Learning Multi-layer Graphs and a Common Representation for Clustering

Sravanthi Gurugubelli and Sundeep Prabhakar Chepuri
Indian Institute of Science, Bangalore, India

***Abstract*—In this paper, we focus on graph learning from multi-view data of shared entities for spectral clustering. We can explain interactions between the entities in multi-view data using a multi-layer graph with a common vertex set, which represents the shared entities. The edges of different layers capture the relationships of the entities. Assuming a smoothness data model, we jointly estimate the graph Laplacian matrices of the individual graph layers and low-dimensional embedding of the common vertex set. We constrain the rank of the graph Laplacian matrices to obtain multi-component graph layers for clustering. The low-dimensional node embeddings, common to all the views, assimilate the complementary information present in the views. We propose an efficient solver based on alternating minimization to solve the proposed multi-layer multi-component graph learning problem. Numerical experiments on synthetic and real datasets demonstrate that the proposed algorithm outperforms state-of-the-art multi-view clustering techniques.**

***Index Terms*—Clustering, graph learning, multi-layer graphs, multi-view data, representation learning.**

I. Introduction

We often observe complementary information about a common source from multiple modalities or through different feature subsets in data analysis. Various aspects of interactions underlying multi-view datasets can be represented using multi-layer graphs, in which each graph layer represents a different view. All the graph layers in a multi-layer graph share the same set of nodes representing the shared entities. However, the edges in distinct layers represent the interactions between the entities in that view [1]. For example, in a social network graph, we can think of people's interests like favorite sport, affiliation, and hobbies as multiple views of people in the network [2]. Each of these views can be represented by a graph giving rise to a multi-layer graph with each layer consisting of the same set of nodes representing people. Also, one can have data from multiple modalities. For example, in brain imaging and analysis, data from different modalities, like functional magnetic resonance imaging, structural magnetic resonance imaging and diffusion tensor imaging [3], may be considered as signals on a multi-layer graph. The edges in different graph layers represent interactions between the same set of brain regions observed in different modalities [4,5].

The underlying graph structure is leveraged to solve several signal processing and machine learning tasks like denoising, spectral clustering, and dimensionality reduction. However, the underlying graph may not always be readily available. Learning graphs from data is an ill-posed problem. Although nearest neighbor graphs, correlation graphs, or Gaussian similarity kernels learnt from data are simple and commonly used choices, they are sensitive to noise or missing samples. Hence, several graph learning solutions that carefully model the data (e.g., using a smoothness or probabilistic graphical model) and incorporate prior information (such as sparsity, product, or multi-component structure) about the resulting topology have been proposed [6]–[13]. Although the existing graph learning methods can be directly used to learn the individual graph layers of a multi-layer graph, the information contained in all the views cannot be captured by any of the individual graph layers. Therefore, we propose multi-layer graph learning from multi-view data in this paper.

Low-dimensional embeddings of the nodes of a graph that encode the structural information about the graph are useful in graph-based spectral clustering and other graph analysis tasks [14,15]. A common technique to compute these low-dimensional node embeddings is to perform a partial eigenvalue decomposition of the graph underlying the data. When dealing with multi-layer graphs, it is necessary to find low-dimensional embeddings of the nodes common to all the views that best capture the complementary information available in different views. Assuming that the individual graph layers of a multi-layer graph are available, there are subspace learning methods that learn a common subspace by merging the multiple subspaces computed from the individual graph layers [16,17]. In the context of multi-view canonical correlation analysis (MCCA), [18] learns a common subspace that is smooth on a known graph from multi-view data.

Instead, in this paper, we jointly learn a multi-layer graph and common low-dimensional node embeddings from multi-view data for multi-view clustering. Specifically, given M views of data, we estimate a graph with M layers having K components. The respective views of data are smooth on the graph topology of individual layers that we learn. The graph is restricted to a K-component graph to cluster data in K groups by imposing rank constraints on the graph Laplacian matrices of the individual graph layers. We propose an efficient solver based on alternating minimization, each subproblem of which is solved optimally. We evaluate the performance of the proposed method for clustering on synthetic and real-world datasets and compare them with state-of-the-art techniques for multi-view clustering [16]–[18]. The results show that our algorithm outperforms single-view clustering that ignores information from different views and state-of-the-art multi-

This work was supported in part by the Pratiksha Trust Fellowship and SERB grant SRG/2019/000619.

view clustering methods.

Notation: Throughout the paper, we use boldface lowercase (respectively, uppercase) to denote column vectors (respectively, matrices). The operator $\mathrm{diag}(.)$ takes a vector as input and returns a square diagonal matrix with the elements of the vector on its main diagonal. The set of positive semi-definite matrices of size $N \times N$ is denoted by $\mathbb{S}_+^N$. $\odot$ and $\otimes$ denote the elementwise Hadamard product and the Kronecker product, respectively. $\mathrm{vec}(\cdot)$ denotes the matrix vectorization operator.

