Low-rank Sparse Decomposition of Graph Adjacency Matrices for Extracting Clean Clusters

Taiju Kanada*, Masaki Onuki*, and Yuichi Tanaka*†
* Tokyo University of Agriculture and Technology, Japan
† PRESTO, Japan Science and Technology Agency, Japan
E-mail: kanada4nenn@msp-lab.org, masaki.o@msp-lab.org, ytnk@cc.tuat.ac.jp

***Abstract*—In this paper, we propose a refining method of graphs having clusters. Clean graphs, i.e., those represent relationships between data clearly, are important for various applications. There have been many graph construction or learning methods, however, graphs obtained from the conventional approaches are not specifically designed to yield clean graphs that have dense connections within clusters whereas sparse ones between them. In this paper, we focus on making dense edges denser and sparse edges sparser for refining graphs. In order to make it possible, we propose a low-rank sparse decomposition of an adjacency matrix. We apply the methodology of robust PCA to the adjacency matrix for the decomposition. To obtain a valid adjacency matrix, we further formulate it in a form applicable to ADMM with proper constraints. In the experiments using synthetic data, we validate that the proposed method effectively refines graphs.**

I. Introduction

Representing data via graphs has been very useful in various areas such as machine learning [1]–[4], computer vision [5] and signal processing [6]–[8]. A graph consists of vertices and (weighted) edges. Graphs are often constructed from observed data by connecting a vertex (representing one data point) to the other ones by using simple methods [4], e.g., fully connected, k-nearest neighbor, and ε-neighbor methods.

These simple methods are easy to use, but they would deteriorate the performance of some applications, such as spectral clustering [4], by using graphs. For an appropriate clustering, vertices within a cluster should be connected denser than those in other clusters, however, graphs constructed by the simple methods only represent the connections between vertices. They cannot often reflect the intra- and inter-cluster connections.

Graph learning methods have recently been proposed to construct a good graph from a set of observations [9]–[13]. They construct a graph by assuming the data prior. Typically, a graph is obtained by solving an optimization problem with a data smoothness term that calculates the smoothness of the signal according to the underlying graph [14]. One also uses the assumption regarding the low-rankness of a matrix form of observations [13]. In this method, a graph is reproduced from the data after a low-rank approximation.

Though these existing methods assume the characteristics of data themselves, the assumptions for the graph itself are not sufficiently utilized. As a result, the conventional methods do not explicitly learn clean graphs that represent strong relationships within clusters and weak relationships among them. In addition, the obtained matrices by these methods do not always represent valid variation operators like having negative edge weights.

In this paper, we propose a refining method of graph adjacency matrices that represent graphs with clusters. Refined clean graphs are assumed to have dense connections within clusters whereas sparse ones between them. To construct a clean graph, the adjacency matrix of the graph is decomposed into low-rank and sparse matrices by solving a convex optimization problem [15]–[18] with additional constraints. The derived low-rank matrix is considered as an optimal adjacency matrix representing a clean graph. As we state later, the low-rankness of the adjacency matrix is expected to make the edges between different clusters sparse and ones within each cluster dense. The optimization method is efficiently solved by using the alternating direction method of multipliers (ADMM) [18] in our method.

Our approach uses the assumptions described as follows. When the ideal data are divided into multiple clusters, the graph made from the data should be a clean graph where each cluster forms complete graphs (or dense graphs close to a complete graph) and the graph has few edges connecting other clusters. Therefore, the clean adjacency matrix can be assumed as a low-rank matrix. This is because the rank of the adjacency matrix is ideally equal to the number of clusters if each cluster forms complete graphs and does not have edges between clusters.

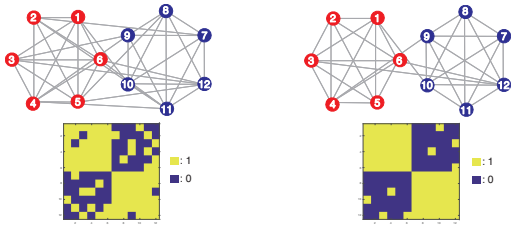
In reality, graphs made from observed data have many noisy edges between clusters. However, compared to the edges in a cluster, the noisy edges are relatively sparse. Therefore, the adjacency matrix representing the noisy edges can be assumed as a sparse matrix. As a result, an adjacency matrix representing clusters could be assumed to be a sum of low-rank and sparse matrices.

