Kernel-based Joint Multiple Graph Learning and Clustering of Graph Signals

Mohamad H. Alizade, Aref Einizade

***Abstract*—Within the context of Graph Signal Processing (GSP), Graph Learning (GL) is concerned with the inference of a graph's topology from nodal observations, i.e., graph signals. However, data is often in mixed form, relating to different underlying structures. This heterogeneity necessitates the joint clustering and learning of multiple graphs. In many real-life applications, there are available node-side covariates (i.e., kernels) that imperatively should be incorporated, which has not been addressed by the rare graph signal clustering approaches. To this end and inspired by the rich K-means framework, we propose a novel kernel-based algorithm to incorporate this node-side information as we jointly partition the signals and learn a graph for each cluster. Numerical experiments demonstrate its effectiveness over the state-of-the-art.**

***Index Terms*—Graph Signal Processing (GSP), Graph Learning (GL), Clustering, Kernel subspace.**

I. Introduction

THE emerging field of graph signal processing (GSP) [1], [2], [3], which is concerned with processing nodal values on the vertices of an underlying meaningful graph (i.e., graph signals), has unveiled many analytical methods. Some datasets land themselves naturally to a graph structure, such as traffic data; however, in many others, the underlying graph topology is not known a priori, leading to an increasing popularity of graph learning (GL) via GSP [4], [5]. While the problem of GL from data is generally ill-imposed, this resolves by modeling their relationship [6]. The first kind of these models resorted to joint probability distributions to describe conditional dependencies as graphs [7]. When the data follows a multivariate Gaussian distribution, the problem simplifies to estimating a sparse precision matrix [8]. However, the resulting graph contains self-loops and negative weights, making them hard to interpret. With the advent of GSP, data is mapped to a set of nodes and treated as graph signals. One type of model considers physical processes like diffusion to explain the observations [9]. Another type assumes that the neighboring nodes have similar values, promoting global smoothness [6][10].

All the above methods deal with homogeneous datasets, i.e., all graph signals relate to one single graph. Nevertheless, most real datasets form clusters with different underlying structures. This heterogeneity means the graph signals are partitioned, and each partition relates to a different (unknown) graph. On the other hand, in many cases, there is node-side information that ultimately changes how each data entity relates to another. This information may be the spatial coordinates of a node in a sensor network or take non-numeric forms such as categorical or text [11]. The fMRI datasets are a prime example: imaging data reveals brain activity in different parts of the brain. However, each graph signal may belong to a different cognitive process and, hence, a different functional network. Furthermore, the brain's physical connections influence functional networks and may serve as prior covariates [12].

In this letter, we propose a novel framework incorporating node-side information into the clustering of graph signals as it jointly learns their explicit topology. To this end, the node-side information is mapped to elements of a kernel's Hilbert space. The kernel matrix represents their covariates in that space, and low-pass filters are formed by combining the Laplacian and inverse of kernel matrices. Precisely inspired by K-means clustering framework [13] and kernel-based GL methods [14], we propose an iterative approach for joint Multiple Kernel-based GL and clustering of graph signals, named **KMGL**[1].

To the best of our knowledge, this is the first work that considers side information in multiple GL and signal clustering and also obtains the denoised version of the graph signals. We utilize the efficient block coordinate descent scheme (BCD) [15] by linking the GL task to the well-studied least squares problem. Previous graph signal clustering methods [11], [16], [17] were not designed to use kernel-based spaces and do not consider node-side covariates. Besides, they work with noiseless signals, a serious practical limitation.

II. GSP Preliminaries

Denote $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ as a weighted undirected graph without self-loops where $\mathcal{V}$ is the node set with $|\mathcal{V}| = n$, the edge set $\mathcal{E}$ is a subset of $\mathcal{V} \times \mathcal{V}$, and $\mathbf{W}$ is the symmetric adjacency matrix. The entity W_{ij} has a positive value if there is an edge between vertices v_i and v_j but zero otherwise. Let $\mathbf{D} = diag(\mathbf{W}\mathbf{1})$ be the diagonal degree matrix where $\mathbf{1}$ is the all-one vector. The Laplacian matrix of $\mathcal{G}$ given by $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is symmetric and positive semi-definite with the eigendecomposition of $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ [2]. Then, the Graph Fourier Transform (GFT) is defined in terms of $\mathbf{U}$. Formally, a graph signal is a function $x : V \to \mathbb{R}$ isomorphic to $\mathbb{R}^n$. Then, the GFT of $\mathbf{x}$ is given by $\hat{\mathbf{x}} = \mathbf{U}^\top \mathbf{x}$ where $\hat{x}_i$ is the spectral component of the ith eigenvector [2]. A graph signal is smooth if connected nodes with a larger weight have more similar values [6]. This is measured via the Laplacian's quadratic form

$$\mathbf{x}^\top \mathbf{L}\mathbf{x} = \sum_{i,j} W_{ij}(x_i - x_j)^2. \tag{1}$$

M. H. Alizade (corresponding author, e-mail: mohamad.h.alizade@ut.ac.ir) is with the Electrical Engineering department of the Tehran University, Iran.

