Learning Graphs with Multiple Temporal Resolutions

Koki Yamada* and Yuichi Tanaka*†
* Graduate School of BASE, Tokyo University of Agriculture and Technology, Tokyo, Japan
† PRESTO, Japan Science and Technology Agency, Saitama, Japan
E-mail: k-yamada@msp-lab.org, ytnk@cc.tuat.ac.jp

***Abstract*—This paper proposes a framework for learning time-varying graphs with multiple temporal resolutions from multivariate time series signals. Our method estimates multiresolution graphs by a top-down approach: Graphs are learned from a segment of the time-series data corresponding to the desired temporal resolution, and we impose a constraint so that the learned graphs at the target temporal resolution are close to that in the lower temporal resolution. The proposed approach overcomes the problem of existing time-varying graph learning methods that must infer graphs in a single temporal resolution. Experimental results with synthetic data demonstrate that our method outperforms a baseline graph learning method.**

***Index Terms*—Graph learning, network topology inference, multiresolution graph, time-varying graph.**

I. Introduction

Many practical applications on data science and mining have to handle massive sensor data. Since these sensors are often distributed nonuniformly in a physical space, analyzing them by taking into account their underlying structure, i.e., graph, will improve the qualities and efficiencies of data analysis drastically. *Graph signal processing* (GSP) [1], [2] is a useful tool to handle such data on graphs, and many applications of GSP have been found [3]–[5]. However, in many problems, graphs are not given a priori. Therefore, *graph learning* [6], [7], techniques and algorithms for estimating a graph from observed data and/or feature values, is required in various applications especially for sensor measurements.

Time-varying graph learning, one of the problem settings in graph learning, aims to infer a set of graphs where its element (i.e., one graph) corresponds to a graph in a specific time slot [8], [9]. In contrast to static graph learning approaches [10]–[12], the time-varying methods assume that the relationship among entities could be changed over time while some global temporal structures are shared during measurements. In [8], [9], time-varying graphs are inferred by imposing constraints for temporal variations of graphs between neighboring temporal windows. In these methods, the temporal window size—temporal resolution—has to be fixed. This means that, even for time-varying graph learning, a set of graphs with one temporal resolution can only be obtained.

In practice, we encounter the cases where data and their underlying graphs present different behaviors in different temporal resolutions. For example, temperatures have hourly, daily, monthly, and even yearly behaviors that correspond to different temporal resolutions. These temperatures are also related spatially where the spatial relationship is represented as a graph. Moreover, graphs in one temporal resolution will be affected by those in different temporal resolutions. Unfortunately, the existing time-varying graph learning methods cannot extract such behaviors because they ignore the relationship among different temporal resolutions.

To overcome the above-mentioned problems, we propose a framework for learning a set of time-varying graphs, i.e., a set of sets of graphs, with multiple temporal resolutions. The proposed method is realized by a top-down approach where it learns graphs from corresponding segments of time series data such that each of the graphs is close to the graph at the lower temporal resolution. We formulate a graph learning problem at each temporal segment of data, that can be done by extending a static graph learning method based on graph smoothness [11], [12]. At the same time, we can incorporate a constraint on the information of the graph in a lower resolution with our formulation. The graph learning problem can be convex and solved efficiently by a primal-dual splitting algorithm [13].

The main contributions of this paper are summarized as follows:

- This study is the first attempt to infer time-varying graphs with multiple temporal resolutions.
- Our framework enables us to infer graphs at an arbitrary temporal resolution from multivariate time series signals.
- We can prevent from missing global structures, i.e., graphs in a low temporal resolution, at even a high temporal resolution.