The remaining part of this paper is organized as follows. In Section II, the proposed method using the low-rank sparse decomposition is described. Experimental results for synthetic data are shown to validate our proposed method in Section III. Finally, Section IV concludes the paper.

Notation: Notation used in this paper is summarized in Table I. Specifically, $\mathbb{T}$ transposes the matrix of a vector form

TABLE I
LIST OF NOTATION

Symbol	Description
N	The number of samples of vertices
K	The number of clusters
$\mathbf{A} \in \mathbb{R}^{N \times N}$	Adjacency matrix
$\mathbf{W} \in \mathbb{R}^{N \times N}$	Adjacency matrix of a noisy graph
$\mathbf{W}_l \in \mathbb{R}^{N \times N}$	Low-rank component of an adjacency matrix
$\mathbf{W}_s \in \mathbb{R}^{N \times N}$	Sparse component of an adjacency matrix
$\mathbb{T} \in \mathbb{R}^{N^2 \times N^2}$	Transposing matrix for a vector form matrix
$\mathbf{I} \in \mathbb{R}^{N^2 \times N^2}$	Identity matrix
$\mathrm{vec}(\cdot)$	Matrix represented by vector form



(a) $\mathrm{rank}(\mathbf{A}) = 12$ (b) $\mathrm{rank}(\mathbf{A}) = 8$

Fig. 1. Relationship between the number of inter-cluster edges and the rank of adjacency matrices

as $\mathbb{T}\mathrm{vec}(\mathbf{W}) = \mathrm{vec}(\mathbf{W}^T)$ for a square matrix $\mathbf{W}$. It satisfies $\mathbb{T}\mathbb{T}^T = \mathbf{I}$ and is a symmetric matrix. $\mathrm{vec}(\cdot)$ represents a vector form in which columns of a matrix are vertically connected.

II. Proposed Method

A. Motivation: Low-Rank Adjacency Matrix

In order to obtain a clean graph whose dense and sparse parts are clearly separated, it is necessary to reduce the number of edges and/or edge weights connecting different clusters. It is also necessary to make edges within each cluster denser. Such adjacency matrices of the graph are often low-rank.

For example, Fig. 1 shows two graphs along with ranks of their adjacency matrices. Assume that all vertices in the graphs have self-loops with weight 1 (self-loops are not shown in the figure). In this example, there exist two clusters. As can be seen, the smaller the number of edges between clusters is, the lower the rank is. If there is no edge between clusters and each cluster is a complete graph, the rank of the adjacency matrix is 2. Therefore, extracting a low-rank matrix from the noisy adjacency matrix corresponds to a refinement of the graph. Fig. 2 shows an expected low-rank sparse decomposition of the adjacency matrix.

B. Proposed Formulation

We consider to decompose a given adjacency matrix $\mathbf{W}$ for an undirected graph. We utilize the similar methodology to robust PCA (RPCA) [15] for reducing the rank of the adjacency matrix.

Here, the minimization problem is formulated to realize the proposed method. The optimization problem is represented as

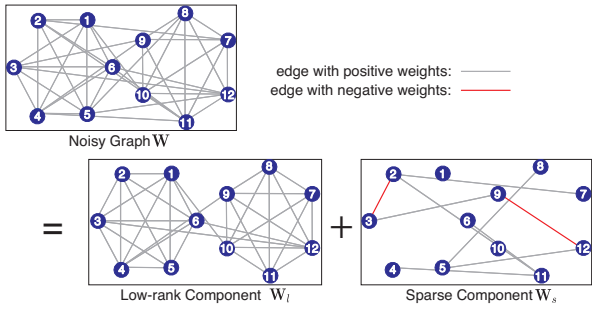


Fig. 2. Expected graph decomposition example applying low-rank sparse decomposition to adjacency matrix.

follows:

$$\underset{\mathbf{W}_l, \mathbf{W}_s}{\arg\min} \ \|\mathbf{W}_l\|_* + \lambda \|\mathbf{W}_s\|_1 \quad \text{s.t.} \begin{cases} \mathbf{W} = \mathbf{W}_l + \mathbf{W}_s \\ \mathbf{W} = \mathbf{W}_l^T + \mathbf{W}_s^T \\ w_{l,ij} \geq 0 \end{cases} \tag{1}$$

where $\mathbf{W}_l$ is the adjacency matrix of the low-rank component, i.e., the refined graph, $\mathbf{W}_s$ is the adjacency matrix of the sparse component, and $w_{l,ij}$ is the (i,j)-element of $\mathbf{W}_l$.