A. Einizade is with the Electrical Engineering Department of the Sharif University of Technology, Iran.

[1] https://github.com/mohamad-h-alizade/KMGL

Equivalently, restricting the graph signal's energy to the first few spectral components achieves the same definition:

$$\mathbf{x}^\top \mathbf{L}\mathbf{x} = \mathbf{x}^\top \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top \mathbf{x} = \hat{\mathbf{x}}^\top \mathbf{\Lambda}\hat{\mathbf{x}} = \sum_i \lambda_i \hat{x}_i^2, \tag{2}$$

where $\lambda_i = \Lambda_{ii}$ (i.e., the ith eigenvalue of $\mathbf{L}$) has a frequency-like interpretation [4]. With this notion of frequency, $\hat{y}_i = h(\lambda_i)\hat{x}_i$ forms a graph filter that either amplifies or attenuates each spectral component. The filtered graph signal

$$\mathbf{y} = \mathbf{U}h(\mathbf{\Lambda})\mathbf{U}^\top \mathbf{x} = h(\mathbf{L})\mathbf{x} \tag{3}$$

can be characterized as applying a linear operator in terms of $\mathbf{L}$. Common choices for low-pass filtering includes $\mathbf{L}^{-1/2}$ or $(\mathbf{I}+\gamma\mathbf{L})^{-1}$, and conversely $\mathbf{L}$ for high-pass filtering [2], [3].

III. Kernel Multiple Graph Learning

A. Problem Statement

We are given a (normalized) dataset $\mathcal{X} = \{\mathbf{x}_i \in \mathbb{R}^n\}_{i=1}^m$ of graph signals residing on the shared vertex set $\mathcal{V}$ and a set of node-side information $\mathcal{P} = \{p_1, \cdots, p_n\}$ with p_i denoting a prior covariate for $v_i \in \mathcal{V}$. The objective is to partition $\mathcal{X}$ into K clusters $\{\mathcal{X}_k\}_{k=1}^K$ (where $|\mathcal{X}_k| = m_k$) and learn their graphs $\{\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_k, \mathbf{W}_k)\}_{k=1}^K$ that best fit their partition in terms of smoothness and the node-side information.

To this end, kernels [18] are selected to separate the representation of information from the algorithm. The selection of kernels also permits the promotion of different features in each cluster and consequently captures various types of relationships in each graph. Additionally, this allows us to implicitly transform each node-side information to a high-dimensional feature vector without conducting any direct computation.

Let $\mathcal{K} : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ be a symmetric positive definite kernel on the node-side information. Based on the Aronszajn theorem [19] there is a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ and a feature map $\phi : \mathcal{P} \to \mathcal{H}$ such that $\mathcal{K}(p_i, p_j) = \langle \phi(p_i), \phi(p_j) \rangle_{\mathcal{H}}$. This maps (possibly infinite) features to each node. The set of node features

$$\mathcal{F} = \{\mathbf{f} \in \mathbb{R}^n | f_i = \phi(p_i)_d, d \in dims(\mathcal{H})\} \tag{4}$$

represents all the n-dimensional feature vectors associated with the kernel. The graph signal $\mathbf{x}$ is expected to match the node-side information and may not deviate from the set of points in $\mathcal{F}$. Thus, the first few principal components of $\mathcal{F}$ approximate $\mathbf{x}$ well. The kernel matrix $[\mathbf{K}]_{ij} = \mathcal{K}(p_i, p_j)$ is the sample covariance of features and its eigenvectors capture these components, then the deviation is evaluated via $H(\mathbf{K})\mathbf{x} = \mathbf{K}^{-1}\mathbf{x}$ for each node. This transformation gets larger when the graph signal has a significant projection in a direction that the feature set is less spread, i.e., a small eigenvalue of $\mathbf{K}$. Alternatively, $H(\mathbf{K})$ can be viewed as a filter based on $\mathcal{P}$ that amplifies the signal in atypical directions that the feature set is spread.

The fitness of a graph signal $\mathbf{x}$ to the underlying graph $\mathcal{G}$ and the node-side information $\mathcal{P}$ is measured via applying a filter in terms of $\mathbf{L}$ and $\mathbf{K}$, then comparing the two signals. The following similarity function is used

$$s(\mathbf{x}, \hat{\mathbf{x}}) = \mathbf{x}^\top \hat{\mathbf{x}} \tag{5}$$

where $\hat{\mathbf{x}}$ is the filtered (denoised) version of $\mathbf{x}$ such that

$$\mathbf{x} - \hat{\mathbf{x}} = \alpha\mathbf{K}^{-1}\hat{\mathbf{x}} + \beta\mathbf{L}\hat{\mathbf{x}} \tag{6}$$

$$\Rightarrow (\mathbf{I} + \alpha\mathbf{K}^{-1} + \beta\mathbf{L})^{-1}\mathbf{x} = lp(\mathbf{K}, \mathbf{L})\mathbf{x} = \mathbf{S}\mathbf{x} = \hat{\mathbf{x}} \tag{7}$$

for some positive α and β. Eq. (6) states that the difference between the two lies in the linear cone of $\mathbf{K}^{-1}\hat{\mathbf{x}}$ and $\mathbf{L}\hat{\mathbf{x}}$. The former means $\hat{\mathbf{x}}$ aligns more with the prior information, and the latter means it is smoother on the underlying graph. Thus, the filter $\mathbf{S}$ has a low-pass behavior on the combination of the $\mathcal{G}$ and $\mathcal{P}$. Specifically, it becomes a typical low-pass graph filter when $\alpha = 0$.

