = 80$).

row vectors with high sparsity can be accurately estimated by increasing m.

4.3 Extension of the estimation method considering existence of non-sparse row vectors

In this section, we extend the proposed method described in the previous section in order to improve the reconstruction performance of the Laplacian matrix. Since social networks have scale-free characteristic, there are several hub nodes. The hub nodes have links to a very large number of nodes and the corresponding row vector $\boldsymbol{l}_i$ is not necessarily sparse. Therefore, the compressed sensing approach might not work effectively. In order to mitigate this effect, we extend the estimation method of each row vector, shown in the previous section. Let the left-hand side of (13) be $\boldsymbol{y}_i$ and ${}^t[\bar{\boldsymbol{v}}_0, \bar{\boldsymbol{v}}_1, \ldots, \bar{\boldsymbol{v}}_{m-1}]$ of the right-hand side be $\boldsymbol{F}$. The proposed method can be expanded as follows.

$$\begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_n \end{pmatrix} = \begin{bmatrix} \boldsymbol{F} & & & 0 \\ & \boldsymbol{F} & & \\ & & \ddots & \\ 0 & & & \boldsymbol{F} \end{bmatrix} \begin{pmatrix} \boldsymbol{l}_1 \\ \boldsymbol{l}_2 \\ \vdots \\ \boldsymbol{l}_n \end{pmatrix}. \tag{15}$$

The matrix of the right-hand side is as follows.

$$\begin{bmatrix} \boldsymbol{F} & & & 0 \\ & \boldsymbol{F} & & \\ & & \ddots & \\ 0 & & & \boldsymbol{F} \end{bmatrix} = [\mathcal{F}_{ij}] = \begin{cases} \bar{v}_s(t), & (i = kn + s,\ j = kn + t + 1) \\ 0, & (\text{otherwise}) \end{cases}. \tag{16}$$

Where $[\mathcal{F}_{ij}]$ is the component of this matrix ($i = 1, \ldots, n^2$, $j = 1, \ldots, n^2$) and $\bar{v}_s(t)$ is the component of the selected eigenvectors ($k = 0, \ldots, n-1$, $s = 1, \ldots, n$, $t = 0, \ldots, n-1$). n denotes the number of nodes. (15) also has the same form as compressed sensing, we can estimate all the row vectors of Laplacian matrix $\boldsymbol{L}$, simultaneously. By reconstructing all row vectors simultaneously, even if the Laplacian matrix has low sparsity, compressed sensing can be expected to work effectively. In addition, as (15) contains all components of Laplacian matrix $\boldsymbol{L}$, we can add some limiting conditions for reconstruction. From the definition of the Laplacian matrix, $\boldsymbol{L}$ for an undirected graph is a symmetric matrix, and

$$L_{ij} = L_{ji}.$$

In addition, the diagonal components of $\boldsymbol{L}$ are

$$L_{ii} = -\sum_{k(\neq i)} L_{ik}.$$

If we use these equations as the constraint condition, Laplacian matrix $\boldsymbol{L}$ can be reconstructed more accurately because the candidate solutions are limited effectively.

Figure 5 shows the average error of the method of reconstructing each row vector and the method of reconstructing all the row vectors, simultaneously. All the link weights of our network model are set to 1; the pattern of eigenvalues uses Mixture. Also, in the extended method for simultaneous reconstruction of all the row vectors, we added all the above constraints. From this figure, in situations where the number of m is the same, we can see that the extended method for simultaneous reconstruction of all the row vectors can most reduce the average error.