Notation: The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ is a weighted graph with N vertices in the set $\mathcal{V}$ and edges in the set $\mathcal{E}$, where the number of nodes and edges are denoted by $N = |\mathcal{V}|$ and $E = |\mathcal{E}|$. The matrix $\mathbf{W}$ denotes a weighted adjacency matrix. In this paper, we assume a graph is undirected with nonnegative edges and does not have self-loops, i.e., $\mathbf{W}$ is symmetric with nonnegative elements and its all diagonal elements are zero. Graph Laplacian is given by $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is the degree matrix defined as $D_{ii} = \sum_j W_{ij}$. The graph Fourier transform (GFT) and the inverse GFT are defined by $\hat{\mathbf{x}} = \mathbf{U}^\top \mathbf{x}$ and $\mathbf{x} = \mathbf{U}\hat{\mathbf{x}}$, where $\mathbf{U}$ is the eighenvector matrix of $\mathbf{L}$

The lowercase and uppercase bold letters represent a vector and matrix, and a calligraphic letter represents a set. We use the following definitions of symbols: $\mathbf{1} = [1, \ldots, 1]^\top$, $\circ$ is Hadamard product of two matrices, and $\lfloor \cdot \rfloor$ represents the floor

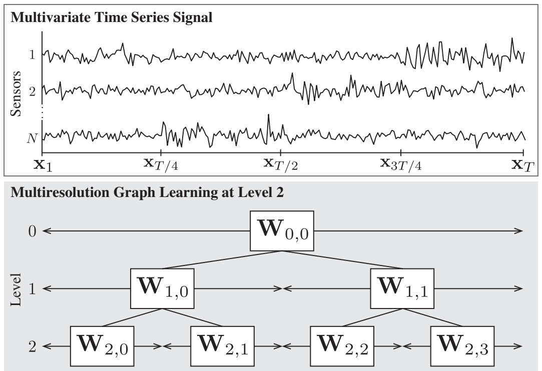


Fig. 1. The overview of multiresolution graph learning.

function.

II. Graph Learning Based on Signal Smoothness

Suppose the following signal generation model of graph signals [12]: $\mathbf{x} = \mathbf{U}\mathbf{h} + \boldsymbol{\epsilon}$, where $\mathbf{x} \in \mathbb{R}^N$ is an observed signal, $\mathbf{h} \in \mathbb{R}^N$ is a latent variable represented in the graph frequency domain, and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_\epsilon^2 \mathbf{I})$ is an additive white Gaussian noise. The maximum a posteriori estimation of $\mathbf{h}$ with the assumption that $p(\mathbf{h}) = \mathcal{N}(0, \boldsymbol{\Lambda}^\dagger)$ and $p(\mathbf{x}|\mathbf{h}) = \mathcal{N}(\mathbf{U}\mathbf{h}, \sigma_\epsilon^2 \mathbf{I})$ leads to the following optimization problem:

$$\min_{\mathbf{L}, \mathbf{y}} \ \|\mathbf{x} - \mathbf{y}\|_2^2 + \alpha \mathbf{y}^\mathsf{T} \mathbf{L} \mathbf{y}, \tag{1}$$

where $\mathbf{y} = \mathbf{U}\mathbf{h}$. When the observed signal is noise-free, the graph is learned by minimizing $\mathbf{x}^\mathsf{T}\mathbf{L}\mathbf{x}$, which measures the smoothness of signals on graphs. While the problem in (1) has been originally solved by an alternative approach, Kalofolias [11] also uses this graph smoothness measure to formulate a graph learning problem as the following convex optimization problem:

$$\min_{\mathbf{W} \in \mathcal{W}_m} \ \|\mathbf{W} \circ \mathbf{Z}\|_1 - \alpha \mathbf{1}^\mathsf{T} \log(\mathbf{W}\mathbf{1}) + \beta ||\mathbf{W}||_F^2, \tag{2}$$

where α and β are parameters, and $\mathcal{W}_m$ is a valid set of weighted adjacency matrices defined by

$$\mathcal{W}_m = \left\{ \mathbf{W} \in \mathbb{R}_+^{N \times N} \mid \mathbf{W} = \mathbf{W}^\mathsf{T}, W_{ii} = 0 \right\}. \tag{3}$$

The first term in (2) corresponds to the smoothness of K observed signals $\{\mathbf{x}_k\}_{k=1}^K$ given by $\|\mathbf{W} \circ \mathbf{Z}\|_1, = \frac{1}{2} \sum_{k=1}^K \mathbf{x}_k^\mathsf{T} \mathbf{L} \mathbf{x}_k$ where $\mathbf{Z}$ is the pairwise distance matrix defined as $Z_{ij} = \sum_{k=1}^K \|(\mathbf{x}_k)_i - (\mathbf{x}_k)_j\|^2$. The second term of (2) forces the degree on each vertex to be positive without preventing edge weights from becoming zero. The third term controls the sparseness of the learned weighted adjacency matrix.

