K-Graphs: An Algorithm for Graph Signal Clustering and Multiple Graph Learning

Hesam Araghi, Mohammad Sabbaqi, and Massoud Babaie–Zadeh, *Senior Member, IEEE*

***Abstract*—In graph signal processing (GSP), graph learning is concerned with the inference of an underlying graph best capable of modeling a dataset of graph signals. However, more complex datasets are derived from multiple underlying graphs. In such instances, it is necessary to learn multiple graph structures, each corresponding to the graph signals residing on the same structure. In other words, the graph signals need to be partitioned into a set of clusters, with a designated topology for each cluster. In this letter, inspired from classical K-means, a new algorithm for multiple graph learning, called K-graphs, is proposed. Numerical experiments demonstrate the high performance of this algorithm, in both graph learning and data clustering.**

***Index Terms*—K-graphs, graph signal processing (GSP), multiple graph learning, K-means, graph Laplacian matrix.**

I. Introduction

GRAPH SIGNALS are generic mathematical tools for representation of the data that lie on complicated structures, such as social and computer networks or digital images. When a well-defined graph is available for a dataset, the analysis and inference of the data can be performed by the tools in the emerging field of graph signal processing (GSP) [1], [2]. However, in many datasets, the underlying graph is not known, and it should be learned from the data. Many GSP-based graph learning algorithms have recently been proposed [3] to infer an underlying graph topology from a dataset of graph signals (e.g., based on the smoothness of the data over the underlying graph), probably by also exploiting some prior knowledge such as edge sparsity or connectivity of the graph [4].

There are still more complicated natural data that cannot be well described by a single graph. In fact, there are multiple latent graphs that rule such data structures. This type of data arises the need for learning multiple graphs from a dataset of graph signals. An example of such datasets are brain fMRI data, where captured signals are related to various brain networks corresponding to different human functions [5].

Before modeling the data on graphs as graph signals, the domain knowledge was mainly represented by joint probability distributions, and graphs were used to express the structural properties of these distributions [6]. For example, in [7]–[9], by assuming that the data follows a multivariate Gaussian distribution, the graph learning problem has been reduced to a Gaussian Markov Random Field (GMRF) estimation, which can be defined entirely by the inverse covariance matrix (i.e., precision matrix) of the data. Then, the graph learning has been performed by precision matrix estimation based on some prior information such as sparsity [7], [8]. In [7], [8], the interpretation of the inferred graph suffers from self-loops and negative weights, so [9] imposed a graph Laplacian matrix structure on the precision matrix to overcome this problem.

By the advent of GSP and treating the data as graph signals, more interpretable methods have been proposed for learning the graph. In [10], the data is modeled using factor analysis with a prior distribution that forces the signal to be smooth on the graph. Then, applying a maximum a posteriori (MAP) estimator results in a non-convex optimization problem, solved by alternating minimization. A computationally more efficient version of this algorithm is proposed in [4]. Other previous works include graph learning by structural dictionary learning [11], and using pre-defined spectral templates to learn the eigenvalues of the Laplacian matrix [12].

However, none of the previously mentioned methods have considered learning of multiple graphs from the data. Up to our best knowledge, the only work addressing this problem is [5], in which an algorithm called graph Laplacian mixture model (GLMM) has been proposed for this purpose. In that work, the dataset consists of smooth signals over different graphs, and a Gaussian mixture model (GMM) is assumed for the data. To obtain graph Laplacian matrices, a MAP estimation problem has been introduced and solved via expectation maximization (EM) algorithm, in which the covariance matrix estimation step has been replaced by a graph learning algorithm.