II. Multi-layer graphs

Consider a M-layer weighted and undirected graph $\mathcal{G}$ with individual graph layers $\mathcal{G}_m = \{\mathcal{V}, \mathbf{W}_m\}$, $m = 1, 2, \ldots, M$, where $\mathcal{V} = \{v_1, \cdots v_N\}$ denotes the common vertex (or node) set with $N = |\mathcal{V}|$ nodes and $\mathbf{W}_m \in \mathbb{R}^{N \times N}$ denotes the weighted adjacency matrix of the mth graph layer $\mathcal{G}_m$. The (i, j)th element of $\mathbf{W}_m$ contains a positive edge weight if two nodes v_i and v_j are connected in $\mathcal{G}_m$ and is zero otherwise. The adjacency matrix $\mathbf{W}_m$ is symmetric as $\mathcal{G}_m$ is undirected. The degree of the nodes in $\mathcal{G}_m$ is defined as $\mathbf{W}_m\mathbf{1}$. The combinatorial graph Laplacian matrix $\mathbf{L}_m \in \mathbb{R}^{N \times N}$ for $\mathcal{G}_m$ is defined as $\mathbf{L}_m = \mathrm{diag}[\mathbf{W}_m\mathbf{1}] - \mathbf{W}_m$. By construction, $\mathbf{L}_m$ is a symmetric, positive semidefinite matrix, and has a zero row sum. The space of all the valid combinatorial graph Laplacian matrices of size N is given by

$$\mathcal{L} = \{\mathbf{L} \in \mathbb{S}_+^N \mid \mathbf{L}\mathbf{1} = \mathbf{0}; L_{ij} = L_{ji} \leq 0,\ \forall i \neq j\}, \quad (1)$$

in which the constraint $L_{ii} \geq 0$, $i = 1, 2, \cdots, M$ is implicit.

The number of connected components in a graph is given by the multiplicity of the zero eigenvalue of its graph Laplacian matrix [15]. Thus the rank of the graph Laplacian matrix of a K-component graph with N nodes is $N-K$. The eigenvectors of the graph Laplacian matrix corresponding to the K zero eigenvalues preserve the nodal connectivity information and hence are used as low-dimensional node embeddings.

A signal (or dataset) indexed using the nodes of a graph is referred to as a graph signal (or data). Consider M datasets $\mathbf{X}_m \in \mathbb{R}^{N \times D_m}$, $m = 1, 2, \ldots, M$ obtained from $M \geq 2$ views of a common source. The nth row of $\mathbf{X}_m$ that contains D_m features of the nth entity resides on the nth node of $\mathcal{G}_m$. That is, we interpret the M-view dataset as a graph signal defined on the multi-layer graph $\mathcal{G}$ with each layer representing a different view of the dataset.

We use smoothness to measure how well a signal matches the underlying graph. A signal is said to be smooth over a graph if the signal values residing on adjacent nodes having large edge weights are similar. For the signal $\mathbf{X}_m \in \mathbb{R}^{N \times D_m}$ residing on $\mathcal{G}_m$ with the graph Laplacian matrix $\mathbf{L}_m$, the smoothness is measured using the total variation with respect to the graph $\mathcal{G}_m$, and is given by $\mathrm{tr}(\mathbf{X}_m^{\mathrm{T}}\mathbf{L}_m\mathbf{X}_m) = \mathrm{tr}(\mathbf{L}_m\mathbf{S}_m)$. Here, $\mathbf{S}_m = \mathbf{X}_m\mathbf{X}_m^{\mathrm{T}}$ is the sample data covariance matrix.

III. Problem statement

In this work, we propose a *rank-constrained multi-layer graph learning* technique for clustering multi-view data. Specifically, we estimate the graph Laplacian matrices $\{\mathbf{L}_m\}_{m=1}^M$ that best explain the M-view dataset $\{\mathbf{X}_m\}_{m=1}^M$ by assuming that each view of the dataset is smooth on the graph of that layer. By constraining the rank of $\mathbf{L}_m$, $m = 1, 2, \ldots, M$ to $R = N - K$, we get K-component graph layers. Thus partitioning the nodes into K clusters. Since the entities in the multi-view dataset correspond to the same nodes in all the views, there exists a common low-dimensional representation of the nodes. To compute the common low-dimensional node embeddings, we propose *joint diagonalization* of $\{\mathbf{L}_m\}_{m=1}^M$:

$$\begin{aligned}\mathbf{L}_m &= \mathbf{U}\mathrm{diag}(\boldsymbol{\lambda}_m)\mathbf{U}^{\mathrm{T}} \\ &= \left[\begin{array}{c|c} \mathbf{Q} & \star \end{array}\right] \left[\begin{array}{c|c} \mathbf{0} & \\ \hline & \star \end{array}\right] \left[\begin{array}{c} \mathbf{Q}^{\mathrm{T}} \\ \hline \star \end{array}\right], \end{aligned} \quad (2)$$

