Learning Laplacian Matrix for Smooth Signals on Graph*

Tianxing Liao, Wen-Qin Wang*, Bang Huang, Jian Xu

School of Information and Communication Engineering,

University of Electronic Science and Technology of China, Chengdu, 611731, China

*Email address: (wqwang@uestc.edu.cn)

Learning a useful Laplacian matrix plays a significant role in graph learning. This paper focuses on smoothness analysis, which leads to the concept of total variation (TV) on graphs, a new learning Laplacian matrix framework and solving it via convex optimization techniques. We show that smoothness analysis leads to an another form that gives a more sophisticated characterization for smooth signals. This unified theoretical framework can learn a meaningful Laplacian matrix by the constraint of total variation. In our experiments, this framework is demonstrated by using three different data sets.

***Index Terms*—learning Laplacian matrix, smoothness analysis,total variation on graphs, graph signal processing, graph learning.**

I. Introduction

Many applications such as energy, social, transportation, and high-dimensional data often manipulate data on graphs. These graphs consist of vertices and edges. We refer to data supported on vertices as graph signals, where valuable information can be conveyed through signals. Unlike traditional data sets, many structured data are able to construct graph signals. In recent years, many researchers focus on analysis and processing signal in the emerging field of signal processing on graphs, such as spectral analysis [1], vertex-frequency analysis [2], spectral estimation [3] and signal representation [4]. The graphic data structure behind data sets, i.e. Laplacian matrix in this paper, which is considered as priori knowledge, is significant for analyzing and processing signal on graphs. However, a meaningful graph isn't always easily available from signal observations in some applications. Therefore, a graph of high quality is critical for signal processing on graphs and machine learning task.

To address this problem, a machine learning approach for estimating graph structure, which is to estimate the inverse of sparse covariance matrix for Gaussian Markov Random Field (GMRF), was proposed. However, learning Laplacian matrix cannot be interpreted as a graph topology for defining graph signals. Dong et al. in [5], [6] proposed a learning framework for signal on graphs, which infers from smooth graph signals under Gaussian prior. This framework learns Laplacian matrix by minimizing the difference between true signals and observations, Laplacian quadratic form and Frobenius norm of Lapacian matrix. The Laplacian quadratic form is considered as a measure of smooth graph signals [1]. In [5], [6], Dong et al. utilized the information of measure of smooth graph signals and obtained Laplacian matrix by solving their optimization framework. To further improve the graph quality, we use the other information, i.e.total variation on graphs, which derives from the Laplacian quadratic form, and get a better result.

This paper is presented an unified theoretical framework to learn a more meaningful Laplacian matrix when only signal observations are available. We cast the problem of learning graphic structure as optimal Laplacian matrix on graphs by using the information of total variation on graphs. In this framework, we utilize the concept of total variation to construct optimal framework. This concept is inspired by classical digital signal processing and is defined in [7]. However, we derive the total variation on graphs from the concept of smooth graph signals and they have the same form. Furthermore, we obtain a more meaningful Laplacian matrix from the signal observations. Numerical results show that the performance of proposed framework is better than machine learning method and the framework of Dong et al.

II. SMOOTHNESS ANALYSIS

We consider signals on an undirected, connected and weighted graph G with N vertices, and denote the weights of the edges connecting vertices n and m as w_{nm}. The graph Laplacian matrix is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{1}$$

where weighted adjacency matrix $\mathbf{W}$ is consist of weighting coefficients w_{nm} and $\mathbf{D}$ is the diagonal matrix with elements $d_n = \sum\limits_{m=1}^{N} w_{nm}$. To obtain a meaningful structure, the relation between two vertices is quantified by the weighted adjacency matrix w_{nm} . Since $\mathbf{L}$ is a real symmetric matrix, Laplacian matrix $\mathbf{L}$ can be decomposed as