III. PROPOSED METHOD

A. Problem Formulation

We consider a problem of estimating multiresolution time-varying graphs from a multivariate time series signal $\mathbf{X} \in \mathbb{R}^{N \times T} = [\mathbf{x}_1, \ldots, \mathbf{x}_T]$, where $\mathbf{x}_t \in \mathbb{R}^N$ is the tth measurement of the time series signal[1]. Our framework recursively performs the following two operations:

1) Dividing $\mathbf{X}$ into two equal-sized temporal segments with no overlapping.
2) Learning two weighted adjacency matrices where each of them corresponds to one data segment with a constraint from the "parent" graph, i.e., one at a lower temporal resolution.

These recursive operations result in multiresolution graphs as shown in Fig. 1. Here, the divided segment is denoted by $\mathbf{X}_{l,m} \in \mathbb{R}^{N \times T/2^l}$, where l and m represent the temporal resolution level and the segment index at the lth level, respectively. Additionally, the weighted adjacency matrix corresponding to $\mathbf{X}_{l,m}$ is denoted by $\mathbf{W}_{l,m} \in \mathbb{R}^{N \times N}$.

To formulate the multiresolution graph learning problem in our framework, we extend the static graph learning problem of (2) with a constraint that promotes to inherit a part of the connectivities at the lower temporal resolution. Consequently, the graph learning problem is formulated by the following convex optimization problem:

$$\begin{aligned} \min_{\mathbf{W}_{l,m} \in \mathcal{W}_m} \ & \|\mathbf{W}_{l,m} \circ \mathbf{Z}_{l,m}\|_1 - \alpha \mathbf{1}^\mathsf{T} \log(\mathbf{W}_{l,m}\mathbf{1}) \\ & + \beta \|\mathbf{W}_{l,m}\|_F^2 + \eta \|\mathbf{W}_{l,m} - \mathbf{W}_{l-1,\lfloor m/2 \rfloor}\|_1, \end{aligned} \tag{4}$$

where $\mathbf{W}_{l-1,\lfloor m/2 \rfloor}$ is the adjacency matrix at the lower resolution that corresponds to the parent of $\mathbf{W}_{l,m}$ (please refer to Fig. 1) and $\mathbf{Z}_{l,m}$ is the pairwise distance matrix of $\mathbf{X}_{l,m}$. Particularly, the fourth term of (4) promotes the sparseness of the difference between $\mathbf{W}_{l,m}$ and $\mathbf{W}_{l-1,\lfloor m/2 \rfloor}$. This prevents from missing global structures in graph learning at high temporal resolutions.

The target matrix $\mathbf{W}_{l,m}$ of (4) must be a symmetric matrix with all-zero diagonal elements. This means we need to consider (4) only for the upper (or lower) triangular part of $\mathbf{W}_{l,m}$. Hence, we can rewrite (4) in a vector form which only contains the upper triangular parts of the matrices. Let $\mathbf{w}$, $\mathbf{z}$, and $\mathbf{c}$ be the vector forms of $\mathbf{W}_{l,m}$, $\mathbf{Z}_{l,m}$, and $\mathbf{W}_{l-1,\lfloor m/2 \rfloor}$, respectively, and also let $\mathbf{S}$ be a linear operator satisfying $\mathbf{S}\mathbf{w} = \mathbf{W}_{l,m}\mathbf{1}$. Then, the vector form of the optimization problem (4) is rewritten as

$$\min_{\mathbf{w} \in \mathcal{W}_v} \ 2\mathbf{z}^\mathsf{T}\mathbf{w} - \alpha \mathbf{1}^\mathsf{T} \log(\mathbf{S}\mathbf{w}) + \beta \|\mathbf{w}\|_2^2 + \eta \|\mathbf{w} - \mathbf{c}\|_1, \tag{5}$$

where $\mathcal{W}_v = \left\{ \mathbf{w} \in \mathbb{R}^{N(N-1)/2} \mid w_i \geq 0 \ (i = 1, 2, \ldots) \right\}$. This set is equivalent to the nonnegative constraint. By converting the original problem into the vector form, we can eliminate the symmetric and diagonal constraints of (3).