In this letter, a new algorithm for multiple graph learning is proposed. More precisely, we are interested to jointly cluster a set of 'graph signals' coming from several unknown graphs and learn the underlying graph for each cluster.[1] To this end, K-means algorithm is reformulated based on the *smoothness* of a signal on a graph, to provide a tool for learning multiple graphs over data with a complicated structure. While [5] assumes a Gaussian distribution for the data and proposes a GMM based

Manuscript received June 6, 2019; revised August 1, 2019; accepted August 6, 2019. Date of publication August 21, 2019; date of current version September 3, 2019. This work was supported in part by Research Office of Sharif University of Technology. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Antonio Paiva. *(Corresponding author: Massoud Babaie-Zadeh.)*

The authors are with the Department of Electrical Engineering, Sharif University of Technology, Tehran 11365-11155, Iran (e-mail: hesam.araghi@ee.sharif.edu; sabbaqi.m@ee.sharif.edu; mbzadeh@yahoo.com).

Digital Object Identifier 10.1109/LSP.2019.2936665

[1]Note that this is fundamentally different from the algorithms such as spectral clustering (SC) [13]–[16] or locally linear approaches for clustering [17], [18]: If the dataset is composed of N data points $\mathbf{x}_1, \ldots, \mathbf{x}_N$, where $\forall i, \mathbf{x}_i \in \mathbb{R}^m$, then in SC-based algorithms the graph has N nodes, while in our work each graph has m nodes.

algorithm, our method is not limited to such an assumption, and as will be seen in simulations, it results in higher accuracy in both clustering and graph estimation, even for a dataset generated with Gaussian distribution.

The rest of the letter is organized as follows. In Section II, some basics of GSP are very briefly reviewed. Our method for multiple graph learning, called K-graphs, is proposed in Section III. Finally, Section IV is devoted to simulation results.

II. GSP Background

In this section, a brief review on the GSP is presented which provides the necessary basics/notations for the graph learning problem. Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$ be an undirected, positive-weighted graph without self-loops, where $\mathcal{V}$ is the vertex set, $\mathcal{E}$ is the edge set, and $\mathbf{W}$ is the symmetric weighted adjacency matrix. If there is an edge between nodes i and j, $\mathbf{W}[i,j]$ (similarly $\mathbf{W}[j,i]$) denotes the positive weight of that edge, otherwise it is zero. In this letter, the square brackets $[i,j]$ after a matrix indicate the (i,j)-th entry of that matrix. A graph signal is a mapping $\mathbf{x} : \mathcal{V} \to \mathbb{R}^m$ that assigns a real value to each vertex, where $|\mathcal{V}| = m$ is the number of nodes. The graph Laplacian matrix for graph $\mathcal{G}$ is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is the weighted degree matrix, that is, $\mathbf{D} = \mathrm{diag}(\mathbf{W} \cdot \mathbf{1})$, in which $\mathbf{1}$ denotes the all-one vector. The set of valid graph Laplacian matrices is defined as

$$\mathcal{L} = \big\{\mathbf{L} \in \mathbb{R}^{m\times m} : \mathbf{L} = \mathbf{L}^T, \mathbf{L}[i,j] \le 0\ (\forall i \ne j), \mathbf{L}\cdot\mathbf{1} = \mathbf{0}\big\}, \quad (1)$$

where $\mathbf{0}$ is the all-zero vector, and $(\cdot)^T$ denotes matrix transposition. Under aforementioned constraints, $\mathbf{L}$ becomes a symmetric positive semi-definite matrix. Therefore, it can be decomposed as $\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$, where $\mathbf{\Lambda}$ is a diagonal matrix with ascending-ordered non-negative eigenvalues and $\mathbf{V}$ is an orthonormal matrix containing the corresponding eigenvectors as its columns. In the GSP context, these eigenvectors are used as the basis for Graph Fourier Transform (GFT) [1]. So, GFT is defined as $\hat{\mathbf{x}} = \mathbf{V}^T\mathbf{x}$, where $\hat{\mathbf{x}}$ denotes the GFT coefficients of the graph signal $\mathbf{x}$. A graph signal is called smooth over $\mathbf{L}$, if most of signal energy is concentrated in a few first GFT coefficients.

