Clustering of Incomplete Data via a Bipartite Graph Structure

Amirhossein Javaheri and Daniel P. Palomar
Hong Kong University of Science and Technology
sajavaheri@connect.ust.hk, palomar@ust.hk

***Abstract*—There are various approaches to graph learning for data clustering, incorporating different spectral and structural constraints through diverse graph structures. Some methods rely on bipartite graph models, where nodes are divided into two classes: centers and members. These models typically require access to data for the center nodes in addition to observations from the member nodes. However, such additional data may not always be available in many practical scenarios. Moreover, popular Gaussian models for graph learning have demonstrated limited effectiveness in modeling data with heavy-tailed distributions, which are common in financial markets. In this paper, we propose a clustering method based on a bipartite graph model that addresses these challenges. First, it can infer clusters from incomplete data without requiring information about the center nodes. Second, it is designed to effectively handle heavy-tailed data. Numerical experiments using real financial data validate the efficiency of the proposed method for data clustering.**

***Index Terms*—Graph learning, data clustering, bipartite graph, heavy-tailed distribution, incomplete data, financial data.**

I. Introduction

Clustering is a fundamental technique in data mining and machine learning [1], with applications spanning bio-informatics [2] image segmentation [3], [4], social networks [5], and financial markets [6]. Among various clustering techniques including hierarchical [7], density-based [8], copula-based [9], and model-based methods [10], there are approaches that leverage graphical models to capture similarities between data points [11]. Spectral graph clustering [12] is an early example of such methods, utilizing heuristic similarity graphs to represent data relationships. A more advanced approach involves learning graphs from data under a Gaussian Markov random field (GMRF) model. The Graphical LASSO (GLASSO) [13] was a foundational method in this area, later enhanced by introducing Laplacian structural constraints to the graph learning problem [14]–[18].

Recent approaches in graph-based clustering incorporate spectral constraints into graph learning to infer specific structures, such as k-component graphs, which explicitly represent clusters [19], [20]. However, these methods often struggle to accurately model data with heavy-tailed distributions, which are prevalent in real-world applications like financial markets. Consequently, recent efforts have focused on generalizing k-component graph learning methods to handle heavy-tailed distributions [21]–[23]. Another category of methods for clustering introduces k-component bipartite graph structures, comprising two types of nodes: cluster centers and their members [24], [25]. While effective, these approaches require additional data for the center nodes to learn the graph. This dependency on center node data, however, poses challenges in applications where such data is unavailable.

In this paper, we propose a bipartite graph learning method for data clustering. Unlike existing approaches (e.g., [25]) which can only be used for complete data, our method is designed to handle scenarios with incomplete information about the center nodes. It is also robust to heavy-tailed data distributions, making it suitable for a wide range of real-world applications, e.g., financial markets.

A. Notations

Vectors and matrices are respectively denoted with bold lowercase and uppercase letters. (e.g., $\mathbf{x}$ and $\mathbf{X}$). The i-th element of a vector $\mathbf{x}$ is denoted by x_i. Also $X_{i,j}$ denotes the (i,j)-th element of $\mathbf{X}$. The notation $\mathbf{x}_{i_1:i_2}$ is defined as $\mathbf{x}_{i_1:i_2} \triangleq [x_{i_1}, \ldots, x_{i_2}]$. We use $\|\mathbf{x}\|$ to denote the ℓ_2 norm of a vector and $\|\mathbf{X}\|_F$ to denote the Frobenius norm of a matrix. Diagonal elements of $\mathbf{X}$ are shown with $\operatorname{diag}(\mathbf{X})$, while $\operatorname{Diag}(\mathbf{x})$ is a diagonal matrix with $\mathbf{x}$ on its main diagonal. The notation $\det^*$ also represents the generalized determinant.