The formulation is basically similar to the RPCA framework [15], but we have two additional constraints. One is a symmetry constraint $\mathbf{W} = \mathbf{W}_l^T + \mathbf{W}_s^T$, that yields undirected edges. If the input matrix is symmetric, the symmetry constraint is not necessary. However, this constraint is added to make it a robust formulation that can output a symmetric matrix even if $\mathbf{W}$ is not symmetric. The other is the nonnegativity constraint $w_{l,ij} \geq 0$, which prohibits edges with negative weights in the refined graph.

C. Algorithm for Solving the Formulation

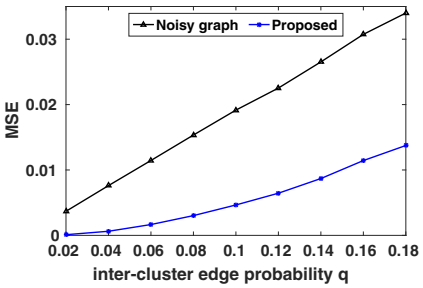
We use ADMM [18] to solve (1). It solves a class of convex optimization problems containing a non-differentiable convex function by using proximal mapping. It is easy to implement, converges quickly, and is widely used in the field of signal and image processing [18]. The objective function (1) is composed of the nuclear norm and the ℓ_1 norm, both of which are convex functions that cannot be differentiated but their proximal mapping can be calculated: That is, ADMM can be applied to.

In order to apply ADMM to the minimization problem (1), it is necessary to reformulate it into an applicable form. In general, the optimization problem that can be solved by ADMM is represented in the following form:

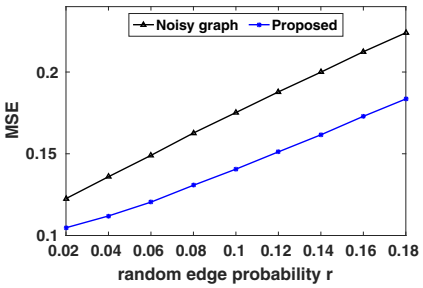
$$\underset{\mathbf{t}, \mathbf{z}}{\arg\min} \ f(\mathbf{t}) + g(\mathbf{z}) \quad \text{s.t. } \mathbf{z} = \mathbf{G}\mathbf{t}, \tag{2}$$

where $\mathbf{t}$ and $\mathbf{z}$ are variable vectors, $\mathbf{G}$ is a column full rank matrix, $f(\mathbf{t})$ is a quadratic function, and $g(\mathbf{z})$ is a function that the proximal mapping can be calculated.

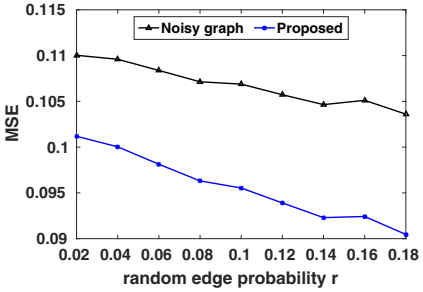
Here we transform (1) into the form of (2). First, there is no quadratic functions in (1), so $f(\mathbf{t}) = 0$. Next, we add



(a) stochastic block graph



(b) community graph-A



(c) community graph-B

Fig. 3. Comparison of refined and noisy graphs by MSE.

constraint terms to the objective function as follows:

$$\begin{aligned}\arg\min_{\mathbf{W}_l,\mathbf{W}_s} \; & \|\mathbf{W}_l\|_* + \lambda\|\mathbf{W}_s\|_1 + \iota_{\mathbf{D}_1}(\mathbf{W}_l + \mathbf{W}_s) \\ & + \iota_{\mathbf{D}_1}(\mathbf{W}_l^T + \mathbf{W}_s^T) + \iota_{\mathbf{D}_2}(\mathbf{W}_l)\end{aligned} \tag{3}$$

where

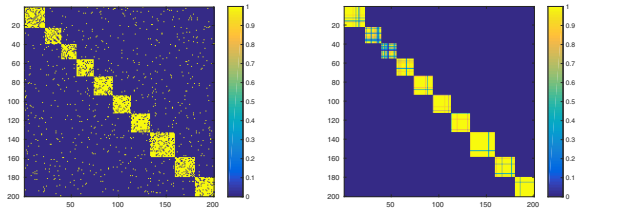
$$\iota_{\mathbf{D}_1}(\mathbf{X}) = \begin{cases} 0 & \text{if } \mathbf{X} \in \mathbf{D}_1 \\ \infty & \text{otherwise} \end{cases}$$

$$\iota_{\mathbf{D}_2}(\mathbf{X}) = \begin{cases} 0 & \text{if } \mathbf{X} \in \mathbf{D}_2 \\ \infty & \text{otherwise} \end{cases}$$