B. Optimization

The optimization problem in (5) can be solved using a primal-dual splitting (PDS) algorithm. We can further rewrite (5) by introducing the indicator function as follows:

$$\min_{\mathbf{w}} \ 2\mathbf{z}^\mathsf{T}\mathbf{w} - \alpha \mathbf{1}^\mathsf{T} \log(\mathbf{S}\mathbf{w}) + \beta \|\mathbf{w}\|_2^2 + \eta \|\mathbf{w} - \mathbf{c}\|_1 + \iota_{\mathcal{W}_v}(\mathbf{w}), \tag{6}$$

where $\iota_{\mathcal{W}_v}$ is the indicator function of $\mathcal{W}_v$. By introducing a dual variable $\mathbf{v}$, we can convert the optimization problem of

[1] Indeed, we can use the same method if multiple $\mathbf{X}$'s can be observed. In this paper, for brevity, we assume we have only one time series data $\mathbf{X}$.

(6) into the applicable form of PDS algorithm as follows:

$$\begin{aligned} f_1(\mathbf{w}) &= \beta\|\mathbf{w}\|_2^2 \quad \text{with } \xi = 2\beta, \\ f_2(\mathbf{w}) &= 2\mathbf{z}^\mathsf{T}\mathbf{w} + \iota_{\mathcal{W}_v}(\mathbf{w}), \\ f_3(\mathbf{v}) &= f_{3,1}(\mathbf{v}_1) + f_{3,2}(\mathbf{v}_2), \end{aligned} \tag{7}$$

where $\mathbf{v} := \mathbf{A}\mathbf{w} = [\mathbf{v}_1^\mathsf{T}\ \mathbf{v}_2^\mathsf{T}]^\mathsf{T}$, $\mathbf{A} = [\mathbf{S}^\mathsf{T}\ \mathbf{I}]^\mathsf{T}$, $f_{3,1}(\mathbf{v}_1) = -\alpha\mathbf{1}^\mathsf{T}\log(\mathbf{v}_1)$, and $f_{3,2}(\mathbf{v}_2) = \eta\|\mathbf{v}_2 - \mathbf{c}\|_1$. The proximal operators for f_2, $f_{3,1}$, and $f_{3,2}$ can be calculated as follows:

$$\left(\mathrm{prox}_{\gamma f_2}(\mathbf{x})\right)_i = \begin{cases} 0 & x_i \le 2\gamma z_i \\ x_i - 2\gamma z_i & \text{otherwise,} \end{cases} \tag{8}$$

$$\left(\mathrm{prox}_{\gamma f_{3,1}}(\mathbf{x})\right)_i = \frac{x_i + \sqrt{x_i^2 + 4\alpha\gamma}}{2}, \tag{9}$$

$$\left(\mathrm{prox}_{\gamma f_{3,2}}(\mathbf{x})\right)_i = \begin{cases} c_i & |x_i - c_i| \le \gamma\eta \\ \mathrm{sgn}(x_i)(|x_i| - \gamma\eta) & \text{otherwise.} \end{cases} \tag{10}$$