for $m = 1, 2, \ldots, M$, where the columns of $\mathbf{U}$ contain the joint eigenvectors (common factors) shared by all the graph layers and the vector $\boldsymbol{\lambda}_m = [\lambda_1(\mathbf{L}_m), \lambda_2(\mathbf{L}_m), \cdots, \lambda_N(\mathbf{L}_m)]^{\mathrm{T}} \in \mathbb{R}_+^N$ contains the eigenvalues of $\mathbf{L}_m$. Here, $\lambda_i(\mathbf{L}_m)$ is the ith eigenvalue of $\mathbf{L}_m$, where we assume $0 = \lambda_1(\mathbf{L}_m) \leq \cdots \leq \lambda_N(\mathbf{L}_m)$. The common eigenvectors corresponding to the K zero eigenvalues of $\{\mathbf{L}_m\}_{m=1}^M$ are collected in $\mathbf{Q} \in \mathbb{R}^{N \times K}$.

To estimate the graph Laplacian matrices corresponding to the graphs in each layer, we propose the following *rank-constrained multi-layer graph learning* (RMGL) optimization problem:

$$\begin{aligned} \underset{\{\mathbf{L}_m, \boldsymbol{\lambda}_m\}_{m=1}^M, \mathbf{U}}{\text{minimize}} \quad & \sum_{m=1}^M \mathrm{tr}(\mathbf{L}_m\mathbf{S}_m) + \alpha_m \|\mathbf{L}_m\|_F^2 \\ \text{subject to} \quad & \mathbf{L}_m \in \mathcal{L}, \mathrm{tr}(\mathbf{L}_m) = N, \mathrm{rank}(\mathbf{L}_m) = R \\ & \mathbf{L}_m = \mathbf{U}\mathrm{diag}(\boldsymbol{\lambda}_m)\mathbf{U}^{\mathrm{T}},\ m = 1, \cdots, M, \\ & \mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}, \end{aligned} \quad (3)$$

where recall that the set $\mathcal{L}$, defined in (1), is the set of all the valid combinatorial graph Laplacian matrices. The first term in the objective function promotes smoothness of $\mathbf{X}_m$ with respect to the graph corresponding to the mth layer. The second term in the objective function with the tuning parameter $\alpha_m > 0$ allows us to control the sparsity (i.e., the number of nonzero entries) of $\mathbf{W}_m$. The trace constraint of the form $\mathrm{tr}(\mathbf{L}_m) = 2\|\mathrm{vec}(\mathbf{W}_m)\|_1 = N$ fixes the scale of the solution and avoids the trivial solution (more details in Section IV-A). The rank constraints on the Laplacian matrices promote K-component graph layers.

For $\mathbf{L}_m \in \mathbb{S}_+^N$, we have [19]

$$\sum_{i=1}^K \lambda_i(\mathbf{L}_m) = \underset{\mathbf{Q} \in \mathbb{R}^{N \times K}, \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_K}{\text{minimize}} \quad \mathrm{tr}(\mathbf{Q}^T\mathbf{L}_m\mathbf{Q}). \quad (4)$$

The optimal $\mathbf{Q}$ is given in (2). Using this property, the constraints $\mathrm{rank}(\mathbf{L}_m) = R$, $\mathbf{L}_m = \mathbf{U}\mathrm{diag}(\boldsymbol{\lambda}_m)\mathbf{U}$, for $m = 1, \cdots, M$, in (3) can be replaced with a sum-of-smallest-eigenvalues regularizer in the objective function as

$$\begin{aligned} \underset{\{\mathbf{L}_m\}_{m=1}^M, \mathbf{Q}}{\text{minimize}} \quad & \sum_{m=1}^M \mathrm{tr}(\mathbf{L}_m\mathbf{S}_m) + \alpha_m \|\mathbf{L}_m\|_F^2 + \beta_m \mathrm{tr}(\mathbf{Q}^T\mathbf{L}_m\mathbf{Q}) \\ \text{subject to} \quad & \mathbf{L}_m \in \mathcal{L}, \quad \mathrm{tr}(\mathbf{L}_m) = N, \quad m = 1, \cdots, M, \\ & \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_K, \end{aligned} \quad (5)$$

where for sufficiently large $\beta_m > 0$, we achieve $\mathrm{rank}(\mathbf{L}_m) = N - K$ for $m = 1, 2, \ldots, M$. The columns of the isometry $\mathbf{Q} \in \mathbb{R}^{N \times K}$ form an orthonormal basis for the low-dimensional subspace common to the M layers of the graph. In particular, the nth row of $\mathbf{Q}$ corresponds to the K-dimensional embedding of the nth node. The problem (5) is non-convex in the variables $\{\mathbf{L}_m\}_{m=1}^M$and $\mathbf{Q}$. In what follows, we present an efficient algorithm to solve for the unknowns $\{\mathbf{L}_m\}_{m=1}^M$and $\mathbf{Q}$ given the multi-view dataset $\{\mathbf{S}_m\}_{m=1}^M$.