We propose to jointly cluster the graph signals and learn multiple graphs consistently with prior node-side information. We name our algorithm Kernel Multiple GL (**KMGL**) which expresses the problem as finding the partition sets $\{\mathcal{X}_k\}_{k=1}^K$ and the Laplacian matrices $\{\mathbf{L}_k\}_{k=1}^K$ as follows

$$\begin{aligned} \max_{\{\mathcal{X}_k, \mathbf{L}_k \in \mathcal{L}\}_{k=1}^K, \{\hat{\mathbf{x}}_i\}_{i=1}^m} \quad & \sum_{k=1}^K \sum_{\mathbf{x} \in \mathcal{X}_k} \mathbf{x}^\top \hat{\mathbf{x}} - \gamma ||\mathbf{L}_k||_F^2 \\ \text{s.t.} \quad & \mathbf{x} - \hat{\mathbf{x}} = \alpha\mathbf{K}_k^{-1}\hat{\mathbf{x}} + \beta\mathbf{L}_k\hat{\mathbf{x}} \quad \forall \mathbf{x} \in \mathcal{X}_k, \forall k \\ & Tr(\mathbf{L}_k) = n \quad \forall k \end{aligned} \tag{8}$$

where $\mathcal{L}$ is the set of valid graph Laplacians

$$\mathcal{L} = \{\mathbf{L} \in \mathbb{R}^{n \times n} | \mathbf{L} = \mathbf{L}^\top, \mathbf{L}\mathbf{1} = \mathbf{0}, L_{ij} \leq 0 \ \ \forall i \neq j\}, \tag{9}$$

and $Tr(.)$ is the trace operator. The first term in Eq. (8) promotes the similarity of the graph signal and its filtered version. This selects for graphs that the low-pass filter in the first constraint affects their cluster the least. The hyperparameters α and β control the trade-off between matching the side information and the smoothness of graph signals, respectively. The second term of the objective regularizes the graphs to have a smaller Frobinius norm. Combined with the second constraint, they affect the sparsity of the learned graphs and avoid trivial solutions. Graphs are sparser as $\gamma \in \mathbb{R}_+$ gets larger. Other regularization terms, e.g., log-barrier [10], are also commonly used to impose connectedness.

B. Algorithm

The problem formulated in Eq. (8) is NP-hard. This is because the selection of partitions affects the optimal graphs and, consequently, the utility function. To avoid solving the problem for every possible partitioning, an iterative solution similar to K-means [13] is proposed that increases the utility at each step. The algorithm first partitions the dataset randomly and then iterates between two steps: 1) Learning a graph for each cluster, and 2) Reassigning the graph signals. This is repeated until the partitions remain the same.

Firstly, for the kth cluster, given the initial (and possibly noisy) graph signals $\{\mathbf{x} \in \mathcal{X}_k\}$ and by fixing their assignments, we solve the following optimization for learning their associated filtered versions and the kth graph:

$$\begin{aligned} \{\mathbf{L}_k, \hat{\mathcal{X}}_k\} = & \operatorname*{argmax}_{\mathbf{L}_k \in \mathcal{L}, \hat{\mathbf{x}} \in \hat{\mathcal{X}}_k} \sum_{\mathbf{x} \in \mathcal{X}_k} \mathbf{x}^\top \hat{\mathbf{x}} - \gamma ||\mathbf{L}_k||_F^2 \\ \text{s.t.} \ \ \forall \mathbf{x} \in \mathcal{X}_k : \ \ & \mathbf{x} - \hat{\mathbf{x}} = \alpha\mathbf{K}_k^{-1}\hat{\mathbf{x}} + \beta\mathbf{L}_k\hat{\mathbf{x}}, \ \ Tr(\mathbf{L}_k) = n, \end{aligned} \tag{10}$$

where the set $\hat{\mathcal{X}}_k$ contains the filtered versions of graph signals for $\mathcal{X}_k$. Although it is less obvious, the above problem can be formulated more familiarly as follows:

Theorem 1. *The maximization problem in Eq. (10) is equivalent to a joint kernel ridge regression from set $\mathcal{P}$ to $\mathcal{X}_k$ and a GL problem.*