Consequently, the primal-dual splitting algorithm for solving (6) is given by the following iteration:

$$\left\lfloor \begin{aligned} &\mathbf{w}^{(n+1)} := \mathrm{prox}_{\gamma_1 f_2}(\mathbf{w}^{(n)} - \gamma_1(2\beta\mathbf{w} + \mathbf{S}^\mathsf{T}\mathbf{v}_1^{(n)} + \mathbf{v}_2^{(n)})), \\ &\quad \mathbf{v}_1^{(n)} \leftarrow \mathbf{v}_1^{(n)} + \gamma_2\mathbf{S}(2\mathbf{w}^{(n+1)} - \mathbf{w}^{(n)}), \\ &\quad \mathbf{v}_2^{(n)} \leftarrow \mathbf{v}_2^{(n)} + \gamma_2(2\mathbf{w}^{(n+1)} - \mathbf{w}^{(n)}), \\ &\mathbf{v}_1^{(n+1)} := \mathbf{v}_1^{(n)} - \gamma_2 \mathrm{prox}_{\frac{1}{\gamma_2}f_{3,1}}\Big(\frac{\mathbf{v}_1^{(n)}}{\gamma_2}\Big), \\ &\mathbf{v}_2^{(n+1)} := \mathbf{v}_2^{(n)} - \gamma_2 \mathrm{prox}_{\frac{1}{\gamma_2}f_{3,2}}\Big(\frac{\mathbf{v}_2^{(n)}}{\gamma_2}\Big). \end{aligned} \right. \tag{11}$$

We summarize the entire algorithm for the multiresolution graph learning in Algorithm 1 shown below.

Algorithm 1 Multiresolution graph learning at level L

Input: $\mathbf{X}$, L
Output: $\mathbf{W}_{l,m}$ $(l = 0, \dots, L,\ m = 0, \dots 2^l - 1)$

```
Compute Z_{0,0} from X
Solve (2) with Z_{0,0} to learn W_{0,0}
for l = 1, ..., L do
    Divide X into 2^l data segments X_{l,0} ... X_{l,2^l-1}
    for m = 0, ..., 2^l - 1 do
        Compute Z_{l,m} from X_{l,m}
        Solve (4) with Z_{l,m} and W_{l-1,⌊m/2⌋} to learn W_{l,m}
    end for
end for
```

IV. Experimental Results

In this section, we perform experiments with synthetic data to validate our multiresolution graph learning approach.

A. Synthetic Dataset

First, we construct a set of time-varying multiresolution graphs with four levels ($l = 0, \dots, 3$) as shown in Fig. 2. The number of vertices N is set to $N = 81$ and edge weights between vertices are random values drawn from a uniform distribution from the interval $[0.3, 1]$.

The lowest resolution graph, i.e., the graph reflecting the global structure, is $\mathbf{W}_{0,0}$ as shown in Fig. 2(a), where the graph has a grid-like structure while the edges only run vertically, except for the horizontal edges at the middle of the grid. In the next resolution level ($l = 1$), horizontal edges have been added as shown in Figs. 2(b) and (c). The common edges in $\mathbf{W}_{0,0}$ and $\mathbf{W}_{1,i}$ are set to have the same weights (and also common edge weights are shared in parent and children graphs for all resolution levels). From $l = 1$ to 2, diagonal edges from top right to bottom left have been added as shown in Figs. 2(d)–(g). Finally, diagonal edges from top left to bottom right have been appended in the highest resolution graphs. In this way, we can construct a set of multiresolution graphs.

From this set of prototype graphs, we then construct time-varying graphs $\{\mathbf{W}_1, \dots, \mathbf{W}_T\}$. We set $T = 240$ in this experiment. Since the number of mutiresolution graphs in the highest resolution is eight, each of them has been duplicated 30 times and then they are concatenated, i.e., $\mathbf{W}_t := \mathbf{W}_{3,\lfloor (t-1)/30 \rfloor}$ $(t = 1, \dots, 240)$.

Then, a multivariate time series signal $\mathbf{X}$ used in the experiment is generated from $\{\mathbf{W}_1, \dots, \mathbf{W}_T\}$ with the following Gaussian Markov random field model: $\mathcal{N}(\mathbf{x}_{t-1}, (\mathbf{L}_t + \sigma^2\mathbf{I})^{-1})$ where $\mathbf{L}_t$ is the graph Laplacian corresponding to $\mathbf{W}_t$, σ^2 is a variance of i.i.d. white Gaussian noise and set to 0.5. Note that $\mathbf{x}_1$ is generated from $\mathcal{N}(0, (\mathbf{L}_t + \sigma^2\mathbf{I})^{-1})$, where we assume $\mathbf{x}_t = \mathbf{0}$ for $t \le 0$.