II. Problem Formulation

Consider a weighted undirected graph with p vertices, where each node represents an element of a signal $\mathbf{x} \in \mathbb{R}^p$, and the weights of the edges encode the relationships between the elements. Suppose we have n measurements of $\mathbf{x}$, represented as the columns of $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p\times n}$. The problem we investigate in this paper is to learn an undirected k-component bipartite graph for clustering the data into k clusters. In this graph structure, one class of nodes represents the centers of the clusters, while the other class represents the members (to be clustered). Assume we divide the nodes into k centers and r members, where $r + k = p$. Let $B_{i,j} \geq 0$ be the weight of the edge connecting the member $i \in \{1, \cdots, r\}$ to the center $j \in \{1, \cdots, k\}$. Then, the Laplacian matrix of the graph can be expressed as

$$\mathbf{L} = \begin{bmatrix} \operatorname{Diag}(\mathbf{B}\mathbf{1}_k) & -\mathbf{B} \\ -\mathbf{B}^\top & \operatorname{Diag}(\mathbf{B}^\top \mathbf{1}_r) \end{bmatrix}, \tag{1}$$

where $\mathbf{B} \in \mathbb{R}_+^{r\times k}$. The weight $B_{i,j}$ models the probability of member node i being within cluster j, Hence, the sum of each row of $\mathbf{B}$ equals one.

Next, we consider a stochastic approach to learning such graph structure from data. In specific, we assume $\mathbf{x}_i$s are

This work was supported by the Hong Kong GRF 16206123 research grant.

drawn from a zero-mean multivariate Student's t-distribution as

$$p(\mathbf{x}_i) \propto \det{}^*(\mathbf{L})^{1/2}\left(1+\frac{\mathbf{x}_i^\top \mathbf{L}\mathbf{x}_i}{\nu}\right)^{-(\nu+p)/2}, \quad \nu > 2.$$

By selecting a sufficiently large value for ν (degrees of freedom), we can also handle general Gaussian-distributed data. Under the above model, the problem of maximum likelihood estimation of the graph from data can be formulated by

$$\begin{array}{ll} \underset{\mathbf{L},\,\mathbf{B}}{\mathsf{min}} & \frac{p+\nu}{n}\sum_{i=1}^n \log\left(1+\frac{\mathbf{x}_i^\top \mathbf{L}\mathbf{x}_i}{\nu}\right) - \log\det{}^*(\mathbf{L}) \\ \mathsf{s.\ t.} & \mathbf{L} = \left[\begin{smallmatrix} \mathrm{Diag}(\mathbf{B}\mathbf{1}_k) & -\mathbf{B} \\ -\mathbf{B}^\top & \mathrm{Diag}(\mathbf{B}^\top\mathbf{1}_r)\end{smallmatrix}\right],\ \mathrm{rank}(\mathbf{L}) = p-k, \\ & \mathbf{B} \geq 0,\ \mathbf{B}\mathbf{1}_k = \mathbf{1}_r. \end{array}$$

Suppose we are only given the first r rows of the data matrix, corresponding to the members, which we refer to as the incomplete data denoted by $\tilde{\mathbf{X}} \in \mathbb{R}^{r\times n}$. We may consider each unavailable row of the data matrix, corresponding to the centers, to be a weighted average of the rows of $\tilde{\mathbf{X}}$ (members). Hence, the augmented data matrix yields as $\mathbf{X} = \begin{bmatrix}\tilde{\mathbf{X}}^\top & \tilde{\mathbf{X}}^\top\mathbf{A}\end{bmatrix}^\top \in \mathbb{R}^{p\times n}$, where $\mathbf{A} \in \mathbb{R}_+^{r\times k}$ denotes the weight matrix with non-negative elements. The sum of these weights for each center node equals unity, and hence, we have $\mathbf{A}^\top\mathbf{1}_r = \mathbf{1}_k$. We also assume $\mathbf{A}$ and $\mathbf{B}$ share the same support (indicating the membership sets of the clusters).