Proof. We show that our problem is equivalent to

$$\begin{aligned}\min_{\mathbf{L}_k \in \mathcal{L}, \mathbf{c} \in \mathcal{C}_k} \sum_{\mathbf{x} \in \mathcal{X}_k} & ||\mathbf{x} - \mathbf{K}_k \mathbf{c}||_2^2 + \alpha \mathbf{c}^\top \mathbf{K} \mathbf{c} \\ & + \beta \mathbf{c}^\top \mathbf{K} \mathbf{L}_k \mathbf{K} \mathbf{c} + \gamma ||\mathbf{L}_k||_F^2 \\ \text{s.t.} \quad & Tr(\mathbf{L}_k) = n \end{aligned} \tag{11}$$

where for any $\mathbf{c} \in \mathcal{C}_k$ there is a filtered signal $\hat{\mathbf{x}} \in \hat{\mathcal{X}}_k$ such that $\hat{\mathbf{x}} = \mathbf{K}_k \mathbf{c}$ as proposed in [14]. We start by writing Eq. (11) in terms of the estimated signals $\hat{\mathbf{x}}$. Note that since $\mathbf{K}$ is positive definite $\mathbf{c}^\top \mathbf{K} \mathbf{c} = \hat{\mathbf{x}}^\top \mathbf{K}^{-1} \hat{\mathbf{x}}$ and we have

$$\begin{aligned}\min_{\mathbf{L}_k \in \mathcal{L}, \hat{\mathcal{X}}_k} \sum_{\mathbf{x} \in \mathcal{X}_k} & ||\mathbf{x} - \hat{\mathbf{x}}||_2^2 + \alpha \hat{\mathbf{x}}^\top \mathbf{K}^{-1} \hat{\mathbf{x}} + \beta \hat{\mathbf{x}}^\top \mathbf{L}_k \hat{\mathbf{x}} + \gamma ||\mathbf{L}_k||_F^2 \\ \text{s.t.} \quad & Tr(\mathbf{L}_k) = n. \end{aligned} \tag{12}$$

The above problem is separable in each $\hat{\mathbf{x}}$ and we start by minimizing over them

$$\hat{\mathbf{x}} = \underset{\hat{\mathbf{x}}}{\operatorname{argmin}} ||\mathbf{x} - \hat{\mathbf{x}}||_2^2 + \alpha \hat{\mathbf{x}}^\top \mathbf{K}^{-1} \hat{\mathbf{x}} + \beta \hat{\mathbf{x}}^\top \mathbf{L}_k \hat{\mathbf{x}} \tag{13}$$

that is convex and differentiable with a gradient of

$$-(\mathbf{x} - \hat{\mathbf{x}}) + \alpha \mathbf{K}_k^{-1} \hat{\mathbf{x}} + \beta \mathbf{L}_k \hat{\mathbf{x}}. \tag{14}$$

Putting the gradient zero results in the Eq. (7). Then, the following can be substituted in Eq. (16)

$$\begin{aligned} & ||\mathbf{x} - \hat{\mathbf{x}}||_2^2 + \alpha \hat{\mathbf{x}}^\top \mathbf{K}^{-1} \hat{\mathbf{x}} + \beta \hat{\mathbf{x}}^\top \mathbf{L}_k \hat{\mathbf{x}} \\ & = ||\mathbf{x} - \hat{\mathbf{x}}||_2^2 + \hat{\mathbf{x}}^\top (\mathbf{x} - \hat{\mathbf{x}}) = \mathbf{x}^\top (\mathbf{x} - \hat{\mathbf{x}}) = ||\mathbf{x}||_2^2 - \mathbf{x}^\top \hat{\mathbf{x}} \end{aligned}$$

to have

$$\begin{aligned}\min_{\mathbf{L}_k \in \mathcal{L}, \hat{\mathcal{X}}_k} \sum_{\mathbf{x} \in \mathcal{X}_k} & ||\mathbf{x}||_2^2 - \mathbf{x}^\top \hat{\mathbf{x}} + \gamma ||\mathbf{L}_k||_F^2 \\ \text{s.t.} \quad & \forall \mathbf{x} \in \mathcal{X}_k : \quad \mathbf{x} - \hat{\mathbf{x}} = \alpha \mathbf{K}_k^{-1} \hat{\mathbf{x}} + \beta \mathbf{L}_k \hat{\mathbf{x}} \\ & Tr(\mathbf{L}_k) = n. \end{aligned} \tag{15}$$

The term $||\mathbf{x}||_2^2$ has no bearing on the minimization, and a sign change in objective results in the maximization (10). □

Based on Theorem 1 and the approach in [14] we solve Eq. (10) by a block-coordinate decent of line (12) that iteratively filters the graph signals by (7) and then solves for

$$\begin{aligned}\mathbf{L}_k = \underset{\mathbf{L}_k \in \mathcal{L}}{\operatorname{argmin}} & \sum_{\forall \hat{\mathbf{x}} \in \hat{\mathcal{X}}_k} \beta \hat{\mathbf{x}}^\top \mathbf{L}_k \hat{\mathbf{x}} + \gamma ||\mathbf{L}_k||_F^2 \\ \text{s.t.} \quad & Tr(\mathbf{L}_k) = n \end{aligned} \tag{16}$$

which is a GL problem in the typical Laplacian quadratic form and can be solved by convex optimization techniques.