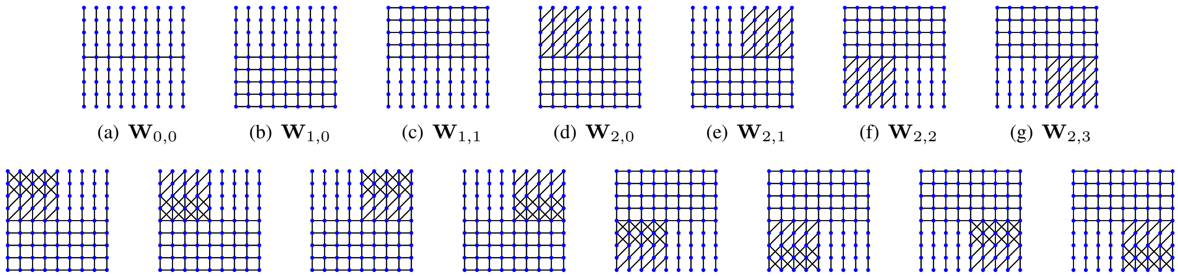
TABLE I
Average graph learning performance at each level.

	F-measure			Relative error		
Level	1	2	3	1	2	3
Baseline	0.814	0.710	0.646	**0.487**	**0.566**	0.661
TVGL [9]	-	-	0.716	-	-	0.638
MRGL	**0.880**	**0.814**	**0.749**	0.544	0.569	**0.619**

B. Performance Comparison

To compare the performance of the proposed method (hereinafter referred to MRGL as an abbreviation of multiresolution graph learning), we use the method that solves (2) instead of (4) in Algorithm 1, i.e., single-resolution graph learning, as a baseline method. Additionally, we also compare the proposed method with a time-varying graph learning (TVGL) method based on the sparseness of temporal variations [9]. The performances of these methods are compared in terms of F-measure and relative error. F-measure reflects the accuracy of structures in learned graphs (whether edges are located in correct positions regardless of their weights) and relative error reflects the accuracy of edge weights.

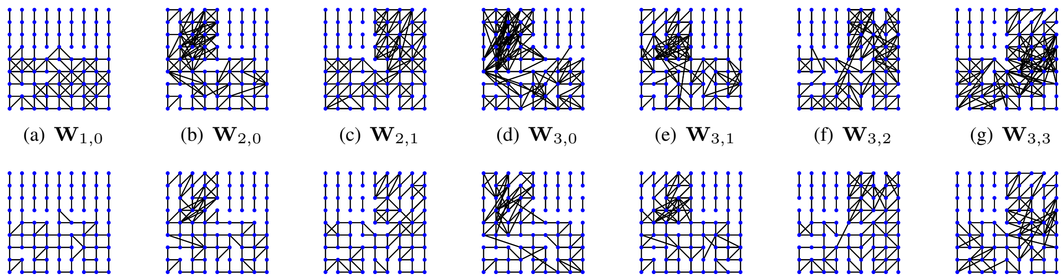
Table I summarizes the average performance of the baseline, TVGL, and MRGL in all levels. Note that we only evaluate the performance of the TVGL in the highest temporal resolution because it is a single-resolution framework. As shown in the table, F-measures of the MRGL outperforms both of the alternative methods consistently in all levels. The relative errors of the baseline and MRGL are comparable, but the baseline is slightly better than the MRGL in the low resolution.



(a) $\mathbf{W}_{0,0}$ (b) $\mathbf{W}_{1,0}$ (c) $\mathbf{W}_{1,1}$ (d) $\mathbf{W}_{2,0}$ (e) $\mathbf{W}_{2,1}$ (f) $\mathbf{W}_{2,2}$ (g) $\mathbf{W}_{2,3}$

(h) $\mathbf{W}_{3,0}$ (i) $\mathbf{W}_{3,1}$ (j) $\mathbf{W}_{3,2}$ (k) $\mathbf{W}_{3,3}$ (l) $\mathbf{W}_{3,4}$ (m) $\mathbf{W}_{3,5}$ (n) $\mathbf{W}_{3,6}$ (o) $\mathbf{W}_{3,7}$

Fig. 2. Visualization of the ground truth graphs.