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^* \tag{2}$$

where $\mathbf{U}$ is the eigenvectors of Laplacian matrix, $\mathbf{\Lambda}$ is the diagonal matrix of its eigenvalues and $\mathbf{U}^*$ is the conjugate transpose of $\mathbf{U}$. $\mathbf{U}$ is defined as a graph Fourier basis [1] and the graph Fourier transform of $\mathbf{x}$ is defined as $\hat{\mathbf{x}} = \mathbf{U}^*\mathbf{x}$. Consider signals $\mathbf{x}$ supported on vertices whose samples are $x(n)$. We apply Laplacian matrix $\mathbf{L}$ to $\mathbf{x}$ and we get $\mathbf{L}\mathbf{x}$. This result can also be inferred from the graph Laplacian quadratic form [1], which is measure the smoothness of signal on graph G. The graph Laplacian quadratic form can be written as

$$S_2(\mathbf{x}) = \mathbf{x}^{\mathbf{T}}\mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{i,j} w_{ij}(x\,(i) - x\,(j))^2 \tag{3}$$

978-1-7281-2345-5/19/$31.00 ©2019 IEEE

Note that the weights are all non-negative. The signals $\mathbf{x}$ on graphs is smooth if the signals $\mathbf{x}$ have similar values between neighboring vertices by an edge with large weight. And the signals $\mathbf{x}$ on graphs is smoother, the graph Laplacian quadratic form $S_2(\mathbf{x})$ is smaller. Especially, when $\mathbf{x}$ has the same value on all its vertices, the value of $S_2(\mathbf{x})$ is zero.

To further describe the smoothness of signals on graphs, the gradient of smooth measure with respect to $\mathbf{x}$ is obtained, i.e. we have

$$\nabla S_2(\mathbf{x})=\begin{bmatrix} \sum_j w_{1j}\left(x\left(1\right)-x\left(j\right)\right) \\ \vdots \\ \sum_j w_{nj}\left(x\left(n\right)-x\left(j\right)\right) \end{bmatrix}=\mathbf{Lx} \tag{4}$$

Note from (4) that the right side of the equation omits the constant 2, but it does not affect the following discussion. The gradient of graph Laplacian quadratic form $\nabla S_2(\mathbf{x})$ is equal to applying Laplacian matrix $\mathbf{x}$ to signals $\mathbf{x}$. Furthermore, they are equivalent relations. They are from different concepts, $\nabla S_2(\mathbf{x})$ is derived from the concept of smooth signals and the other side is from the concept of linear transformation.

The gradient of the smooth function is only descirbes the smoothness of each vertex. To measure the smoothness of the whole signals $\mathbf{x}$ on graph, we add the gradient of the smooth function on all the vertices together, i.e.

$$S_1(\mathbf{x})=\sum_i\left|\sum_j w_{ij}\left[x\left(i\right)-x\left(j\right)\right]\right|=\|\mathbf{Lx}\|_1 \tag{5}$$

Equation (5) shows that sum of the gradient of smooth measure is L1 norm of the Laplacian operator of signal $\mathbf{x}$. In fact, $S_1(\mathbf{x})$ can also describe total variation on graphs. As $S_1(\mathbf{x})$ infers from the graph Laplacian quadratic form $S_2(\mathbf{x})$, it reflects structural information on graphs and gives a sophisticated characterization. From above, we have

Lemma 1. Assume that Laplacian matrix is a normalized graph Laplacian, then

$$S_1(\mathbf{x})=\|\mathbf{L}_{\mathrm{norm}}\mathbf{x}\|_1=\|\mathbf{x}-\mathbf{W}_{\mathrm{norm}}\mathbf{x}\|_1 \tag{6}$$

Note from (6) that it has the same form as in [7]. In [7], the idea that defines total variation on graphs, is inspired by classical digital signal processing, and it has the same total variation form. Different from the idea in [7], we derive this form from the idea of smooth graph signals. Compared to the graph Laplacian quadratic form, it penalizes transient changes less. In general, it is more sensitive to small total variation than quadratic form. By utilizing this information of total variation, a meaning graph topology can be obtained. In this paper, we also call $\|\mathbf{Lx}\|_1$ the total variation on graphs.

III. LEARNING LAPLACIAN MATRIX FROM SMOOTH SIGNALS

Assume that the vertices number N is known, and we have M samples value on each vertex. Laplacian matrix $\mathbf{L}$ is unknown. Let $\mathbf{X}=\left[\begin{array}{ccc}\mathbf{x}^1 & \cdots & \mathbf{x}^M\end{array}\right]\in R^{N\times M}$ be a graph signal sample matrix. Our goal is to learn Laplacian matrix $\mathbf{L}$ from smooth graph signals, i.e. $\mathbf{X}$. The smoothness and the spectral content of signals depend on graph topological structure.