In the second step of the algorithm, we refine the partitions by fixing the graphs and assigning each graph signal to the most compatible cluster. Let $\hat{\mathbb{X}}(\mathbf{x}) = \{\hat{\mathbf{x}}_k | \hat{\mathbf{x}}_k = lp(\mathbf{K}_k, \mathbf{L}_k)\mathbf{x}, k = 1, \ldots, K\}$ be the set of filtered graph signals of $\mathbf{x}$ over all graphs. We specify the assignment of $\mathbf{x}$, i.e., $i(\mathbf{x})$, as follows:

$$i(\mathbf{x}) = \underset{k:\, \hat{\mathbf{x}}_k \in \hat{\mathbb{X}}(\mathbf{x})}{\operatorname{argmax}} \quad \mathbf{x}^\top \hat{\mathbf{x}}_k. \tag{17}$$

Then, the partitions are refined such that $\mathcal{X}_k = \{\mathbf{x} \in \mathcal{X} | i(\mathbf{x}) = k\}$ for $k = 1, ..., K$. The proposed kernel multiple GL (**KMGL**) algorithm is summarized in Algorithm 1.

Algorithm 1 KMGL

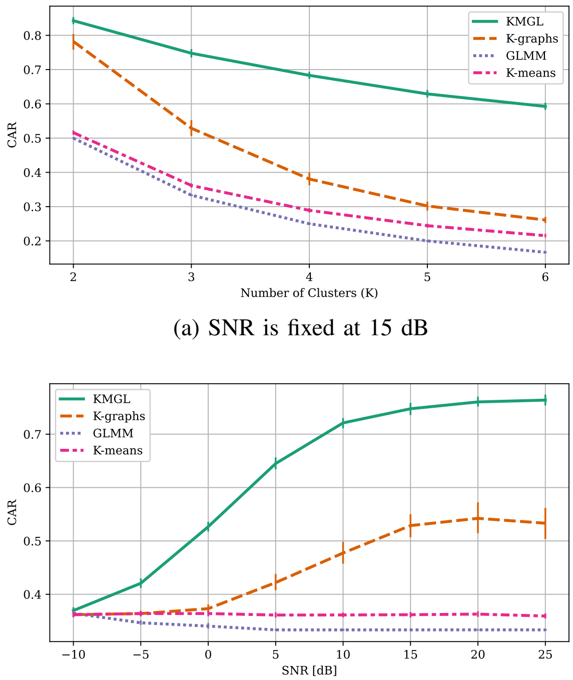
```
Input: Graph signals $\mathcal{X}$, number of clusters $K$, Kernel matrices $\{\mathbf{K}_k\}_{k=1}^K$, hyperparameters $\alpha, \beta, \gamma$, tolerance $\epsilon$
Output: Partition set $\{\mathcal{X}_k\}_{k=1}^K$, graph Laplacians $\{\mathbf{L}_k\}_{k=1}^K$, filtered signals $\{\forall \mathbf{x} \in \mathcal{X} : \hat{\mathbf{x}}\}$
Initialization: Randomly partition $\mathbf{X}$ into $K$ clusters.
repeat
    for $k = 1 : K$ do                          ▷ GL and filtering (denoising)
        repeat
            Filter every $\mathbf{x} \in \mathcal{X}_k$ with $lp(\mathbf{K}_k, \mathbf{L}_k)$ in Eq. (7)
            Learn the graph Laplacian $\mathbf{L}_k$ via Eq. (16)
        until $\mathbf{L}_k$ converges: $||\Delta L_k||_F \leq \epsilon$
    end for
    for $\mathbf{x} \in \mathcal{X}$ do                        ▷ Refining the clusters
        Update the cluster of $\mathbf{x}$ via Eq. (17)
    end for
until partitions are unchanged
```

Theorem 2. *The KMGL algorithm converges in a finite number of iterations.*

Proof. For each algorithm step, the objective function of (8) is non-decreasing. This is because in the first step when the graphs are updated via (10), the maximization is over the same terms, and thus the new graphs would not decrease the objective. Precisely, the optimization 10 is biconvex [14], and therefore, utilizing BCD reaches unique solutions for each subproblem which guarantees to reach a stationary point [15]. In the second step, when the clusters are refined, we directly increase the term $\mathbf{x}^\top \hat{\mathbf{x}}$ for each graph signal, or it remains the same. Since $||\mathbf{L}_k||_F^2$ is fixed, the objective is also non-decreasing here. Lastly, there are finite assignments of m graph signals to K clusters, and consequently, the algorithm has to converge. □