(a) $\mathbf{W}_{1,0}$ (b) $\mathbf{W}_{2,0}$ (c) $\mathbf{W}_{2,1}$ (d) $\mathbf{W}_{3,0}$ (e) $\mathbf{W}_{3,1}$ (f) $\mathbf{W}_{3,2}$ (g) $\mathbf{W}_{3,3}$

(h) $\mathbf{W}_{1,0}$ (i) $\mathbf{W}_{2,0}$ (j) $\mathbf{W}_{2,1}$ (k) $\mathbf{W}_{3,0}$ (l) $\mathbf{W}_{3,1}$ (m) $\mathbf{W}_{3,2}$ (n) $\mathbf{W}_{3,3}$

Fig. 3. Visualization of learned multiresolution graphs: (Top) graphs learned by the baseline method. (Bottom) graphs learned by the MRGL.

On the other hand, the MRGL outperforms the baseline and TVGL in the highest resolution. This implies that the proposed multiresolution constraint in (4) would improve the performance in a higher resolution.

Fig. 3 visualizes the learned multiresolution graphs while it only shows a part of the graphs due to the limitation of space. Clearly, the graphs learned by our proposed method have a sparser structure than the baseline. Also, thanks to the multiresolution constraint, graphs obtained by the MRGL in higher resolutions inherit the structure in the lower level, while those in the baseline do not have such characteristics as seen in Figs. 3(b) and (d) as well as Figs. 3(c) and (g). This is because the baseline method has to run its graph learning method independently in each temporal resolution.

V. Conclusion

In this paper, we propose the first framework for learning graphs with multiple temporal resolutions from multivariate time series data. The proposed framework utilizes a top-down approach to construct a hierarchical time-varying graphs. The graph learning problem in our framework can be formulated as a convex optimization problem with a constraint such that higher-resolution graphs inherit the structure in the lower resolution. The convex optimization problem can be solved using a primal-dual splitting algorithm. The experimental results demonstrated that our method can estimate edges in correct positions and have a multiresolution property as expected.

References

[1] 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 Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[2] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[3] W. Huang, L. Goldsberry, N. F. Wymbs, S. T. Grafton, D. S. Bassett, and A. Ribeiro, "Graph frequency analysis of brain signals," *IEEE J. Sel. Top. Signal Process.*, vol. 10, no. 7, pp. 1189–1203, 2016.

[4] S. Ono, I. Yamada, and I. Kumazawa, "Total generalized variation for graph signals," in *Proc. IEEE Int. Conf. Acoust. Speech. Signal Process.*, 2015, pp. 5456–5460.

[5] M. Onuki, S. Ono, M. Yamagishi, and Y. Tanaka, "Graph signal denoising via trilateral filter on graph spectral domain," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 137–148, 2016.

[6] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 44–63, 2019.

[7] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro, "Connecting the dots: Identifying network structure via graph signal processing," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 16–43, 2019.

[8] V. Kalofolias, A. Loukas, D. Thanou, and P. Frossard, "Learning time varying graphs," in *Proc. IEEE Int. Conf. Acoust. Speech. Signal Process.*, 2017, pp. 2826–2830.

[9] K. Yamada, Y. Tanaka, and A. Ortega, "Time-varying graph learning based on sparseness of temporal variation," in *Proc. IEEE Int. Conf. Acoust. Speech. Signal Process.*, 2019, pp. 5411–5415.

[10] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under Laplacian and structural constraints," *IEEE J. Sel. Top. Signal Process.*, vol. 11, no. 6, pp. 825–841, 2017.

[11] V. Kalofolias, "How to learn a graph from smooth signals," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2016, pp. 920–929.

[12] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning Laplacian matrix in smooth graph signal representations," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6160–6173, 2016.

[13] L. Condat, "A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms," *J Optim Theory Appl*, vol. 158, no. 2, pp. 460–479, 2013.