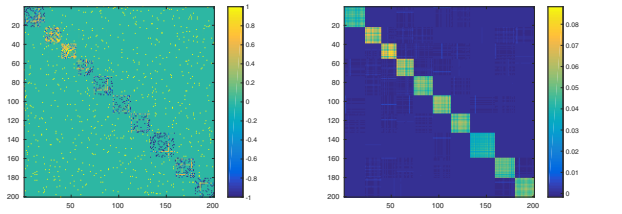
$$\mathbf{D}_1 := \{\mathbf{X} \in \mathbf{R}^{N\times N} |\; \mathbf{X} = \mathbf{W}\}$$

$$\mathbf{D}_2 := \{\mathbf{X} \in \mathbf{R}^{N\times N} |\; \forall i,j,\; \mathbf{X}_{ij} \geq 0\}.$$

Then, $\mathbf{W}_l$ and $\mathbf{W}_s$ are transformed into vector forms $\text{vec}(\mathbf{W}_l)$ and $\text{vec}(\mathbf{W}_s)$, respectively, and variable vectors $\mathbf{z} = [\mathbf{z}_1^T, \mathbf{z}_2^T, \ldots, \mathbf{z}_5^T]^T$ are set to $\mathbf{z}_1 = \text{vec}(\mathbf{W}_l)$, $\mathbf{z}_2 = \text{vec}(\mathbf{W}_s)$,

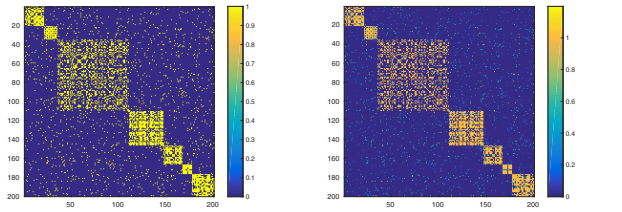


(a) noisy graph ($q = 0.21$) (b) low-rank component (refined graph)

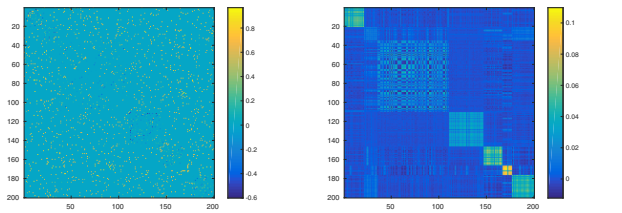


(c) sparse component (d) Output matrix by SSC

Fig. 4. Example of graph adjacency matrix decomposition by proposed method (stochastic block graph).



(a) noisy graph ($q = 0.04$) (b) low-rank component (refined graph)



(c) sparse component (d) Output matrix by SSC

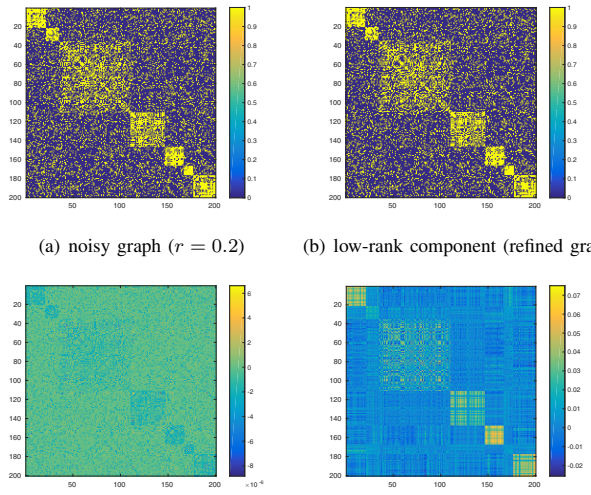
Fig. 5. Example of graph adjacency matrix decomposition by proposed method (community graph-A).