An essential expression in GSP is the graph Laplacian quadratic form [1], which measures the signal smoothness:

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \sum_{i,j}\mathbf{W}[i,j](x_i - x_j)^2 = \sum_{k=1}^{m}\lambda_k|\hat{x}_k|^2, \quad (2)$$

where λ_k's are the eigenvalues of $\mathbf{L}$, and x_i and $\hat{x}_k$ are the i-th and k-th entries of the vectors $\mathbf{x}$ and $\hat{\mathbf{x}}$, respectively. Clearly, the quadratic form (2) is a non-negative value that penalizes the difference between two strongly connected nodes, hence it measures signal variations over the graph.

Based on the definition of GFT, graph filtering can be done by modifying the eigenvalues of $\mathbf{L}$ via a function $h(\lambda_k)$ [1]. For lowpass filtering, a common approach is to set $h(\lambda_k) = \lambda_k^{-1/2}$ $(h(0) = 0)$ [10], which is equivalent to multiplying the input signal $\mathbf{x}$ by the matrix $\mathbf{L}^{-1/2} \triangleq \mathbf{V}(\mathbf{\Lambda}^\dagger)^{1/2}\mathbf{V}^T = (\mathbf{L}^\dagger)^{1/2}$, where $(\cdot)^\dagger$ indicates Moore-Penrose pseudo-inverse.

III. K-Graphs Algorithm

Let the dataset $\mathcal{X}$ consist of N graph signals $\mathbf{x}_n \in \mathbb{R}^m$, $n = 1,\dots,N$, and each signal comes from one of the K undirected graphs $\mathcal{G}_1,\dots,\mathcal{G}_K$. It is also not known that which graph signal has come from which graph. Then, the objective of multiple graph learning is to find the set of these K graphs $\{\mathcal{G}_k\}_{k=1}^K$ from the data. This problem can also be seen as partitioning the dataset into K clusters $\mathcal{X}_1,\dots,\mathcal{X}_K$ that have the best fit on the graphs $\mathcal{G}_1,\dots,\mathcal{G}_K$, respectively. We use the Laplacian matrix $\mathbf{L}_k$ to represent the k-th undirected graph $\mathcal{G}_k$, and our criterion of fitness of a graph signal $\mathbf{x}$ on $\mathcal{G}_k$ is the smoothness of the signal over the graph, as defined in (2). In other words, the smaller $\mathbf{x}^T\mathbf{L}_k\mathbf{x}$, the better fit of the signal $\mathbf{x}$ on the k-th graph.

To solve the above problem, we adopt an algorithm inspired from K-means. In K-means, the goal is to find cluster centers $\mathbf{c}_k$ and cluster sets $\mathcal{X}_k$, $k = 1,\dots,K$, that minimize the objective function [19], [20]

$$\sum_{\mathbf{x}\in\mathcal{X}_1} d(\mathbf{x},\mathbf{c}_1)^2 + \sum_{\mathbf{x}\in\mathcal{X}_2} d(\mathbf{x},\mathbf{c}_2)^2 + \cdots + \sum_{\mathbf{x}\in\mathcal{X}_K} d(\mathbf{x},\mathbf{c}_K)^2 = \sum_{k=1}^{K}\sum_{\mathbf{x}\in\mathcal{X}_k} d(\mathbf{x},\mathbf{c}_k)^2, \quad (3)$$

where $\mathbf{x}$ is a multi-dimensional point in the dataset $\mathcal{X}$, and $d(\mathbf{x},\mathbf{c}_k)$ is a distance function measuring the dissimilarity between $\mathbf{x}$ and $\mathbf{c}_k$. For minimizing (3), K-means uses alternating minimization for $\mathbf{c}_k$'s and $\mathcal{X}_k$'s, which contains an initialization and then iterating between two steps:

Initialization: Partition $\mathbf{x}_1,\dots,\mathbf{x}_N$ randomly into K clusters $\mathcal{X}_1,\dots,\mathcal{X}_K$.