To learn Laplacian matrix, which reflects graph topology, from graph signal sample matrix, we propose a new learning Laplacian framework. This uses further information on smooth graphs, i.e. total variation on graphs. Then, we compare this framework to machine learning method for estimating graphic structure and graph learning framework of Dong et al. [5]. The framework of Dong et al. is derived from maximum a posteriori (MAP) under Gaussian prior.

A. Machine Learning Method

To estimate a sparse inverse covariance matrix for GMRF, which is a machine learning method for estimating graphic structure, Lake and Tenenbaum propose in [8] to solve the following L1-regularized log-determinant optimization problem:

$$\begin{aligned} &\min_{\mathbf{L}_{pre}\succ 0,W,\sigma^2} tr\left(\mathbf{L}_{pre}\mathbf{S}\right)-\log\det\left(\mathbf{L}_{pre}\right)+\frac{\lambda}{p}\|\mathbf{W}\|_1 \\ &s.t. \quad \mathbf{L}_{pre}=\mathbf{D}-\mathbf{W}+\frac{1}{\sigma^2}\mathbf{I} \\ &\qquad w_{ii}=0, i=1,\ldots,N \\ &\qquad w_{ij}\geq 0, i=1,\ldots,N; i=1,\ldots,N \\ &\qquad \sigma^2>0 \end{aligned} \tag{7}$$

where $\mathbf{W}$ is weight adjacency matrix with elements which are the weighting coefficients w_{ij}. $\mathbf{L}_{pre}$ is the inverse covariance matrix to be estimated. $\mathbf{S}=\frac{1}{N}\mathbf{X}\mathbf{X}^T$ is sample covariance matrix. λ is a regularization parameter, and $\|\bullet\|_1$ denotes L1-norm. This optimization problem can be interpreted as estimating the inverse covariance matrix of a multivariate Gaussian distribution. As $\mathbf{L}_{pre}$ is not a graph Laplacian matrix, it only reflects partial relationship between graph topology and observations rather than interpreting as graphic topology that support graph signals. The optimization equation (7) can be solved by using CVX toolbox. This algorithm is called GL-LogDet.

B. Graph Learning Framework of Dong et al.

Dong et al. consider the following model:

$$\mathbf{y}=\mathbf{U}\hat{\mathbf{x}}+u_{\mathbf{x}}+\mathbf{e} \tag{8}$$

where $\hat{\mathbf{x}}$ is the graph Fouries transform of $\mathbf{x}$ with the graph Fourier basis $\mathbf{U}$, and $u_{\mathbf{x}}$ is the mean of $\mathbf{x}$. $\mathbf{y}$ and $\mathbf{e}$ denote the graph signal observations and the multivariate Gaussian noise with mean zero and covariance $\sigma_{\mathbf{e}}^2\mathbf{I}_N$, respectively.Then, Dong et al. deduce maximum a posterior estimation of $\hat{\mathbf{x}}$ and obtain the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{L}\in R^{N\times N},\mathbf{X}\in R^{N\times M}} \|\mathbf{Y}-\mathbf{X}\|_F+\alpha tr\left(\mathbf{X}^T\mathbf{L}\mathbf{X}\right) \\ &\qquad +\beta\|\mathbf{L}\|_F \\ &s.t. \quad tr\left(\mathbf{L}\right)=N \\ &\qquad \mathbf{L}_{ij}=\mathbf{L}_{ji}\leq 0, i\neq j \\ &\qquad \mathbf{L}\cdot\mathbf{1}=\mathbf{0} \end{aligned} \tag{9}$$

Note that $\mathbf{L}$ and $\mathbf{X}$ are respectively Laplacian matrix and graph signals with noiseless version of graph signal observations $\mathbf{Y}$. $\mathbf{0}$, $\mathbf{1}$ and $tr(\bullet)$ are the constant one vector and the zero vector, the trace of a matrix, respectively. Besides, α is the tune parameter.