$\mathbf{z}_3 = \text{vec}(\mathbf{W}_l + \mathbf{W}_s)$, $\mathbf{z}_4 = \text{vec}(\mathbf{W}_l^T + \mathbf{W}_s^T)$, $\mathbf{z}_5 = \text{vec}(\mathbf{W}_l)$. Here, (3) is represented as follows:

$$\begin{aligned}\arg\min_{\mathbf{W}_l,\mathbf{W}_s} \; & \|\text{vec}^{-1}(\mathbf{z}_1)\|_* + \lambda\|\mathbf{z}_2\|_1 + \iota_{\mathbf{D}_1}(\mathbf{z}_3) + \iota_{\mathbf{D}_1}(\mathbf{z}_4) + \iota_{\mathbf{D}_2}(\mathbf{z}_5) \\ \text{s.t. } & \mathbf{z} = \mathbf{G}\mathbf{t},\end{aligned}$$

where

$$\mathbf{G} = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{I} \\ \mathbf{I} & \mathbf{I} \\ \mathbb{T} & \mathbb{T} \\ \mathbf{I} & 0 \end{bmatrix}, \; \mathbf{t} = \begin{bmatrix} \text{vec}(\mathbf{W}_l) \\ \text{vec}(\mathbf{W}_s) \end{bmatrix}.$$



(a) noisy graph ($r = 0.2$) (b) low-rank component (refined graph)

(c) sparse component (d) Output matrix by SSC

Fig. 6. Example of graph adjacency matrix decomposition by proposed method (community graph-A).

It is a form that can be solved by ADMM.

The following variables are iteratively calculated until $\mathbf{t}$ converges.

$$\mathbf{t}^{(n+1)} = \arg\min_{\mathbf{t}} \frac{\rho}{2}\|\mathbf{z}^{(n)} - \mathbf{G}\mathbf{t} - \mathbf{d}^{(n)}\|_2^2$$
$$\mathbf{z}^{(n+1)} = \mathrm{prox}(\mathbf{G}\mathbf{t}^{(n+1)} + \mathbf{d}^{(n)})$$
$$\mathbf{d}^{(n+1)} = \mathbf{d}^{(n)} + \mathbf{G}\mathbf{t}^{(n+1)} - \mathbf{z}^{(n+1)},$$

where the number of iterations is represented by n and $\mathrm{prox}(\cdot)$ is a proximal mapping. Each proximal mapping of $\mathbf{z}_1, \ldots, \mathbf{z}_6$ is shown as follows:

$$\mathbf{z}_1^{(n+1)} = \mathrm{vec}(\mathbf{U}\widehat{\mathbf{\Lambda}}\mathbf{V}^T),$$
$$\widehat{\mathbf{\Lambda}} = \mathrm{diag}(\max(\sigma_1 - \gamma, 0), \ldots, \max(\sigma_N - \gamma, 0)),$$

where $\mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$ is the singular value decomposition of $\mathbf{W}_l^{(n+1)} + \mathbf{d}_1^{(n)}$ and γ is a constant. Then,

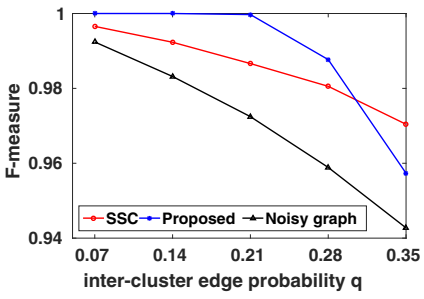
$$z_{2,i}^{(n+1)} = \mathrm{sgn}(x_i) \cdot \max(|x_i| - \gamma, 0), \tag{4}$$

where the ith element of $\mathbf{z}_2^{(n+1)}$ is $z_{2,i}^{(n+1)}$, $\mathbf{x} = \mathrm{vec}(\mathbf{W}_s^{(n+1)} + \mathbf{d}_2^{(n)})$, and $\mathrm{sgn}(\cdot)$ is the signum function. Furthermore, $\mathbf{z}_3^{(n+1)} = \mathbf{z}_4^{(n+1)} = \mathrm{vec}(\mathbf{W})$ and $z_{5,i}^{(n+1)} = \max(y_i, 0)$, where $\mathbf{y} = \mathrm{vec}(\mathbf{W}_s^{(n+1)} + \mathbf{d}_5^{(n)})$.