IV. Experiments

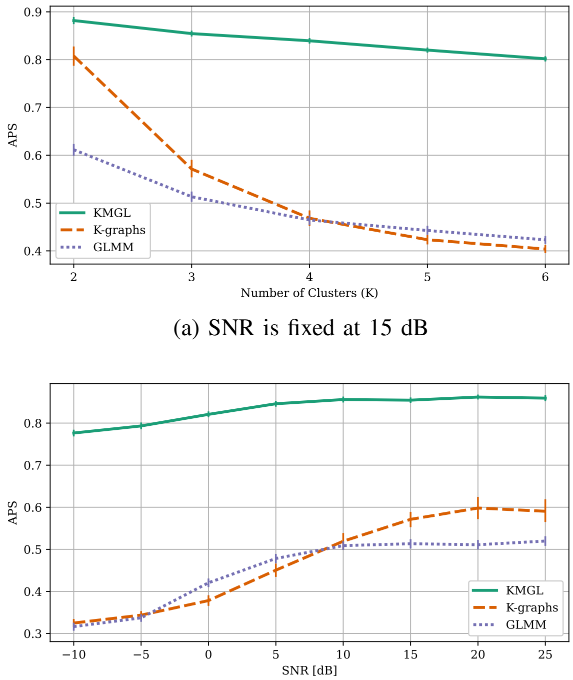
In this section, the performance of the KMGL algorithm on synthetic data is evaluated and compared to the GLMM [11] and K-graphs [16]. We draw random Erdos–Renyi graphs of n nodes with a (binary) connection probability of $pr((v_i, v_j) \in \mathcal{E}) = 0.3$ for $i \neq j$. Edge weights are normalized such that the sum of weights is n. Similar to [14], graph signals of the kth cluster are generated according to $\forall \mathbf{x} \in \mathcal{X}_k : \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_k + \sigma_\epsilon \mathbf{I})$, where $\mathbf{K}_k = (\mathbf{I} + \eta \mathbf{L}_k)^{-1}$ is the kernel matrix corresponding to the kth cluster and σ_ϵ is the noise-level. This common choice of kernel leads to globally smooth signals [14]. We select $\eta = 10$ for the experiments. The models



(a) SNR is fixed at 15 dB

(b) Results for 3 clusters

Fig. 1: Clustering performance of KMGL compared to K-graphs, GLMM, and K-means based on CAR when (a) K and (b) SNR increases. $\alpha = \beta = 10^{-2}$, $\gamma = 10^{-4}$.



(a) SNR is fixed at 15 dB

(b) Results for 3 clusters

Fig. 2: GL performance of KMGL compared to K-graphs, and GLMM based on APS when (a) K and (b) the SNR increases. $\alpha = \beta = 10^{-2}$, $\gamma = 10^{-4}$.

are examined on different σ_ϵ and number of clusters (K). We select the noise level such that the signal-to-noise ratio

$$SNR = 10\log_{10}\left(\frac{1}{Kn\sigma_\epsilon^2}\sum_{c=1}^{K} Tr(\mathbf{K}_c)\right) \tag{18}$$

is varied uniformly.

Models are evaluated based on their clustering performance and the quality of learned graphs. The former is measured via clustering accuracy ratio (CAR), which finds the best map between the cluster indices of samples and the partitions, then measures the number of correctly clustered signals to their total number [16]. The latter is evaluated by average precision score (APS), where the ability of the model to detect the presence of edges is considered. APS automatically varies the threshold that weights are declared present and then compares them with the ground truth. [14]

The datasets have $m = 500$ signals, sampled equally from $K \in \{2 : 5\}$ different graphs with $n = 20$ nodes. We perform preliminary tests for each model to tune the hyperparameters and then keep them intact through the experiments. Furthermore, we restrict our model so that $\alpha = \beta$ equally prioritizes smoothness and side information. In the subsequent figures, each data point consists of 50 independent realizations. The compared models are applied 10 times for each realization and are evaluated on their best try, determined via their objective function, while the proposed algorithm is applied only once.

Fig. 1 displays the effective clustering performance of the KMGL algorithm. Part (a) shows that the number of clusters less affects the proposed algorithm, and (b) shows its robustness against noise compared to the state-of-the-art. Fig. 2 shows the ability of models to recover and learn the graphs. Similar to Fig. 1, KMGL outperforms the compared models both in part (a) when the number of clusters is increased and (b) when the noise-rejection behavior is studied. It is worth mentioning that the ability of the model to recover the graphs even in high noise levels was also seen in [14] and further shows the benefits of incorporating side information. As shown in the figures, K-graphs perform more robustly than the GLMM, probably due to utilizing graph signal smoothness. The k-means algorithm is also added as a baseline to represent a model without knowledge of the underlying graphs.

V. Conclusion

In this letter, we proposed the KMGL algorithm to incorporate node-side information in clustering graph signals and learning multiple graphs. The node-side information was represented via kernels, and we offered a filter-based framework to model its relation with the data and the underlying graphs.

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, 2013.

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

[3] G. Leus, A. G. Marques, J. M. Moura, A. Ortega, and D. I. Shuman, "Graph signal processing: History, development, impact, and outlook," *IEEE Signal Processing Magazine*, vol. 40, no. 4, pp. 49–60, 2023.

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

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

[6] 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, 2016.