IV. Proposed solver

In this section, we solve the problem in (5) by alternatingly minimizing it with respect to $\{\mathbf{L}_m\}_{m=1}^M$ and $\mathbf{Q}$, while keeping the other variable fixed.

A. Update of $\{\mathbf{L}_m\}_{m=1}^M$

Given $\mathbf{Q}$, the problem (5) simplifies to the following convex optimization problem

$$\begin{aligned} \underset{\{\mathbf{L}_m\}_{m=1}^M}{\text{minimize}} \quad & \sum_{m=1}^{M} \mathrm{tr}(\mathbf{L}_m \mathbf{R}_m) + \alpha_m \left\|\mathbf{L}_m\right\|_F^2 \\ \text{subject to} \quad & \mathbf{L}_m \in \mathcal{L}, \quad \mathrm{tr}(\mathbf{L}_m) = N, \quad m = 1, \cdots, M, \end{aligned} \tag{6}$$

where we have introduced the $N \times N$ matrix $\mathbf{R}_m = \mathbf{S}_m + \beta_m \mathbf{Q}\mathbf{Q}^{\mathrm{T}}$. Since $\mathbf{L}_m$ is a symmetric matrix, this quadratic program can be solved very efficiently by solving only for the *upper triangular entries* of $\mathbf{L}_m$ as described next.

Let us define a duplication matrix $\mathbf{D} \in \mathbb{R}^{N^2 \times V}$ with $V = N(N+1)/2$ as $\mathbf{D}^{\mathrm{T}} = \sum_{i \geq j} \boldsymbol{\delta}_{ij} \mathrm{vec}^{\mathrm{T}}(\boldsymbol{\Theta}_{ij})$, where the vector $\boldsymbol{\delta}_{ij} \in \mathbb{R}^V$ has 1 at position $(j-1)N + i - \frac{1}{2}j(j-1)$ and zero elsewhere. The matrix $\boldsymbol{\Theta}_{ij} \in \mathbb{R}^{N \times N}$ has -1 at the positions (i, j) and (j, i), 1 at (i, i), and zero elsewhere. Let us collect the absolute values of the V nonduplicated entries of $\mathbf{L}_m$ in $\mathbf{l}_m \in \mathbb{R}_+^V$. Then we have $\mathrm{vec}(\mathbf{L}_m) = \mathbf{D}\mathbf{l}_m$, for $m = 1, 2, \ldots, M$.

Using this transformation, we express the two equality constraints in (6), namely, $\mathrm{tr}(\mathbf{L}_m) = \mathrm{vec}(\mathbf{I})^T \mathrm{vec}(\mathbf{L}_m) = N$ and $\mathbf{L}\mathbf{1} = (\mathbf{1}^T \otimes \mathbf{I})\mathrm{vec}(\mathbf{L}_m) = \mathbf{0}$ as $\mathbf{C}\mathbf{l}_m = \mathbf{d}$, where $\mathbf{C} = [\mathbf{D}^{\mathrm{T}}\mathrm{vec}(\mathbf{I}), \mathbf{D}^{\mathrm{T}}(\mathbf{1} \otimes \mathbf{I})]^{\mathrm{T}} \in \mathbb{R}^{N+1 \times V}$ and $\mathbf{d} = [N, \mathbf{0}]^T \in \mathbb{R}^{N+1}$. Here, $\otimes$ denotes the Kronecker product. Similarly, we express the first term in the objective function of (6) as $\mathrm{tr}(\mathbf{L}_m \mathbf{R}_m) = \mathrm{vec}^{\mathrm{T}}(\mathbf{R}_m)\mathrm{vec}(\mathbf{L}_m) = \mathbf{r}_m^{\mathrm{T}} \mathbf{l}_m$. The second term in the objective function of (6) simplifies to $\alpha_m \|\mathbf{L}_m\|_F^2 = \alpha_m \mathrm{vec}(\mathbf{L}_m)^{\mathrm{T}} \mathrm{vec}(\mathbf{L}_m) = \frac{1}{2}\mathbf{l}_m^{\mathrm{T}} \mathrm{diag}(\mathbf{p}_m)\mathbf{l}_m$, where we have used the fact that $\mathbf{D}^{\mathrm{T}}\mathbf{D}$ is a positive definite diagonal matrix to obtain $\mathrm{diag}(\mathbf{p}_m) = 2\alpha_m \mathbf{D}^{\mathrm{T}}\mathbf{D}$. Now, we can simplify (6) as

$$\begin{aligned} \underset{\{\mathbf{l}_m\}_{m=1}^M}{\text{minimize}} \quad & \sum_{m=1}^{M} \frac{1}{2}\mathbf{l}_m^{\mathrm{T}} \mathrm{diag}(\mathbf{p}_m)\mathbf{l}_m + \mathbf{r}_m^{\mathrm{T}} \mathbf{l}_m \\ \text{subject to} \quad & \mathbf{C}\mathbf{l}_m = \mathbf{d}, \quad \mathbf{l}_m \succeq \mathbf{0}, \quad m = 1, \cdots, M. \end{aligned} \tag{7}$$