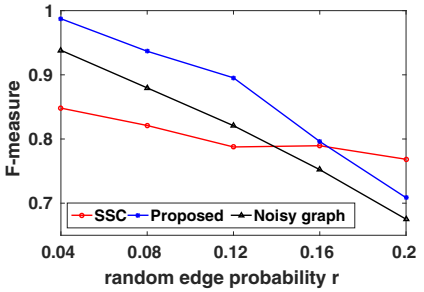
III. Experimental Results

A. Experiment Setup

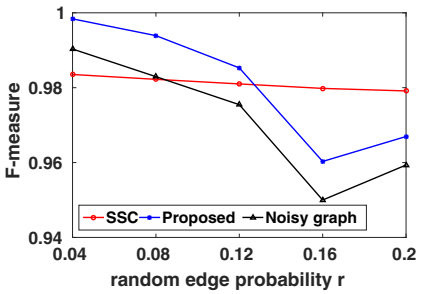
We performed an experiment to restore clean adjacency matrices by refining noisy graphs with the proposed method. Note that the performance of the proposed method depends on λ in (1). In the experiment, the range of λ is set to $[0, 1]$. Since there are random factors in the experiments, we perform 30 iterations.



(a) stochastic block graph



(b) community graph-A



(c) community graph-B

Fig. 7. Comparison of refined and noisy graphs and SSC by the best F-measures.

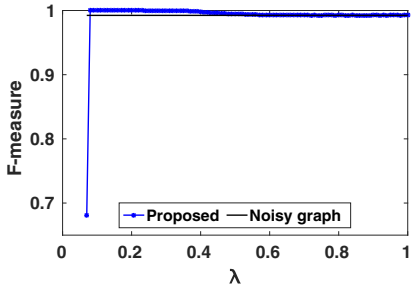
We measure the performance of the proposed method by MSE between the adjacency matrix of the ideal clean graph and that of the refined one. MSE is represented as follows:

$$\mathrm{MSE}(\mathbf{W}_{\mathrm{ideal}}, \mathbf{W}_l) = \frac{1}{N^2}\|\mathbf{W}_{\mathrm{ideal}} - \mathbf{W}_l\|_F^2, \tag{5}$$

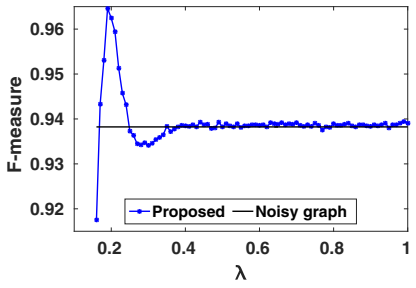
where $\mathbf{W}_{\mathrm{ideal}} \in \mathbb{R}^{N \times N}$ is an adjacency matrix of the ideal clean graph and $\mathbf{W}_l \in \mathbb{R}^{N \times N}$ is that of a refined graph. We define the ideal clean graph as a graph in which each cluster is a complete graph and is disconnected to other clusters.

For experiments, a stochastic block graph and two community graphs are used as input. The stochastic block graph can be constructed with probability $p \leq 1$ and $q = 1 - p$, where p is the probability of the intra-cluster edges and q is the probability of the inter-cluster edges. $K = 10$ and binary edge weights are used.

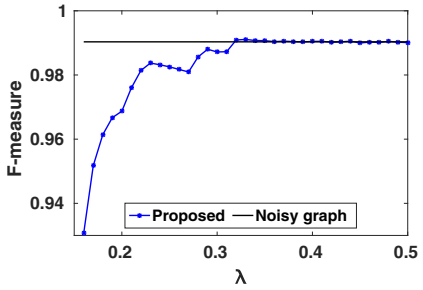
For the community graphs, intra-cluster edges are gen-