[7] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[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] R. Shafipour, S. Segarra, A. G. Marques, and G. Mateos, "Identifying the topology of undirected networks from diffused non-stationary graph signals," *IEEE Open Journal of Signal Processing*, vol. 2, pp. 171–189, 2021.

[10] V. Kalofolias, "How to learn a graph from smooth signals," in *Artificial intelligence and statistics*, pp. 920–929, PMLR, 2016.

[11] H. P. Maretic and P. Frossard, "Graph laplacian mixture model," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 261–270, 2020.

[12] W. Huang, T. A. Bolton, J. D. Medaglia, D. S. Bassett, A. Ribeiro, and D. Van De Ville, "A graph signal processing perspective on functional brain imaging," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 868–885, 2018.

[13] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Information Sciences*, 2022.

[14] X. Pu, S. L. Chau, X. Dong, and D. Sejdinovic, "Kernel-based graph learning from smooth signals: A functional viewpoint," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 7, pp. 192–207, 2021.

[15] A. Beck and L. Tetruashvili, "On the convergence of block coordinate descent type methods," *SIAM journal on Optimization*, vol. 23, no. 4, pp. 2037–2060, 2013.

[16] H. Araghi, M. Sabbaqi, and M. Babaie-Zadeh, "k-graphs: An algorithm for graph signal clustering and multiple graph learning," *IEEE Signal Processing Letters*, vol. 26, no. 10, pp. 1486–1490, 2019.

[17] A. Karaaslanli and S. Aviyente, "Simultaneous graph signal clustering and graph learning," in *International Conference on Machine Learning*, pp. 10762–10772, PMLR, 2022.

[18] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," 2008.

[19] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American mathematical society*, vol. 68, no. 3, pp. 337–404, 1950.

[20] S. K. Narang, A. Gadde, E. Sanou, and A. Ortega, "Localized iterative methods for interpolation in graph structured data," in *2013 IEEE Global Conference on Signal and Information Processing*, pp. 491–494, IEEE, 2013.

Appendix

A. Clustering and Learning with Missing Data

In this section, we extend the KMGL algorithm to support partially observed graph signals. Specifically, we change how each graph signal is filtered by first recovering its missing values. To this end, we adopt an iterative approach for graph signal reconstruction. The asymptotic behavior of the resulting algorithm is studied and we conduct numerical experiments to measure its effectiveness.

Let $\mathcal{O}$ denote the observed subspace of the graph signal $\mathbf{x}$. Then, the downsampling operator $\mathbf{J} : \mathbb{R}^n \rightarrow \mathcal{O}$ maps the partially observed signal to this space. Moreover, $\mathbf{M} = \mathbf{J}^T\mathbf{J}$ represents the downsampling and then upsampling operation where $\mathbf{M}$ is a diagonal masking matrix with $M_{ii} = 1$ if we observe the ith component of $\mathbf{x}$ and zero otherwise. $\mathbf{Mx}$ interpolates the missing values of $\mathbf{x}$ with zeroes, however, this incorporates neither the graph structure nor the side information.

To extend the KMGL algorithm, we need to revisit how each graph signal $\mathbf{x}$ relates to its filtering $\hat{\mathbf{x}}$. Previously, the low-pass filter $\mathbf{S}$ related the two in Eq. (7). Since $\mathbf{x}$ is partially observed,

$$\begin{aligned} \mathbf{x}^1 &= \mathbf{Mx} \\ \hat{\mathbf{x}}^t &= \mathbf{Sx}^t \\ \mathbf{x}^{t+1} &= \hat{\mathbf{x}}^t + \mathbf{M}(\mathbf{x}^1 - \hat{\mathbf{x}}^t) \end{aligned} \tag{19}$$

iteratively recovers its missing values as suggested in [20]. The first line initially interpolates $\mathbf{x}$ with zero. The second line applies a low-pass filter to update the missing nodes based on the others. The structure of the graph and the side information dictate how missing values relate to others. The last line ensures that $\mathbf{x}^t$ remains unchanged in the observed nodes. Lastly, the similarity of the reconstructed graph signal $\mathbf{x}^t$ with its low-passed filtering $\hat{\mathbf{x}}^t$ is compared in $\mathcal{O}$:

$$s(\mathbf{x}, \hat{\mathbf{x}}^t) = (\mathbf{Jx}^t)^T(\mathbf{J}\hat{\mathbf{x}}^t) = \mathbf{x}^T\mathbf{M}\hat{\mathbf{x}}^t. \tag{20}$$

Theorem 3. *As the number of iterations t increases, $\hat{\mathbf{x}}_t$ has the asymptotic solution of* $(\mathbf{M} + \alpha\mathbf{K}^{-1} + \beta\mathbf{L})^{-1}\mathbf{Mx} = \hat{\mathbf{x}}$.