Step 1: Fix $\mathcal{X}_k$'s and recompute the centers by $\mathbf{c}_k = \mathrm{argmin}_{\mathbf{c}}\sum_{\mathbf{x}\in\mathcal{X}_k} d(\mathbf{x},\mathbf{c})^2$ for $k = 1,\dots,K$. For the Euclidean distance, $\mathbf{c}_k$ is the centroid of the points in $\mathcal{X}_k$, i.e. $\frac{1}{|\mathcal{X}_k|}\sum_{\mathbf{x}\in\mathcal{X}_k}\mathbf{x}$.

Step 2: Fix $\mathbf{c}_k$'s and assign each data point $\mathbf{x} \in \mathcal{X}$ to one of the K clusters $\mathcal{X}_1,\dots,\mathcal{X}_K$ such that $\forall \mathbf{x} \in \mathcal{X}_k$ we have $k = \mathrm{argmin}_{k'} d(\mathbf{x},\mathbf{c}_{k'})$.

Now, inspired from (3), we express the multiple graph learning problem as

$$\underset{\substack{\mathbf{L}_1,\dots,\mathbf{L}_K\\ \mathcal{X}_1,\dots,\mathcal{X}_K}}{\text{minimize}} \quad \sum_{k=1}^{K}\sum_{\mathbf{x}\in\mathcal{X}_k}\mathbf{x}^T\mathbf{L}_k\mathbf{x} + \sum_{k=1}^{K} f(\mathbf{L}_k), \quad \text{s.t.} \quad \mathbf{L}_k \in \mathcal{L}, \quad 1 \le k \le K, \quad (4)$$

where the set $\mathcal{L}$ is defined in (1). The term $\mathbf{x}^T\mathbf{L}_k\mathbf{x}$ measures the smoothness of the signal $\mathbf{x}$ on $\mathbf{L}_k$, and $f(\mathbf{L}_k)$ consists of regularization terms. The main roles of these terms are 1) to prevent the trivial solution $\forall k, \mathbf{L}_k = \mathbf{0}$, and 2) to be able to control the number of off-diagonal, non-zero entries of $\mathbf{L}_k$, which is twice the number of edges in the graph $\mathcal{G}_k$ [4].

Here, instead of finding a center point $\mathbf{c}_k$ for each cluster $\mathcal{X}_k$, we are performing a single graph learning algorithm to compute a Laplacian matrix $\mathbf{L}_k$ such that the graph signals in $\mathcal{X}_k$ are smooth on the corresponding graph $\mathcal{G}_k$. Replacing center points with Laplacian matrices helps K-graphs to understand the structures of the clusters using the Laplacian matrices, unlike K-means that can only represent each cluster with a single center point.

Algorithm 1: K-Graphs.

Input: dataset $\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$, number of clusters K.
Output: graph Laplacian matrices $\mathbf{L}_k$ for $k = 1, \ldots, K$ and cluster indices i_n for $n = 1, \ldots, N$.
1: Initialize cluster indices $i_n^{(0)}$ for $n = 1, \ldots, N$.
2: $j \leftarrow 0$
3: **repeat**
4: Update $\mathbf{L}_k^{(j)}$ with partition $\mathcal{X}_k$ by solving (5) for all $k = 1, \ldots, K$.
5: Calculate $i_n^{(j+1)}$ from (6) with $\mathbf{L}_k^{(j)}$'s for $n = 1, \ldots, N$.
6: $j \leftarrow j + 1$
7: **until** convergence

For the initialization of the algorithm, we can randomly and independently partition the graph signals into K clusters.