To solve this optimization problem, Dong et al. employ an alternating minimization scheme where they fix one variable and solve the other one. In particular, firstly, by setting $\mathbf{X} = \mathbf{Y}$, Dong et al. solve the following problem:

$$\begin{aligned} \min_{\mathbf{L}\in R^{N\times N}} & \alpha tr\left(\mathbf{X}^T\mathbf{L}\mathbf{X}\right) + \beta\|\mathbf{L}\|_F \\ s.t. \quad & tr\left(\mathbf{L}\right) = N \\ & \mathbf{L}_{ij} = \mathbf{L}_{ji} \leq 0, i \neq j \\ & \mathbf{L}\cdot\mathbf{1} = \mathbf{0} \end{aligned} \tag{10}$$

where β is the tune parameter. Secondly, Dong et al. assume that $\mathbf{L}$ is known and solve the following optimization problem:

$$\min_{X\in R^{N\times M}} \|\mathbf{Y}-\mathbf{X}\|_F + \alpha tr\left(\mathbf{X}^T\mathbf{L}\mathbf{X}\right) \tag{11}$$

Equation (10) can be solved using convex optimization toolbox while (11) have closed-form solution. The whole algorithm is called GL-SigRep.

C. The Proposed Framework

Smoothness analysis leads to total variation to describe smooth graph signals. However, state-of-the-art methods don't use this information. And we propose a new framework to learning Laplacian matrix by using total variation on graphs. The optimization problem can be expressed as:

$$\begin{aligned} \min_{\mathbf{L}\in R^{N\times N},\mathbf{X}\in R^{N\times M}} & \|\mathbf{Y}-\mathbf{X}\|_F + \varsigma tr\left(\mathbf{X}^T\mathbf{L}\mathbf{X}\right) \\ & + \zeta\|\mathbf{L}\mathbf{X}\|_1 + \xi\|\mathbf{L}\|_F \\ s.t. \quad & tr\left(\mathbf{L}\right) = N \\ & \mathbf{L}_{ij} = \mathbf{L}_{ji} \leq 0, i \neq j \\ & \mathbf{L}\cdot\mathbf{1} = \mathbf{0} \end{aligned} \tag{12}$$

where ς, ζ and ξ are all the tunable parameters. Compared with equation (9), this optimization scheme take the advantage of total variation on graphs.

In order to solve this optimization problem, we set $\mathbf{X} = \mathbf{Y}$ and then have the following optimization problem :

$$\begin{aligned} \min_{\mathbf{L}\in R^{N\times N}} & \varsigma tr\left(\mathbf{X}^T\mathbf{L}\mathbf{X}\right) + \zeta\|\mathbf{L}\mathbf{X}\|_1 + \xi\|\mathbf{L}\|_F \\ s.t. \quad & tr\left(\mathbf{L}\right) = N \\ & \mathbf{L}_{ij} = \mathbf{L}_{ji} \leq 0, i \neq j \\ & \mathbf{L}\cdot\mathbf{1} = \mathbf{0} \end{aligned} \tag{13}$$

where Lapalacian matrix in this scheme is symmetric , which means that there are only $\frac{N(N+1)}{2}$ entries to solve. Thus,we solve the half-vectorization of $\mathbf{L}$ and rewrite the problem of (13) as:

$$\begin{aligned} \min_{vech(\mathbf{L})} & \varsigma vec(\mathbf{X}\mathbf{X}^T)^T\mathbf{M}_{du}vech(\mathbf{L}) \\ & + \zeta\|\mathbf{M}_{du}vech(\mathbf{L})vec(\mathbf{X})\|_1 \\ & + \xi vech(\mathbf{L})^T\mathbf{M}_{du}^T\mathbf{M}_{du}vech(\mathbf{L}) \\ s.t. \quad & \mathbf{A}vech(\mathbf{L}) = \mathbf{0} \\ & \mathbf{B}vech(\mathbf{L}) \leq \mathbf{0} \end{aligned} \tag{14}$$

where $vech(\mathbf{L})$ and $vec(\mathbf{L})$ are the half-vectorization vectorization and the vectorization of $\mathbf{L}$,respectively. $\mathbf{M}_{du}$ and $\mathbf{0}$ denote duplication matrix, respectively. $\mathbf{A}$ and $\mathbf{B}$ are respectively dependent on equality constraints and inequality constraints. See more detail in [9] .

We utilize convex optimization method [10] to solve equation (14), and this framework achieves a better performance than GL-LogDet and GL-SigRep. We denote this learning algorithm as GL-TV.