This is a special form of a convex quadratic program in which the matrix associated with the quadratic term is diagonal. It is computationally efficient and equivalent to solve for each $\mathbf{l}_m$, $m = 1, 2, \ldots, M$ separately as this convex program is separable in these variables.

The Lagrangian function for the problem (7) associated to the variable $\mathbf{l}_m$ is given by

$$\begin{aligned} \mathcal{J}(\mathbf{l}_m, \boldsymbol{\lambda}_m, \boldsymbol{\mu}_m) = & \frac{1}{2}\mathbf{l}_m^{\mathrm{T}} \mathrm{diag}(\mathbf{p}_m)\mathbf{l}_m \\ & + \mathbf{r}_m^{\mathrm{T}} \mathbf{l}_m + \boldsymbol{\mu}_m^{\mathrm{T}}(\mathbf{d} - \mathbf{C}\mathbf{l}_m) - \boldsymbol{\lambda}_m^{\mathrm{T}} \mathbf{l}_m, \end{aligned}$$

for $m = 1, 2, \ldots, M$, where $\boldsymbol{\mu}_m \in \mathbb{R}^{N+1}$ and $\boldsymbol{\lambda}_m \in \mathbb{R}^V$ are the Lagrange multipliers corresponding to the equality and inequality constraints, respectively. The Karush-Kuhn-Tucker (KKT) conditions are given by

$$\begin{aligned} & \mathrm{diag}(\mathbf{p}_m)\mathbf{l}_m^\star + \mathbf{r}_m - \mathbf{C}^{\mathrm{T}}\boldsymbol{\mu}_m^\star - \boldsymbol{\lambda}_m^\star = \mathbf{0}, \\ & \mathbf{C}\mathbf{l}_m^\star = \mathbf{d}, \quad \mathbf{l}_m^\star \succeq \mathbf{0}, \quad \boldsymbol{\lambda}_m^\star \odot \mathbf{l}_m^\star = \mathbf{0}. \end{aligned}$$

Eliminating the variable $\boldsymbol{\lambda}_m^\star$ and solving for $\mathbf{l}_m^\star$, we get $\mathbf{l}_m^\star(\boldsymbol{\mu}_m^\star) = \left\{\mathrm{diag}^{-1}(\mathbf{p}_m)[\mathbf{C}^{\mathrm{T}}\boldsymbol{\mu}_m^\star - \mathbf{r}_m]\right\}_+$, where $\{\cdot\}_+$ denotes the elementwise projection onto the nonnegative orthant. Using $\mathbf{l}_m^\star(\boldsymbol{\mu}_m^\star)$ in the second KKT condition, we can compute $\boldsymbol{\mu}_m^\star$ iteratively as

$$\mathbf{l}_m^{(k)} = \left\{\mathrm{diag}^{-1}(\mathbf{p}_m)[\mathbf{C}^{\mathrm{T}}\boldsymbol{\mu}_m^{(k)} - \mathbf{r}_m]\right\}_+, \tag{8}$$

$$\boldsymbol{\mu}_m^{(k+1)} = \boldsymbol{\mu}_m^{(k)} - \rho[\mathbf{C}\mathbf{l}_m^{(k)} - \mathbf{d}], \tag{9}$$

where $\rho > 0$ is the step size. We initialize the iterations with $\boldsymbol{\mu}_m^{(0)}$. The computational complexity of these iterations is dominated by the matrix-vector multiplication. Since $\mathbf{C}$ is sparse with $N(N+1)$ non-zero entries, the above iterative procedure approximately costs order N^2 flops. For M variables, the total computation cost is approximately order MN^2 flops.

B. Update of $\mathbf{Q}$

Given $\{\mathbf{L}_m\}_{m=1}^M$, (5) reduces to the following eigenvalue problem

$$\begin{aligned} \underset{\mathbf{Q} \in \mathbb{R}^{N \times K}}{\text{minimize}} \quad & \mathrm{tr}(\mathbf{Q}^T \mathbf{L} \mathbf{Q}) \\ \text{subject to} \quad & \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_K \quad \text{and} \quad \mathbf{L} = \sum_{m=1}^{M} \beta_m \mathbf{L}_m. \end{aligned} \tag{10}$$

The optimal $\mathbf{Q}$ is given by the K eigenvectors corresponding to the K smallest eigenvalues of $\mathbf{L} = \sum_{m=1}^M \beta_m \mathbf{L}_m$. Computing this partial eigendecomposition approximately costs KN^2 flops [20].