After initialization, at the first step of our algorithm, each $\mathbf{L}_k$ is updated by a single graph learning from the graph signals in $\mathcal{X}_k$. Many single graph learning algorithms [3], [4] for obtaining $\mathbf{L}_k$ from the signals of $\mathcal{X}_k$ can be described as

$$\mathbf{L}_k = \underset{\mathbf{L}\in\mathcal{L}}{\operatorname{argmin}} \sum_{\mathbf{x}\in\mathcal{X}_k} \mathbf{x}^T\mathbf{L}\mathbf{x} + f(\mathbf{L}). \tag{5}$$

In the second step of our algorithm, the Laplacian matrices $\mathbf{L}_k$'s are fixed, and the graph signals are assigned to the clusters based on smoothness over the graph, using

$$i_n = \underset{1\le k\le K}{\operatorname{argmin}} \frac{\mathbf{x}_n^T\mathbf{L}_k\mathbf{x}_n}{\operatorname{trace}\{\mathbf{L}_k\}}, \tag{6}$$

where $i_n \in \{1, \ldots, K\}$ denotes the cluster index of the graph signal $\mathbf{x}_n$, and then $\mathcal{X}_k$ is formed as $\mathcal{X}_k = \{\mathbf{x}_n : 1 \le n \le N, i_n = k\}$ for $k = 1, \ldots, K$. The term $\operatorname{trace}\{\mathbf{L}_k\}$ in the denominator of (6) eliminates the dependency of $\mathbf{x}_n^T\mathbf{L}_k\mathbf{x}_n$ on the scale of the Laplacian matrix $\mathbf{L}_k$. As in K-means, to escape from local minima, it is appropriate to run the algorithm with various initializations and report the best clustering result.

The final proposed multiple graph learning algorithm (called K-graphs) is summarized in Algorithm 1.

Theorem 1: K-graphs algorithm converges in a finite number of iterations.

Proof: At each iteration of Algorithm 1, the objective function of (4) does not increase, because in line 4 the minimization problem (5) is solved by a single graph learning algorithm for each $k = 1, \ldots, K$, and in line 5 the graph signals are assigned to the clusters such that the objective function decreases or does not change. Moreover, the possible number of ways to assign N graph signals to K clusters is a finite number. Consequently, the algorithm must converge in a finite number of iterations.

Remark: Note that the above theorem is valid for Algorithm 1, in which line 4 is solved exactly. However, where line 4 is going to be solved with another iterative algorithm, the imperfections of that algorithm may affect the convergence of K-graphs. Theorem 1 shows that such imperfections are not intrinsically from K-graphs, but from the algorithm used to solve line 4 of K-graphs. ■

IV. Numerical Results

In this section, the performance of K-graphs is numerically studied. We compare K-graphs with GLMM [5]. To generate synthetic graph signals, we create the random graphs with `gsp_sensor` command from GSPBOX toolbox [21]. Then, to create smooth signals for each graph, we use lowpass filtered version of white noises on that graph. In more details, a graph signal $\mathbf{s}_\mathcal{G}$ on graph $\mathcal{G}$ is created as $\mathbf{s}_\mathcal{G} = \mathbf{L}^{-1/2}\mathbf{w}$, where $\mathbf{w}$ is a white multivariate Gaussian signal with $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{L}$ is the Laplacian matrix of the graph, and $\mathbf{I}$ denotes the identity matrix. Then, the lowpass filtered graph signal $\mathbf{s}_\mathcal{G}$ is corrupted by a white Gaussian noise $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2\mathbf{I})$ independent of $\mathbf{w}$, i.e. $\mathbf{x} = \mathbf{s}_\mathcal{G} + \mathbf{n}$. In the following experiments, the dataset $\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$ consists of $N = 1000$ graph signals, and $\mathbf{x}_n$'s are 30×1 vectors ($m = 30$). Each graph signal in the dataset comes with the equal probability ($1/K$) from the K graphs $\mathcal{G}_1, \ldots, \mathcal{G}_K$. The signal to noise ratio (SNR) of the dataset is defined as $\mathrm{SNR} \triangleq 10\log_{10}(\mathbb{E}\{\|\mathbf{s}_\mathcal{G}\|_2^2\}/\mathbb{E}\{\|\mathbf{n}\|_2^2\})$, where $\mathbb{E}\{\cdot\}$ indicates the statistical expectation and $\mathbb{E}\{\|\mathbf{s}_\mathcal{G}\|_2^2\}$ is obtained by iterated expectations as

$$\mathbb{E}\{\|\mathbf{s}_\mathcal{G}\|_2^2\} = \mathbb{E}_k\{\mathbb{E}\{\|\mathbf{s}_{\mathcal{G}_k}\|_2^2\}\} = \frac{1}{K}\sum_{k=1}^K \operatorname{trace}\{\mathbf{L}_k^\dagger\}. \tag{7}$$