Consider the i-th column of $\mathbf{X}$ as $\mathbf{x}_i^\top = [\tilde{\mathbf{x}}_i^\top \ \tilde{\mathbf{x}}_i^\top\mathbf{A}]^\top$, where $\tilde{\mathbf{x}}_i$ is the i-th column of $\tilde{\mathbf{X}}$. Also let $\tilde{\mathbf{S}}_i = \tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_i^\top$. Then with simple calculations we get:

$$\begin{aligned} &\mathbf{x}_i^\top\mathbf{L}\mathbf{x}_i = h_i + \mathrm{tr}(\mathbf{B}\mathbf{G}_i(\mathbf{A})), \quad h_i = \tilde{\mathbf{x}}_i^\top\tilde{\mathbf{x}}_i = \mathrm{tr}(\tilde{\mathbf{S}}_i), \\ &\mathbf{G}_i(\mathbf{A}) = -2\mathbf{A}^\top\tilde{\mathbf{S}}_i + \mathrm{diag}\left(\mathbf{A}^\top\tilde{\mathbf{S}}_i\mathbf{A}\right)\mathbf{1}_r^\top. \end{aligned} \tag{2}$$

Using this equation, the problem would be restated as

$$\begin{array}{ll} \underset{\mathbf{L},\,\mathbf{B},\,\mathbf{A}}{\mathsf{min}} & \frac{p+\nu}{n}\sum_{i=1}^n \log\left(1+\frac{h_i+\mathrm{tr}(\mathbf{B}\mathbf{G}_i(\mathbf{A}))}{\nu}\right) \\ & -\log\det{}^*(\mathbf{L}) \\ \mathsf{s.\ t.} & \mathbf{L} = \left[\begin{smallmatrix} \mathbf{I}_r & -\mathbf{B} \\ -\mathbf{B}^\top & \mathrm{Diag}(\mathbf{B}^\top\mathbf{1}_r)\end{smallmatrix}\right],\ \mathrm{rank}(\mathbf{L}) = p-k, \\ & \mathbf{B}\mathbf{1}_k = \mathbf{1}_r,\ \mathbf{A}^\top\mathbf{1}_r = \mathbf{1}_k, \\ & \mathbf{A},\mathbf{B} \geq \mathbf{0},\ \mathbf{A}\odot\mathbb{1}(\mathbf{B}=\mathbf{0}) = \mathbf{0}, \end{array} \tag{3}$$

where we have used $\mathbf{B}\mathbf{1}_k = \mathbf{1}_r$ and $\mathrm{Diag}(\mathbf{1}_r) = \mathbf{I}_r$ in the first equality constraint.

III. Proposed Method

By relaxing the constraint $\mathbf{L} = \left[\begin{smallmatrix} \mathbf{I}_r & -\mathbf{B} \\ -\mathbf{B}^\top & \mathrm{Diag}(\mathbf{B}^\top\mathbf{1}_r)\end{smallmatrix}\right]$, the augmented Lagrangian function yields as

$$\begin{aligned} L_\rho(\mathbf{A},\mathbf{B},\mathbf{L}) = &\frac{p+\nu}{n}\sum_{i=1}^n \log\left(1+\frac{h_i+\mathrm{tr}(\mathbf{B}\mathbf{G}_i(\mathbf{A}))}{\nu}\right) \\ &-\log\det{}^*(\mathbf{L}) + \frac{\rho}{2}\left\|\mathbf{L} - \left[\begin{smallmatrix} \mathbf{I}_r & -\mathbf{B} \\ -\mathbf{B}^\top & \mathrm{Diag}(\mathbf{B}^\top\mathbf{1}_r)\end{smallmatrix}\right]\right\|_F^2 \\ &+ \langle \mathbf{L} - \left[\begin{smallmatrix} \mathbf{I}_r & -\mathbf{B} \\ -\mathbf{B}^\top & \mathrm{Diag}(\mathbf{B}^\top\mathbf{1}_r)\end{smallmatrix}\right], \mathbf{Y}\rangle \end{aligned} \tag{4}$$

Now, we use a variant of the ADMM method [26] to solve problem (3) by alternating minimization of the augmented Lagrangian function. Here, we have 3 update steps corresponding to the primal variables $\mathbf{L}$, $\mathbf{B}$, and $\mathbf{A}$, and one update step for the dual variable $\mathbf{Y}$. These are given as follows:

A. $\mathbf{L}$ *update step:*

The update step for $\mathbf{L}$ is obtained by solving the following subproblem

$$\begin{aligned} \mathbf{L}^{l+1} = \underset{\substack{\mathbf{L}\succeq\mathbf{0}, \\ \mathrm{rank}(\mathbf{L})=p-k}}{\mathsf{argmin}} \ &\frac{\rho}{2}\left\|\mathbf{L} - \left[\begin{smallmatrix} \mathbf{I}_r & -\mathbf{B}^l \\ -\mathbf{B}^{l\top} & \mathrm{Diag}(\mathbf{B}^{l\top}\mathbf{1}_r)\end{smallmatrix}\right] + \frac{1}{\rho}\mathbf{Y}^l\right\|_F^2 \\ &- \log\det{}^*(\mathbf{L}) \end{aligned}$$

The closed-form solution to this is given as

$$\mathbf{L}^{l+1} = \frac{1}{2\rho}\mathbf{U}^l\left(\mathbf{\Sigma}^l + (\mathbf{\Sigma}^{l^2} + 4\rho\mathbf{I})^{1/2}\right)\mathbf{U}^{l\top}, \tag{5}$$

with $\mathbf{\Sigma}^l$ being a diagonal matrix having the largest $p-k$ eigenvalues of $\rho\left[\begin{smallmatrix} \mathbf{I}_r & -\mathbf{B}^l \\ -\mathbf{B}^{l\top} & \mathrm{Diag}(\mathbf{B}^{l\top}\mathbf{1}_r)\end{smallmatrix}\right] - \mathbf{Y}^l$ with corresponding eigenvectors $\mathbf{U}^l$.

B. $\mathbf{B}$ *update step:*

The subproblem associated with the update step of $\mathbf{B}$ is formulated as follows:

$$\begin{aligned} \mathbf{B}^{l+1} &= \underset{\mathbf{B}\geq\mathbf{0},\,\mathbf{B}\mathbf{1}_k=\mathbf{1}_r}{\mathsf{argmin}} \ f_{\mathbf{B}}(\mathbf{B}) \\ f_{\mathbf{B}}(\mathbf{B}) &= \frac{p+\nu}{n}\sum_i \log\left(1+\frac{h_i+\mathrm{tr}(\mathbf{B}\mathbf{G}_i(\mathbf{A}^l))}{\nu}\right) + \\ &\quad \frac{\rho}{2}\left\|\mathbf{L}^{l+1} - \left[\begin{smallmatrix} \mathbf{I}_r & -\mathbf{B} \\ -\mathbf{B}^\top & \mathrm{Diag}(\mathbf{B}^\top\mathbf{1}_r)\end{smallmatrix}\right] + \frac{1}{\rho}\mathbf{Y}^l\right\|_F^2. \end{aligned} \tag{6}$$

This problem does not admit a closed-form solution. Thus, we first simplify the problem using the majorization-minimization (MM) technique [27]. For this, we need to find a (smooth) majorization function $f_{\mathbf{B}}^S(\mathbf{B};\mathbf{B}^l)$ with the following properties:

$$\begin{aligned} f_{\mathbf{B}}^S(\mathbf{B};\mathbf{B}^l) &> f_{\mathbf{B}}(\mathbf{B}), \quad \forall\mathbf{B}\neq\mathbf{B}^l \\ f_{\mathbf{B}}^S(\mathbf{B}^l;\mathbf{B}^l) &= f_{\mathbf{B}}(\mathbf{B}^l), \end{aligned} \tag{7}$$

where $\mathbf{B}^l$ is a constant matrix.