(a) stochastic block graph, $q = 0.07$

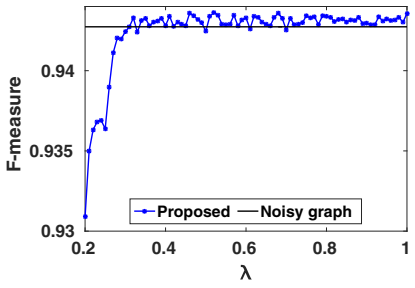


(b) community graph-A, $r = 0.04$

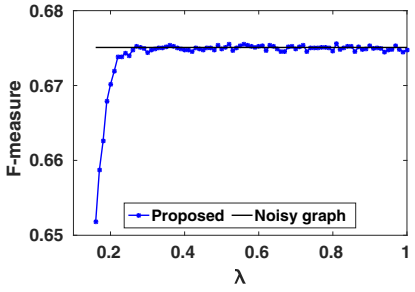


(c) community graph-B, $r = 0.04$

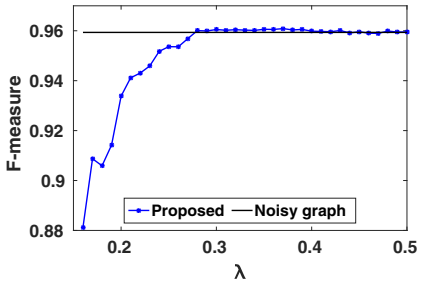
Fig. 8. Comparison of F-measures between the noisy and refined graphs (small noise case).



(a) stochastic block graph, $q = 0.35$



(b) community graph-A, $r = 0.2$



(c) community graph-B, $r = 0.2$

Fig. 9. Comparison of F-measures between the noisy and refined graphs (large noise case).

erated by the ϵ-neighbor method. To generate inter-cluster connections, random edges are added with probability $r \leq 1$ between any pair of nodes (not restricted to inter-clusters). Two community graphs are considered: Community graph-A has binary edge weights, whereas community graph-B has edge weights given by a Gaussian kernel. In the community graph-B, two-dimensional coordinates are given for each vertex, where vertices in each cluster have concentrated coordinates. Therefore, the weighting criterion is based on a Euclidean distance between vertices. In both of the graphs, $K = 7$. These graphs are constructed with GSPBOX [19].

B. Comparisons of MSE and Refined Graphs

Fig. 3 shows the comparisons of MSEs as a function of the probabilities of inter-cluster edge connectivity. The best λ realizing the lowest MSE is used for comparison purpose. As can be seen, refined adjacency matrices present better MSEs than the noisy ones as expected.

Figs. 4, 5, and 6 show the adjacency matrices of the noisy graph, the refined graph (low-rank component), the sparse component, and the output matrix $\mathbf{U}\mathbf{U}^T$ of convex sparse spectral clustering (SSC) (details are shown in Section III-C).

In Fig. 4(b), it is clear that the proposed method makes dense parts denser and sparse parts sparser. It is also observed the proposed method succeeded to reduce edge weights between clusters and increase edge weights in Fig. 5(b). At the same time, the number of edges in the clusters is slightly increased. In contrast, Fig. 6(a) has many noisy edges between clusters: it is difficult to decompose it into low-rank and sparse components. In Figs. 4(d), 5(d), and 6(d), SSC seems to emphasize intra-cluster components regardless of the inter-cluster or random edge probabilities q and r. However, the

inter-cluster edges remain as shown in Fig. 4(d), even when q is not very large.

C. Application to Spectral Clustering

We also perform an experiment of spectral clustering, which is one of the possible applications of our method. Spectral clustering uses eigenvectors of a graph Laplacian that can be calculated from an adjacency matrix. The performance of the proposed method is compared with SSC [20].

SSC solves the minimization problem as follows:

$$\underset{\mathbf{U}\in\mathbb{R}^{n\times k}}{\arg\min}\ \langle \mathbf{U}\mathbf{U}^T, \mathbf{L}\rangle_F + \beta\|\mathbf{U}\mathbf{U}^T\|_0 \ \text{ s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I}. \quad (6)$$

where $\mathbf{L}$ is the normalized graph Laplacian that represents the input noisy graph, k is the number of clusters, β is a hyperparameter, and $\langle\cdot,\cdot\rangle_F$ is Frobenius inner product. SSC needs to know k a priori. After solving (6), a sparse matrix containing k eigenvectors of $\mathbf{L}$ is obtained as $\mathbf{U}$. In the experiment, the range of β is experimentally determined within the interval $[10^{-5}, 10^{-3}]$.

We use F-measure as a clustering evaluation criterion. It has a value (0, 1], and a value closer to 1 indicates a better clustering result. F-measure of spectral clustering using the ideal clean graph naturally becomes 1.

The results are shown in Fig. 7. The vertical axis is F-measure and the horizontal axis represents inter-cluster or random edge probability q or r. As seen in the figure, the proposed method improves performance of spectral clustering compared to the noisy graphs. SSC is robust against edge probability q or r, but its performance improvement is often slight for small q and r.