From $\mathbb{E}\{\|\mathbf{n}\|_2^2\} = m\sigma_n^2$ and (7), we have

$$\mathrm{SNR} = 10\log_{10}\left(\frac{1}{Km\sigma_n^2}\sum_{k=1}^K \operatorname{trace}\{\mathbf{L}_k^\dagger\}\right). \tag{8}$$

For initialization of the algorithms, K-graphs randomly partitions the dataset into K clusters, and GLMM starts with K random Laplacian matrices [5]. Both algorithms are run with 50 different random starting points in each realization and the final output is the one that minimizes the following clustering objective function:

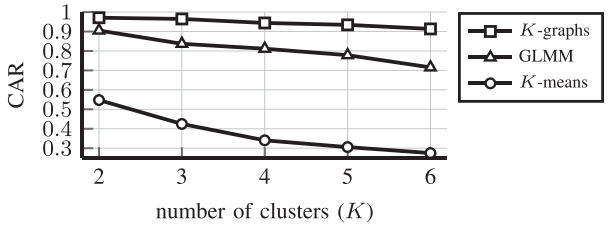
$$\sum_{k=1}^K \sum_{\mathbf{x}\in\mathcal{X}_k} \frac{\mathbf{x}^T\mathbf{L}_k\mathbf{x}}{\operatorname{trace}\{\mathbf{L}_k\}}. \tag{9}$$

For the regularization term $f(\mathbf{L}_k)$, as proposed in [4], we use

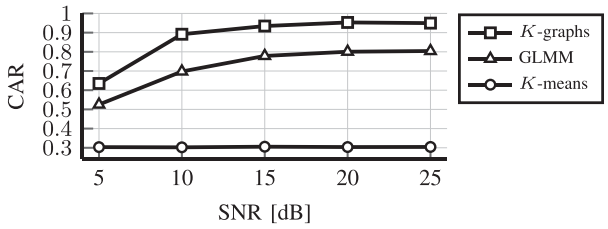
$$f(\mathbf{L}_k) = -\alpha\sum_{i=1}^m \log(\mathbf{L}_k[i,i]) + \beta\sum_{i\sim j} |\mathbf{L}_k[i,j]|^2, \tag{10}$$

where α and β are controlling parameters. The first (logarithmic) term in the right-hand side of (10) eliminates the possibility of zero trivial solution by forcing the degrees of the nodes to be positive, and the second (quadratic) term is added to control the number of off-diagonal, non-zero elements in the Laplacian matrices. For more details on the effects of regularization terms and choosing the parameters α and β, interested readers are redirected to [4]. We use the command `gsp_learn_graph_log_degrees` in GSPBOX toolbox [21] to solve the single graph learning problem (5) with the algorithm of [4].

In the following, the clustering capability of the algorithm is studied in Subsection IV-A. In Subsection IV-B, the multiple graph learning performance of K-graphs is compared with GLMM based on the deviation of the Laplacian matrices from