IV. NUMERICAL EXPERIMENTS

In this section, we evaluate the performance of GL-TV by different data sets, including random graph with Gaussian weights (Gaussian), Erdos-Renyi random graph (ER) and scale-free graph with preferential attachment (pa). We compare GL-TV to GL-SigRep and GL-LogDet through both visual and quantitative comparisons. And we use the evaluation criteria,named Recall [11], to compare their performance. To solve the above optimization problems, We use the CVX toolbox. Note that α, β, γ, ς, ζ, ξ are six important parameters for these optimization problems and different data sets have different parameters.

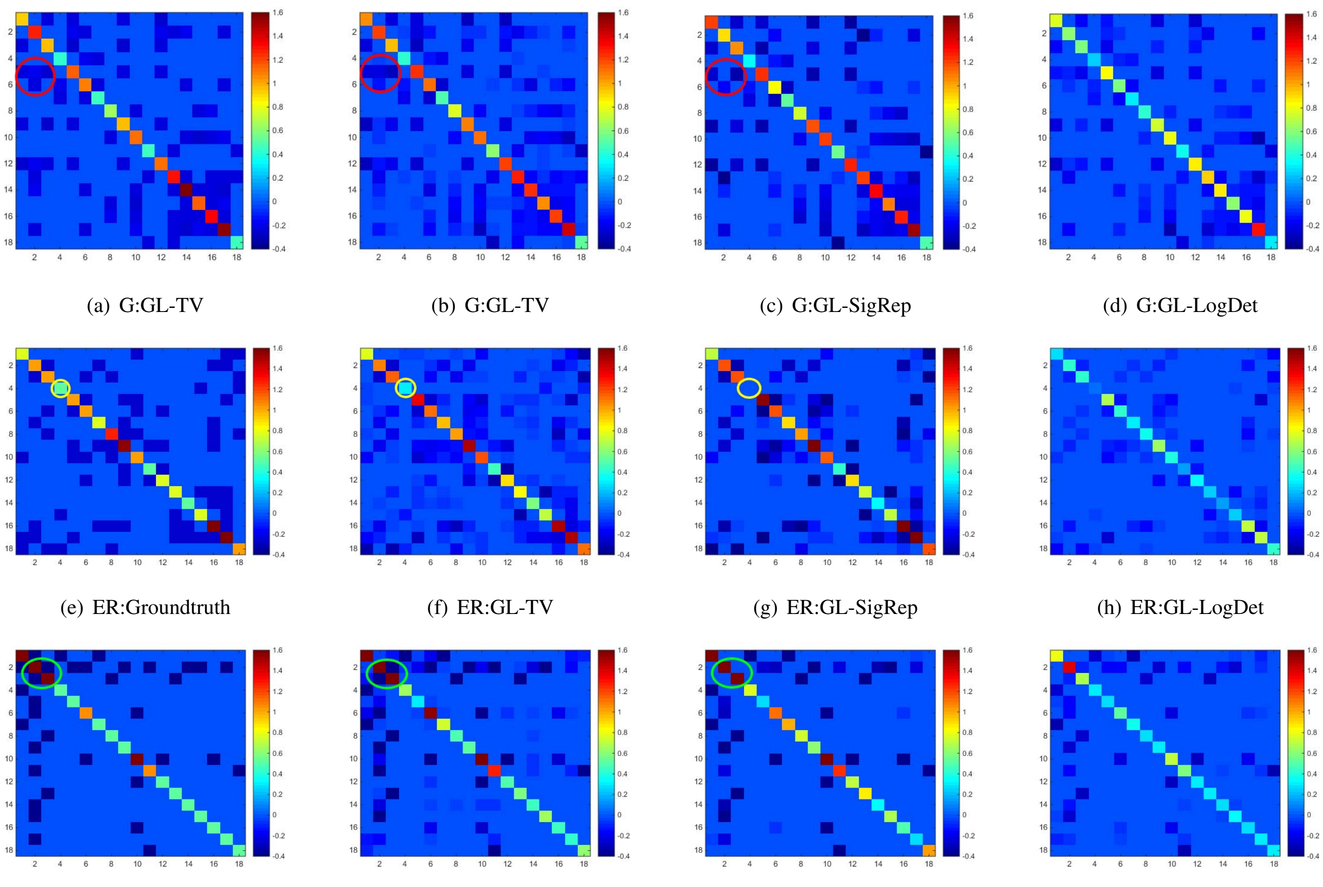
Consider Gaussian graph represented by a graph G with 18 vertices, we use Gaussian radial basis(GRF) function $\exp\left(-\frac{d(i,j)}{2\sigma^2}\right)$ as weighting coefficients w_{nm}. The threshold parameter of GRF is equal to 0.8 and σ = 0.55.