Proof. The iterations in (19) converge to a fixed point if $\mathbf{S}$ is a non-expansive operator [20]. since $\alpha\mathbf{K}^{-1} + \beta\mathbf{L}$ is positive definite, all the eigenvalues of $\mathbf{S}$ have the absolute value of less than one. Consequently, $\mathbf{S}$ is non-expansive and $\hat{\mathbf{x}}_t$ converges. In the converged point $\hat{\mathbf{x}} = \hat{\mathbf{x}}^t = \hat{\mathbf{x}}^{t+1}$ we have

$$\begin{aligned} &\hat{\mathbf{x}} = \mathbf{S}(\hat{\mathbf{x}} + \mathbf{M}(\mathbf{x}_1 - \hat{\mathbf{x}})) = \mathbf{S}(\hat{\mathbf{x}} + \mathbf{Mx} - \mathbf{M}\hat{\mathbf{x}}) \Rightarrow \\ &\mathbf{M}(\mathbf{x} - \hat{\mathbf{x}}) = \mathbf{S}^{-1}\hat{\mathbf{x}} - \hat{\mathbf{x}} = \alpha\mathbf{K}^{-1}\hat{\mathbf{x}} + \beta\mathbf{L}\hat{\mathbf{x}} \Rightarrow \\ &(\mathbf{M} + \alpha\mathbf{K}^{-1} + \beta\mathbf{L})^{-1}\mathbf{Mx} = \hat{\mathbf{x}} \end{aligned} \tag{21}$$

which completes the proof. □

We remark that Eq. (21) also results from solving the following convex problem as shown in [14],

$$\min_{\hat{\mathbf{x}}} ||\mathbf{M}(\mathbf{x} - \hat{\mathbf{x}})||_2^2 + \alpha\hat{\mathbf{x}}^T\mathbf{K}^{-1}\hat{\mathbf{x}} + \beta\hat{\mathbf{x}}^T\mathbf{L}\hat{\mathbf{x}}. \tag{22}$$

To summarize, the Alg. 1 is generalized to support partially observed graph signals as follows. In line 5, every signal is now filtered via Eq. (21) where $\mathbf{M}$ is the diagonal masking matrix associated with $\mathbf{x}$. The Laplacian and the kernel also relate to each signal's cluster. In line 10, the signals are reassigned based on

$$i(\mathbf{x}) = \underset{k:\, \hat{\mathbf{x}}_k \in \hat{\mathbb{X}}(\mathbf{x})}{\operatorname{argmax}} \quad \mathbf{x}^T\mathbf{M}\hat{\mathbf{x}}_k \tag{23}$$

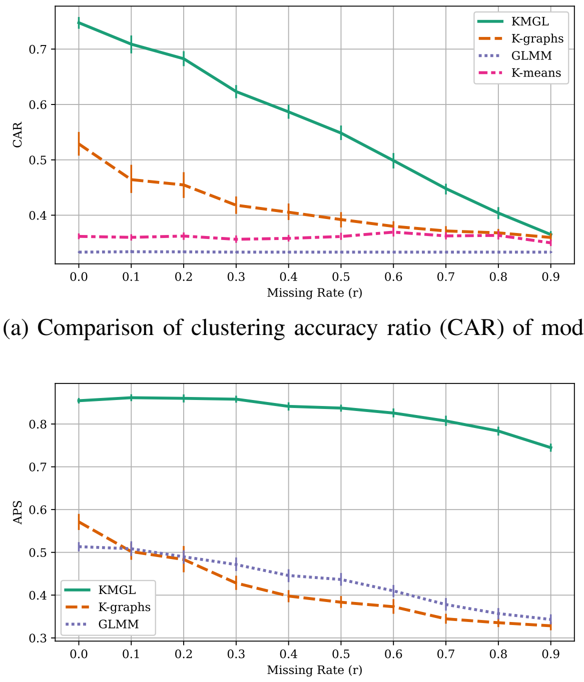
where

$$\hat{\mathbb{X}} = \{\hat{\mathbf{x}}_k | (\mathbf{M} + \alpha\mathbf{K}_k^{-1} + \beta\mathbf{L}_k)^{-1}\mathbf{Mx} = \hat{\mathbf{x}}_k, 1 \leq k \leq |\mathcal{C}|\}$$

is similar to the previous set but with the filters in Eq. (21).

The rest of this section discusses the validity of the proposed algorithm in numerical experiments. Each entry of each graph signal has a $(1 - r)$ probability of missing where r is the missing rate. Hence, the element of the diagonal mask matrix is generated via $M_{ii} \sim Bernoulli(1 - r)$. For the compared models, the missing values are set to their statistical expectation, i.e. zero, as this is a natural and unbiased setting in practice.

The performance of models is evaluated based on their clustering accuracy ratio (CAR) and average precision score (APS) of the recovered graphs. The metrics are plotted against the missing rate as it changes uniformly between zero and one. Fig. 3 summarizes the results and shows the higher robustness of KMGL. Each data point in the figures is the average of 20 independent realizations with different graphs and data.



(a) Comparison of clustering accuracy ratio (CAR) of models

(b) Comparison of average precision score (APS) of models

Fig. 3: Performance of KMGL algorithm on partially observed signals compared to K-graphs, GLMM, and K-means in terms of (a) clustering and (b) learning graphs. SNR=15 and 3 clusters.