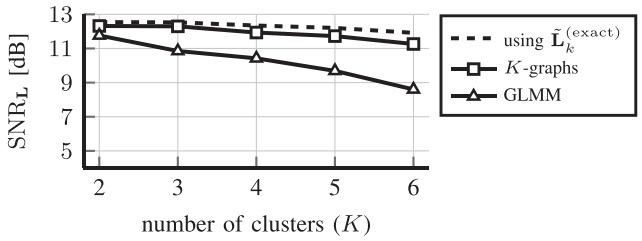


(a) different number of clusters for $\mathrm{SNR} = 15\,\mathrm{dB}$

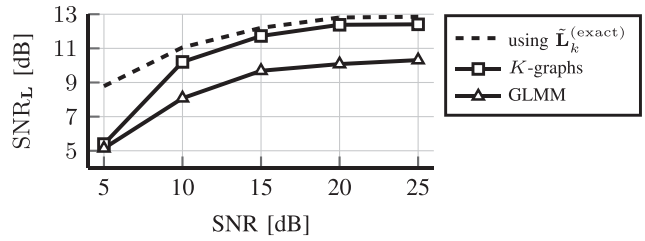


(b) different values of SNR for $K = 5$

Fig. 1. Comparing clustering accuracy ratio (CAR) of K-graphs, GLMM, and K-means for (a) different number of clusters (K) and (b) for different values of SNR, with $m = 30$ and $N = 1000$.



(a) different number of clusters for $\mathrm{SNR} = 15\,\mathrm{dB}$



(b) different values of SNR for $K = 5$

Fig. 2. Estimation accuracy of the Laplacian matrices ($\mathrm{SNR}_{\mathbf{L}}$) for K-graphs and GLMM along with the case that the Laplacian matrices are learned from true clusters (using $\tilde{\mathbf{L}}_k^{(\mathrm{exact})}$) for (a) different number of clusters (K) and (b) for different values of SNR, with $m = 30$ and $N = 1000$.

the ground truth. In all simulations, each point in the figures is the average of 200 independent realizations. The maximum number of iterations for GLMM algorithm is set to 50.

A. Clustering Performance of K-Graphs

In the first experiment, the clustering accuracy of K-graphs is assessed for different number of clusters K and different values of SNR. To find the mapping of the true cluster labels to the calculated cluster labels, we search through all $K!$ permutations in calculated labels and consider the best matching with the true labels as the correct mapping. Then, the ratio of the matching between the true and calculated labels to the total number of graph signals is defined as *clustering accuracy ratio (CAR)*.

In Fig. 1, the clustering accuracy ratio of K-graphs is compared with GLMM. Moreover, the result of classical K-means is added as an algorithm that does not benefit the graphical structure of the data. `k-means++` [22] is used for initialization of K-means and again the best clustering result of 50 runs with different initialization points is chosen as the output (objective function (3) is used for finding the best result). As it can be seen from the curves in Fig. 1a, the accuracy of K-graphs is close to 1 and the accuracy of GLMM is lower than K-graphs. The worst performance belongs to K-means, which fails to cluster the graph signals properly. Figure 1b shows that K-graphs outperforms the two other algorithms for different SNRs.

It seems that the higher performance of K-graphs compared with GLMM stems from the hard clustering property, which is inherited from K-means. Particularly, in K-graphs, each data point is used in the learning of only the graph on which the data point is the smoothest. On the other hand, each data point in GLMM participates partially (proportional to its posterior probabilities) in the learning of the other graphs as well, which may have negative effects on estimating the corresponding Laplacian matrices.

B. Multiple Graph Learning Performance of K-Graphs

In the second experiment, the similarity of the estimated and true graphs is measured for both algorithms. Our similarity criterion is the following SNR

$$\mathrm{SNR}_{\mathbf{L}} = 10\log_{10}\left(\frac{\sum_{k=1}^{K}\|\tilde{\mathbf{L}}_k^{(\mathrm{true})}\|_F^2}{\sum_{k=1}^{K}\|\tilde{\mathbf{L}}_k^{(\mathrm{true})} - \tilde{\mathbf{L}}_k\|_F^2}\right), \qquad (11)$$