Then, we sample 100 observations on each vertex. And we learn Laplacian matrix from signal observations $\mathbf{Y}$ and compare these three algorithms by visualization and quantification. The top row of Fig.1 shows that GL-TV algorithm is more consist with groundtruth than GL-SigRep and GL-LogDet. Especially, we can visually see the difference between them in detail in red circles.

Moreover, we consider pa graph and ER graph with 18 vertices. As shown in Fig. 1, the middle and the bottom rows , Laplacian matrix learned by GL-TV is also more consist with groundtruth than GL-SigRep and GL-LogDet. Therefore, GL-TV get a more meaningful Laplacian matrix.

TABLE I
GL-TV, GL-SIGREP AND GL-LOGDET PERFORMANCE COMPARISON ON DIFFERENT SETS

Algorthm	Gaussian	ER	pa
GL-TV	0.9760	0.9808	0.9981
GL-SigRep	0.9186	0.8644	0.9205
GL-LogDet	0.8856	0.9214	0.9266



(a) G:GL-TV (b) G:GL-TV (c) G:GL-SigRep (d) G:GL-LogDet

(e) ER:Groundtruth (f) ER:GL-TV (g) ER:GL-SigRep (h) ER:GL-LogDet

(i) pa:Groundtruth (j) pa:GL-TV (k) pa:GL-SigRep (l) pa:GL-LogDet

Fig. 1. The learned graph Laplacian matrices. From the left to the right, each column is the groundtruth Laplacian matrix,the Laplacian matrix learned by GL-TV, the Laplacian matrix learned by GL-SigRep, the Laplacian matrix learned by GL-LogDet, respectively. The rows from the top to the bottom, each row is the Laplacian matrix learned from the Gaussian RBF graph, the ER graph and the pa graph, respectively.

Besides, we quantify the performance of these three algorithms. This result is the mean of the 500 independent experiments on tree different graphs. Table I shows that GL-L1 has the state-of-the-art performance than GL-SigRep and GL-LogDet. Especially, from the pa graph experiment, GL-TV has outstanding performance, i.e. 0.9981, and the other two algorithm is about 0.92. Therefore, TV constraint significantly improves the performance for learning Laplacian matrix.

V. CONCLUSION

This paper analyzed the measure of smooth graph signals, i.e. the Laplacian quadratic form, and presented a framework for learning a more meaningful graph. This novel framework is constructed by using the total variation under smooth prior. Experiment results indicate that the proposed framework achieves better performance than existing frameworks when only signal observations are available. This unified theoretical framework has both theory and practice meaning to the further research of graph learning.

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 Processing Magazine*, vol. 30, no. 3, pp. 83–98, May 2013.

[2] L. Stankovic, M. Dakovic, and E. Sejdic, “Vertex-frequency analysis: A way to localize graph spectral components [lecture notes],” *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 176–182, July 2017.

[3] N. Perraudin and P. Vandergheynst, “Stationary signal processing on graphs,” *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3462–3477, July 2017.

[4] 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 Magazine*, vol. 31, no. 5, pp. 80–90, Sept 2014.

[5] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, “Learning laplacian matrix in smooth graph signal representations,” *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6160–6173, Dec 2016.

[6] ——, “Laplacian matrix learning for smooth graph signal representation,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 3736–3740.

[7] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs: Frequency analysis,” *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, June 2014.

[8] B. Lake and J. Tenenbaum, “Discovering structure by learning sparse graph,” in *Proceedings of the 33rd Annual Cognitive Science Conference*, 2010, pp. 778–783.

[9] K. M. Abadir and J. R. Magnus, *Matrix Algebra*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.:

Cambridge Univ. Press, 2004.
[11] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.