Figs. 8 and 9 show the behavior of the proposed method according to λ. Fig. 8 corresponds to the graphs with fewer inter-cluster edges, whereas Fig. 9 is for the case that different clusters are connected by relatively dense edges. Since the output adjacency matrix of the proposed method becomes close to noisy graph as λ becomes large, the performance of the proposed method also converges to that of the noisy graph as λ becomes large.

In Figs. 8(a)and 8(b), the noisy graphs are made with small q and r, i.e. they are close to the ideal graph. The performance improvements are observed around $\lambda = 0.2$. Since the proposed method succeeds to extract the intra-cluster edges as a low-rank component, this causes the performance improvements.

In contrast, Figs. 9(a)and 9(b) are for the case with relatively large q and r. In this case, the density of the inter-cluster edges are high as well as that of the intra-cluster ones. That means it is difficult to extract intra-cluster edges as a low-rank component. Therefore, the performance improvements are slight.

Figs. 8(c) and 9(c) are the results for weighted graphs. Similar to the dense edge case, the improvements are not very significant. This is because the low-rankness of adjacency matrices cannot be extracted well in the proposed formulation. That is, modifications of the formulation are needed for the weighted graphs.

IV. Conclusions

In this paper, we propose a graph refining method using a low-rank sparse decomposition of adjacency matrices representing graphs. In the proposed method, we applied a low-rank sparse decomposition based on the formulation of RPCA with additional constraints to obtain valid adjacency matrices for undirected graphs. The optimization problem is convex and is solved by using ADMM. In the experiment, the proposed method cleans up the noisy graphs and the clean graphs have expected characteristics and improves spectral clustering performances. The proposed method can be applied without knowing the number of clusters of data beforehand. Although this paper performed experiments only with synthetic data, it will be possible to improve the performance of some applications such as community detection using a graph estimated from real data.

References

[1] C. Bishop, *Pattern Recognition and Machine Learning (Inforation Science and Science and Statistics)*. Springer, 2007.

[2] N. Tremblay, G. Puy, R. Gribonval, and P. Vandergheynst, "Compressive Spectral Clustering," *ArXiv e-prints*, Feb. 2016.

[3] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the nystrom method," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 2, pp. 214–225, 2004.

[4] U. V. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[5] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[6] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[7] Y. Tanaka and A. Sakiyama, "M-channel oversampled graph filter banks," *IEEE Transactions on Signal Processing*, vol. 62, no. 14, pp. 3578–3590, 2014.

[8] M. Onuki, S. Ono, M. Yamagishi, and Y. Tanaka, "Graph signal denoising via trilateral filter on graph spectral domain," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 137–148, 2016.

[9] C. Hu, L. Cheng, J. Sepulcre, G. E. Fakhri, Y. M. Lu, and Q. Li, "A graph theoretical regression model for brain connectivity learning of Alzheimer's disease," in *Proc. IEEE Int. Symp. Biomedical Imaging (ISBI)*, 2013.

[10] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Laplacian matrix learning for smooth graph signal representation," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 3736–3740.

[11] V. Kalofolias, "How to learn a graph from smooth signals," in *Proc. Journal of Machine Learning Research (JMLR)*, 2016, pp. 920–929.

[12] E. Pavez and A. Ortega, "Generalized Laplacian precision matrix estimation for graph signal processing," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2016, pp. 6350–6354.

[13] L. Rui, H. Nejati, S. H. Safavi, and N.-M. Cheung, "Simultaneous low-rank component and graph estimation for high-dimensional graph signals: Application to brain imaging," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2017, pp. 4134–4138.

[14] V. Kalofolias, "How to learn a graph from smooth signals," in *19th International Conference on Artificial Intelligence and Statistics*, 2016, pp. 920–929.

[15] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the Association for Computing Machinery*, vol. 58, no. 3, p. 11, 2011.

[16] X. Zhou, C. Yang, H. Zhao, and W. Yu, "Low-rank modeling and its applications in image analysis," *ACM Computing Surveys (CSUR)*, vol. 47, no. 2, p. 36, 2015.

[17] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2233–2246, 2012.

[18] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[19] N. Perraudin, J. Paratte, D. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, "Gspbox: A toolbox for signal processing on graphs," *arXiv preprint arXiv:1408.5781*, 2014.

[20] C. Lu, S. Yan, and Z. Lin, "Convex Sparse Spectral Clustering: Single-View to Multi-View," *IEEE Transactions on Image Processing*, vol. 25, pp. 2833–2843, Jun. 2016.