The complete alternating minimization procedure is summarized as Algorithm 1. It approximately costs order $(M+K)N^2$ flops per iteration. Furthermore, it can be shown that each limit of the sequence of updates of the optimization problems in (7) and (10) satisfy the KKT conditions of (5).

V. Numerical experiments

In this section, we evaluate our framework in terms of clustering performance on a synthetically generated dataset and two real-world datasets. We use normalized mutual information (NMI) as a measure to assess the clustering performance. We compare the clustering performances of the proposed method `RMGL` with both common baseline methods and state-of-the-art multi-view spectral clustering methods. The single-view graph learning method based on signal representation (`GL-SigRep`) in [6] serves as our first baseline

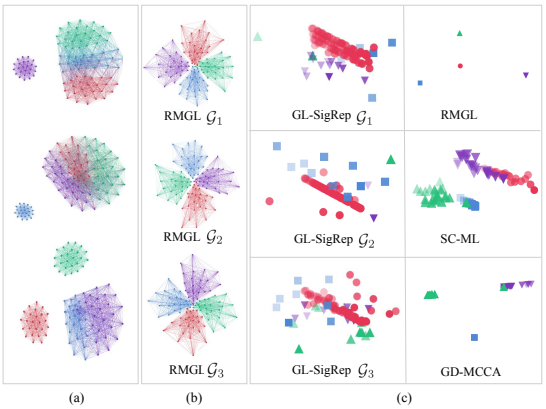

Fig. 1: (a) Ground truth graphs. (b) Reconstructed graphs. (c) Node embeddings.

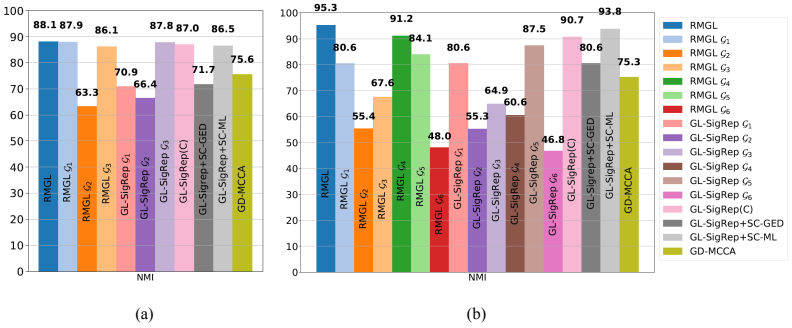

Fig. 2: NMI. (a) UCI hand-written digits (b) COIL-20 datasets.

method. We learn the graph layers independently for each of the views available using `GL-SigRep`. We also learn a graph Laplacian matrix by giving the concatenated views $\mathbf{X} \in \mathbb{R}^{N \times \sum_{m=1}^{M} D_m}$ as input to `GL-SigRep`. We refer to the second baseline method that uses concatenated data as input as `GL-SigRep(C)`. Spectral clustering is then performed to find the clusters on the graph Laplacian matrices estimated using the above mentioned two methods. We also compare our method with state-of-the-art methods for multi-view spectral clustering, which are based on a two-step approach of graph learning followed by subspace merging. Graph-regularized dual multi-view CCA (`GD-MCCA`) [18] learns a low-dimensional representation of the data common to all the views by minimizing the distance between the canonical variables and the common low-dimensional representations while leveraging a graph structure of the common source. We compute the common graph required by `GD-MCCA` as in [18], by taking each of the affinity matrices $\mathbf{D}_m^{-1}\mathbf{X}_m\mathbf{X}_m^{\mathrm{T}}$ as the common graph. Here, $\mathbf{D}_m$ is the degree matrix of the similarity graph $\mathbf{X}_m\mathbf{X}_m^T$. We finally consider the common graph giving the best results for comparison. The multi-view spectral clustering methods, `SC-GED` [16] and `SC-ML` [17], obtain the common node embeddings by computing a joint spectrum via a generalized eigendecomposition and merging the node embeddings from multiple graph layers, respectively. We give the independent graph layers estimated from [6] as input to `SC-GED` and `SC-ML` to find the common node embeddings and perform spectral clustering.