where $\|\cdot\|_F$ indicates the Frobenius norm, and $\tilde{\mathbf{L}}_k^{(\mathrm{true})}$ and $\tilde{\mathbf{L}}_k$ are the normalized versions of true ($\mathbf{L}_k^{(\mathrm{true})}$) and estimated ($\mathbf{L}_k$) Laplacian matrices divided by their traces, respectively, for $k = 1, \ldots, K$. Additionally, we introduce another Laplacian matrix $\tilde{\mathbf{L}}_k^{(\mathrm{exact})}$ which is learned from the true clustered graph signals in cluster k. Calculating $\mathrm{SNR}_{\mathbf{L}}$ with $\tilde{\mathbf{L}}_k^{(\mathrm{exact})}$ instead of $\tilde{\mathbf{L}}_k$ can offer a good performance criterion for our proposed algorithm.

Figure 2 illustrates $\mathrm{SNR}_{\mathbf{L}}$ for K-graphs and GLMM along with the one obtained from $\tilde{\mathbf{L}}_k^{(\mathrm{exact})}$. As Fig. 2a depicts, K-graphs follows the ideal case, while in GLMM algorithm $\mathrm{SNR}_{\mathbf{L}}$ drops as K increases. In Fig. 2b, it is seen that the estimation accuracy of both algorithms are far below the ideal case in low SNR, and K-graphs gets closer to the ideal case compared with GLMM.

V. Conclusion

In this letter, we proposed an algorithm called K-graphs for multiple graph learning problem, where signal smoothness on the underlying graph is used as a dissimilarity measure. Numerical simulations demonstrated that K-graphs achieves a good performance in both clustering accuracy and multiple 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 Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

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

[3] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Process. Mag.*, vol. 36, pp. 44–63, May 2019.

[4] V. Kalofolias, "How to learn a graph from smooth signals," in *Proc. Int. Conf. Artif. Intell. Statist.*, vol. 51, Cadiz, Spain, 2016, pp. 920–929.

[5] H. P. Maretic and P. Frossard, "Graph Laplacian mixture model," Oct. 2018, *arXiv:1810.10053v1*.

[6] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA: MIT Press, 2009.

[7] A. P. Dempster, "Covariance selection," *Biometrics*, vol. 28, pp. 157–175, 1972.

[8] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[9] B. Lake and J. Tenenbaum, "Discovering structure by learning sparse graphs," in *Proc. 32nd Annu. Meeting Cogn. Sci. Soc.*, Portland, OR, USA, Aug. 2010, pp. 778–784.

[10] 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, Dec. 2016.

[11] H. P. Maretic, D. Thanou, and P. Frossard, "Graph learning under sparsity priors," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2017, pp. 6523–6527.

[12] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 3, pp. 467–483, Sep. 2017.

[13] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2002, pp. 849–856.

[14] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[15] K. Zhan, C. Zhang, J. Guan, and J. Wang, "Graph learning for multiview clustering," *IEEE Trans. Cybern.*, vol. 48, no. 10, pp. 2887–2895, Oct. 2018.

[16] K. Zhan, F. Nie, J. Wang, and Y. Yang, "Multiview consensus graph clustering," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1261–1270, Mar. 2019.

[17] M. Wu and B. Schölkopf, "A local learning approach for clustering," in *Proc. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2006, pp. 1529–1536.

[18] Y. Yang, D. Xu, F. Nie, S. Yan, and Y. Zhuang, "Image clustering using local discriminant models and global integration," *IEEE Trans. Image Process.*, vol. 19, no. 10, pp. 2761–2773, Oct. 2010.

[19] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York, NY, USA: Wiley, 1973.

[20] S. Z. Selim and M. A. Ismail, "K-means-type algorithms: A generalized convergence theorem and characterization of local optimality," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 1, pp. 81–87, Jan. 1984.

[21] N. Perraudin *et al.*, "GSPBOX: A toolbox for signal processing on graphs," Aug. 2014, *arXiv:1408.5781v2*. [Online]. Available: https://epfl-lts2.github.io/gspbox-html/index.html

[22] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2007, pp. 1027–1035.