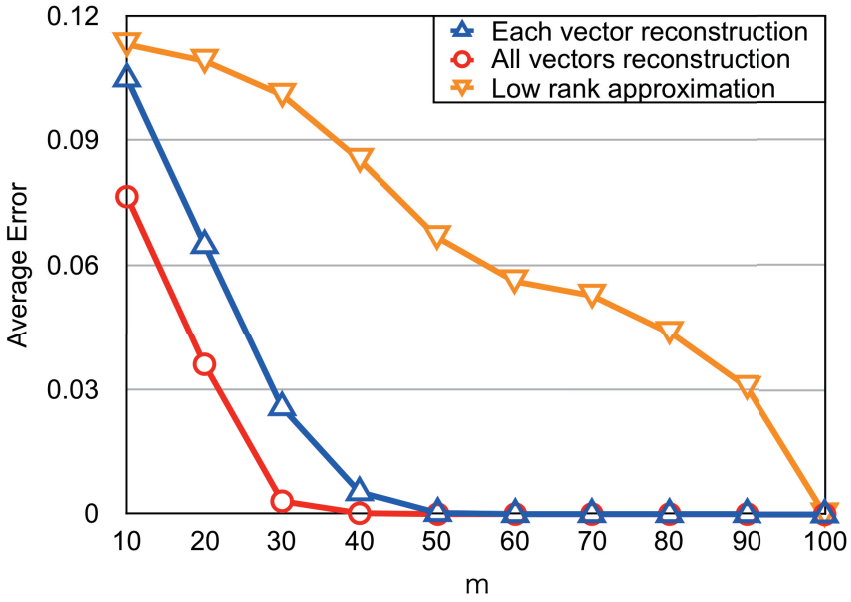

Fig. 5. Average error of all-vector estimation.

Figure 5 also shows the average error of reconstruction based on low-rank approximation (14). Here, we use m of the largest eigenvalues for reconstruction. From Fig. 5, the average error of low-rank approximation has larger error than our proposed method. Therefore, we can recognize that the proposed method has better performance than low-rank approximation for Laplacian matrix reconstruction.

Since the dimensionality of (15) is very high for large-scale networks, it causes the problem in computational cost. However, we can expect that the sparsity of the Laplacian matrix is valid for actual social networks. The experimental network model generated by BA model has hub nodes whose links are not sparse, and we need the aforementioned method to recover the sparsity. On the other hand, links of hub nodes in actual social networks is sparse because even hub nodes have links very smaller than the number of users. Therefore, the situation that we need the method of (15) is rare case.

5. Conclusion

In this paper, we have proposed a method for reproducing a sparse Laplacian matrix from just some of its eigenvalues and eigenvectors by using compressed sensing. This method can be used to investigate the structure of social networks from less than complete information. The performance of this method was verified by experiments. The results showed that to improve the estimation accuracy, it is necessary to use both large eigenvalues and small eigenvalues. In situations where the eigenvalues and eigenvectors contain noise, the estimated matrix may contain relatively large errors even if many eigenvalues are used. Since the Laplacian matrix of social networks has high sparsity, the estimation

accuracy is improved by reconstructing all the row vectors of the Laplacian matrix simultaneously, even if some nodes have high nodal degree. In addition, the proposed method is superior to low-rank approximation in terms of accuracy.

In future research, we will evaluate the performance of the proposed method by using actual data of eigenvalues and eigenvectors observed by the network resonance method.

Acknowledgments

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

Appendix

A. Network resonance method

Let $\boldsymbol{x}(t) = {}^t(\{x_i(t)\})$ be the vector representing the state of nodes at time t. When periodic external force F with frequency ω is imposed on node j, the motion equation of the oscillation forced on the network is given by

$$\frac{\mathrm{d}^2\boldsymbol{x}(t)}{\mathrm{d}t^2} + \gamma\,\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} + \boldsymbol{L}\,\boldsymbol{x}(t) = (F\,\cos\omega t)\,\mathbf{1}_{\{j\}}, \tag{A-1}$$

where γ is a constant, and $\mathbf{1}_{\{j\}}$ is a column vector such that only the j-th component is 1 and all others are 0. Steady state solution $\boldsymbol{x}(t)$ is obtained as

$$\boldsymbol{x}(t) = \sum_{\mu=0}^{n-1} A_\mu(\omega)\cos\big(\omega t + \theta_\mu(\omega)\big)\boldsymbol{v}_\mu, \tag{A-2}$$

where $A_\mu(\omega)$ and $\theta_\mu(\omega)$ are the amplitude and phase of mode μ, respectively. They can be written as