Algorithm 1 Rank-constrained multi-layer graph learning

function `RMGL`$(\{\mathbf{p}_m, \mathbf{r}_m\}_{m=1}^M$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{d}$, K, `Tol`, ρ, `MaxIter`$)$
  Initialize $k \leftarrow 0$
  while $k <$ `MaxIter` **do**
    for $m = 1$ to M **do**
      Initialize $\boldsymbol{\mu}_m \leftarrow 0$
      while $\|\mathbf{C}\mathbf{l}_m - \mathbf{d}\|_2 <$ `Tol` **do**
        $\mathbf{l}_m \leftarrow \left\{\mathrm{diag}^{-1}(\mathbf{p}_m)[\mathbf{C}^T\boldsymbol{\mu}_m - \mathbf{r}_m]\right\}_+$
        $\boldsymbol{\mu}_m \leftarrow \boldsymbol{\mu}_m - \rho[\mathbf{C}\mathbf{l}_m - \mathbf{d}]$
      $\mathbf{L}_m \leftarrow$ `mat`$(\mathbf{D}\mathbf{l}_m)$
        ▷ `mat` is the inverse vectorization operator.
    $\mathbf{L} \leftarrow \sum_{m=1}^M \beta_m \mathbf{L}_m$
    $\mathbf{Q} \leftarrow$ `eigs`$(\mathbf{L}, K)$
      ▷ `eigs` computes the eigenvectors associated to the K smallest eigenvalues of the matrix argument.
    $k \leftarrow k + 1$
  return $\{\mathbf{L}_m\}_{m=1}^M$ and $\mathbf{Q}$

We first curate a synthetic dataset to illustrate the merit of the proposed method through a pedagogical approach. We construct a 3-layer graph, as shown in Fig. 1(a). Each of the graph layers consists of $N = 100$ nodes with an equal number of samples from four classes, indicated by different colors. The graph layers are so constructed that they carry complementary information that none of the individual graphs provide. Each layer has a different class of nodes disconnected from the rest of the nodes that have inter-class connections. The denser within-class connections than the inter-class connections present in all the views convey that the nodes with different colors belong to different classes. We generate data $\mathbf{X}_m \in \mathbb{R}^{100 \times 15}$, $m = 1, 2, 3$ by taking the eigenvectors corresponding to the 15 smallest eigenvalues of the individual graph Laplacian matrices and corrupting it with zero-mean additive white Gaussian noise of variance 0.01. The estimated graph layers with K-components from the proposed method `RMGL` are shown in Fig. 1(b). From Fig. 1(c), we can see that the embeddings obtained using the three estimated graph layers using `GL-SigRep` fail to incorporate the combined information from different views. In contrast, `RMGL` assigns the same node representations to the points belonging to one class and hence are better clustered by our proposed method than the representations obtained using `SC-GED`, `SC-ML`, and `GD-MCCA`.

`RMGL`'s performance on the real datasets, UCI hand-written digits [21] and COIL-20 [22], shows its merit more accurately. The UCI dataset consists of 200 instances of images of the 10 digits 0 to 9. We consider six different views of this data as in [18] with $N = 2000$, $D_1 = 216$, $D_2 = 76$, $D_3 = 64$, $D_4 = 6$, $D_5 = 240$, and $D_6 = 47$. The COIL-20 dataset is an image dataset consisting of 20 classes of objects with 72 images

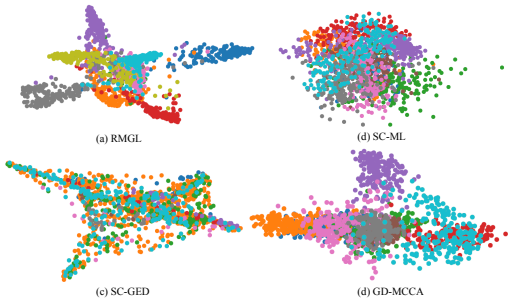

Fig. 3: Node embeddings for the UCI hand-written digits dataset.

of each object captured at different angles. We construct three views of this data related to the local binary pattern, histogram of oriented gradients, and pixel values of resized images with $N = 1440$, $D_1 = 59$, $D_2 = 36$, and $D_3 = 1024$.

With the different views of data taken as input, we estimate $\{\mathbf{L}_m\}_{m=1}^M$ and $\mathbf{Q}$. The rows of $\mathbf{Q}$ are then given as input to the k-means algorithm for clustering. We also perform spectral clustering on $\{\mathbf{L}_m\}_{m=1}^M$. The accuracy of the clusters obtained using these can be treated as a measure of the correctness of the estimated graph. The parameters involved in the different methods considered are chosen to obtain the best possible NMI. The values of $\{\beta_m\}_{m=1}^M$, for `RMGL`, can be chosen according to the importance of the available views by choosing a higher value of β_m for a view m that is more informative. Our model gives the best results for the following choice of parameters: $\alpha_m = 100$, $m = 1, \ldots, 6$, $\beta_1 = \beta_2 = \beta_3 = 2$, and $\beta_4 = 20, \beta_5 = 7, \beta_6 = 1$ for the UCI dataset. For the COIL-20 dataset, we use $\alpha_m = 100$, $m = 1, 2, 3$, and $\beta_1 = 12, \beta_2 = 2, \beta_3 = 15$. The bar plot in Fig. 2 shows the NMI values averaged over 20 experiments. The NMI variance using the proposed and competing methods is very small (about 10^{-6}), and hence not shown. The NMI scores of the different methods in Fig. 2 suggest that using $\mathbf{Q}$ from `RMGL` as an input to the k-means algorithm gives the best clustering accuracy when compared to the other considered methods (including performing spectral clustering on the estimated graph layers either from `RMGL`, `GL-SigRep` or `GL-SigRep(C)`). Fig. 3 shows the node embeddings corresponding to the 2000 samples from the ten classes of data in the UCI dataset set. We show the two-dimensional node representations obtained from the last two columns of the common K-dimensional subspaces computed using `RMGL`, `SC-GED`, `SC-ML`, `GD-MCCA`. The embeddings obtained from `RMGL` show better separability, with the node embeddings from the same cluster concentrated and those from different clusters far apart.

VI. Conclusions

We developed a framework for multi-view clustering, where we simultaneously learn a multi-layer graph with a common vertex set and a low-dimensional representation for the nodes common to all the views. We presented an efficient solver based on an alternating minimization procedure to solve the proposed non-convex optimization problem. We demonstrated via numerical experiments on synthetic and real datasets that the proposed method performs better than the existing multi-view clustering methods based on CCA or merging the node embeddings of the individual graph layers.

References

[1] Y. Yang and H. Wang, "Multi-view clustering: A survey," *Big Data Mining and Analytics*, vol. 1, no. 2, pp. 83–107, June 2018.

[2] D. Greene and P. Cunningham, "Producing a unified graph representation from multiple social network views," in *Proc. of the 5th annual ACM web science conf.*, Paris, France, May 2013.

[3] V. D. Calhoun and J. Sui, "Multimodal fusion of brain imaging data: A key to finding the missing link(s) in complex mental illness," *Biological psychiatry. Cognitive neuroscience and neuroimaging*, vol. 1, no. 3, p. 230—244, May 2016.

[4] F. Battiston, V. Nicosia, M. Chavez, and V. Latora, "Multilayer motif analysis of brain networks," *Chaos: An Interdisciplinary J. of Nonlinear Science*, vol. 27, no. 4, Mar. 2017.

[5] M. Vaiana and S. F. Muldoon, "Multilayer brain networks," *J. of Nonlinear Science*, vol. 30, no. 5, pp. 2147–2169, Jan. 2018.

[6] 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.

[7] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Connecting the dots: Identifying network structure via graph signal processing," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 44–63, May 2019.

[8] M. Coutino, S. P. Chepuri, and G. Leus, "Sparsest network support estimation: A submodular approach," in *Proc. of IEEE Data Science Workshop (DSW)*, Lausanne, Switzerland, June 2018.

[9] S. P. Chepuri, M. Coutino, A. G. Marques, and G. Leus, "Distributed analytical graph identification," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*.

[10] S. P. Chepuri, S. Liu, G. Leus, and A. O. Hero, "Learning sparse graphs under smoothness prior," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, New Orleans, USA, Mar. 2017.

[11] S. K. Kadambari and S. P. Chepuri, "Product graph learning from multi-domain data with sparsity and rank constraints," *arXiv preprint arXiv:2012.08090*, Dec. 2020.

[12] M. Gonzalo, S. Santiago, M. Antonio G, and R. Alejandro, "Connecting the dots: Identifying network structure via graph signal processing," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 16–43, Dec. 2019.

[13] S. K. Kadambari and S. P. Chepuri, "Learning product graphs from multidomain signals," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, Barcelona, Spain, May 2020.

[14] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *arXiv preprint arXiv:1709.05584*, Sep. 2017.

[15] U. Von Luxburg, "A tutorial on spectral clustering," *Stats. and comput.*, vol. 17, no. 4, pp. 395–416, Aug. 2007.

[16] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "Clustering with multi-layer graphs: A spectral perspective," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5820–5831, Nov. 2012.

[17] ——, "Clustering on multi-layer graphs via subspace analysis on Grassmann manifolds," *IEEE Trans. Signal Process.*, vol. 62, no. 4, pp. 905–918, Feb. 2014.

[18] C. Jia, W. Gang, and G. B. Giannakis, "Graph multiview canonical correlation analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 4, pp. 905–918, June 2014.

[19] T. Tao, *Topics in random matrix theory*. American Mathematical Soc., 2012.

[20] L. K. Saul and S. T. Roweis, "An introduction to locally linear embedding," *unpublished. Available at: http://www..nyu.edu/˜roweis/lle/publications.html*, Jan. 2001.

[21] R. P. Duin, "UCI multiple features dataset of handwritten digits extracted from dutch utility maps," 1998 (accessed Aug., 2020). [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Multiple+Features

[22] S. Nane, S. Nayar, and H. Murase, "Columbia object image library: Coil-20," *Dept. Comp. Sci., Columbia University, New York, Tech. Rep*, Feb. 1996.