$$A_\mu(\omega) = \frac{F\,v_\mu(j)}{\sqrt{(\omega_\mu^2-\omega^2)^2+(\gamma\omega)^2}}, \tag{A-3}$$

$$\theta_\mu(\omega) = \tan^{-1}\left(\frac{\gamma\,\omega}{\omega^2-\omega_\mu^2}\right), \tag{A-4}$$

where $\omega_\mu = \sqrt{\lambda_\mu}$. From (A-3), $A_\mu(\omega)$ is maximized when the frequency of the external force, ω, is

$$\omega = \sqrt{\omega_\mu^2 - \gamma^2/2}. \tag{A-5}$$

This phenomenon is called resonance. From the frequency at which resonance occurs, we can estimate eigenfrequency ω_μ. If eigenfrequency ω_μ is given, we can estimate the eigenvalue λ_μ by using $\lambda_\mu = \omega_\mu^2$.

In estimating eigenvectors, we focus on not the position but the height of amplitude peaks. For details, please see [14, 21].

References

[1] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[2] K. Yuta, N. Ono, and Y. Fujiwara, "Structural analysis of human network in social networking services," *Transactions of Information Processing Society of Japan*, vol. 47, no. 3, pp. 865–874, 2005.

[3] M. Uchida and S. Shirayama, "Analysis of network structure and model estimation for SNS," *Transactions of Information Processing Society of Japan*, vol. 47, no. 9, pp. 2840–2849, 2006.

[4] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, pp. 47–97, 2002.

[5] F.R.K. Chung, "Lectures on spectral graph theory," *CBMS Lecture Notes*, AMS Publications, Providence, 1995.

[6] D. Spielman, "Spectral Graph Theory," Chapter 18 of *Combinatorial Scientific Computing*, Chapman and Hall/CRC, pp. 495–524, 2012.

[7] A. Sandryhaila and J.M.F. Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, 2013.

[8] A. Sandryhaila and J.M.F. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Processing Magagine*, vol. 31, no. 5, pp. 80–90, 2014.

[9] 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.

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

[11] C. Takano and M. Aida, "Proposal of new index for describing node centralities based on oscillation dynamics on networks," *IEEE GLOBECOM 2016*, 2016.

[12] C. Takano and M. Aida, "Revealing of the underlying mechanism of different node centralities based on oscillation dynamics on networks," *IEICE Transactions on Communications*, vol. E101-B, no. 8, pp. 1820–1832, August 2018.

[13] S. Furutani, C. Takano, and M. Aida, "Proposal of the network resonance method for estimating eigenvalues of the scaled Laplacian matrix," *The 8th International Conference on Intelligent Networking and Collaborative Systems (INCoS 2016) Workshop (WIND 2016)*, 2016.

[14] S. Furutani, C. Takano, and M. Aida, "Method for estimating eigenvectors of a scaled Laplacian matrix using the resonance of oscillation dynamics on networks," *IEEE/ACM International Conference on Social Networks Analysis and Mining (ASONAM 2017)*, Sydney, Australia, 2017.

[15] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer Publishing Company, Incorporated, 2010.

[16] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[17] E. Candes and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[18] K. Hayashi, M. Nagahara, and T. Tanaka, "A user's guide to compressed sensing for communications systems," *IEICE Transactions on Communications*, vol. E96-B, no. 3, pp. 685–712, 2013.

[19] H. Zheng, F. Yang, X. Tian, and X. Gan, "Data gathering with compressive sensing in wireless sensor networks: A random walk based approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 35–44, 2015.

[20] T. Matsuda, M. Nagahara, and K. Hayashi, "Link quality classifier with compressed sensing based on ℓ_1-ℓ_2 optimization," *IEEE Communications Letters*, vol. 15, no. 10, pp. 1117–1119, 2011.

[21] S. Furutani, C. Takano, and M. Aida, "Network resonance method: Estimating network structure from the resonance of oscillation dynamics," *IEICE Transactions on Communications*, vol. E102-B, no. 4, pp. 799–809, April 2019.