Lemma 1. *The function $f_{\mathbf{B}}(\mathbf{B})$ in (6) can be majorized as*

$$f_{\mathbf{B}}^S(\mathbf{B};\mathbf{B}^l) = \mathrm{tr}(\mathbf{B}\mathbf{H}^l) + \rho\left\|\mathbf{B}\right\|_F^2 + \frac{\rho}{2}\mathbf{1}_r^\top\mathbf{B}\mathbf{B}^\top\mathbf{1}_r + C(\mathbf{B}^l) \tag{8}$$

where

$$\begin{aligned} \mathbf{H}^l &= \mathbf{P}^l + \rho\left(\mathbf{M}_{rk}^{l\top} + \mathbf{M}_{kr}^l - \mathrm{diag}(\mathbf{M}^l)_{r+1:p}\mathbf{1}_r^\top\right), \\ \mathbf{P}^l &= \frac{p+\nu}{n}\sum_i \frac{\mathbf{G}_i(\mathbf{A}^l)}{h_i + \mathrm{tr}\left(\mathbf{B}^l\mathbf{G}_i(\mathbf{A}^l)\right) + \nu}, \\ \mathbf{M}^l &= \left[\begin{smallmatrix} \mathbf{M}_{rr}^l & \mathbf{M}_{rk}^l \\ \mathbf{M}_{kr}^l & \mathbf{M}_{kk}^l\end{smallmatrix}\right] = \mathbf{L}^{l+1} + \frac{1}{\rho}\mathbf{Y}^l. \end{aligned}$$

Proof. See Appendix A. □

Using this lemma, the upperbounded problem can be expressed as

$$\mathbf{B}^{l+1} = \underset{\mathbf{B}\geq\mathbf{0},\ \mathbf{B}\mathbf{1}_k=\mathbf{1}_r}{\mathsf{argmin}} \ f_{\mathbf{B}}^S(\mathbf{B};\mathbf{B}^l). \tag{9}$$

To solve this, we use the projected gradient descent (PGD) method [28] with the the step-size μ as:

$$\begin{aligned}\mathbf{B}^{m+1} &= \mathcal{P}_{\mathcal{S}_{\text{row}}}\left(\mathbf{B}^m - \mu\nabla f^S_{\mathbf{B}}(\mathbf{B}^m;\mathbf{B}^l)\right),\\ \nabla f^S_{\mathbf{B}}(\mathbf{B}^m;\mathbf{B}^l) &= \mathbf{H}^{l\top} + 2\rho\mathbf{B}^m + \rho\mathbf{1}_r\mathbf{1}_r^\top\mathbf{B}^m.\end{aligned} \tag{10}$$

where $\mathcal{P}_{\mathcal{S}_{\text{row}}}$ is the operator that projects each row of the operand matrix onto the simplex $\{\mathbf{x} \geq \mathbf{0},\, \langle\mathbf{x},\mathbf{1}\rangle = 1\}$. The solution to this problem can be found efficiently via water-filling algorithms [29].

C. A update step:

The subproblem for the update step of $\mathbf{A}$ is

$$\begin{array}{ll}\mathsf{min} & f_{\mathbf{A}}(\mathbf{A})\\ \mathsf{s.\ t.} & \mathbf{A} \geq \mathbf{0},\ \mathbf{A}^\top\mathbf{1}_r = \mathbf{1}_k,\ \mathbf{A}\odot\mathbb{1}(\mathbf{B}^{l+1} = 0) = \mathbf{0},\end{array} \tag{11}$$

where

$$f_{\mathbf{A}}(\mathbf{A}) = \frac{p+\nu}{n}\sum_{i=1}^{n}\log\left(1 + \frac{h_i + \operatorname{tr}(\mathbf{B}^{l+1}\mathbf{G}_i(\mathbf{A}))}{\nu}\right). \tag{12}$$

Here, we again apply the MM to solve the problem, due to difficulty in obtaining the closed-form solution. We use the following lemma to construct a majorization for $f_{\mathbf{A}}(\mathbf{A})$.

Lemma 2. *Let $\mathbf{a}_j$ denote the j-th -column of $\mathbf{A}$. Then, for constant $\mathbf{A}^l$, $f_{\mathbf{A}}(\mathbf{A})$ in (12) can be majorized via*

$$f^S_{\mathbf{A}}(\mathbf{A};\mathbf{A}^l) = \textstyle\sum_{j=1}^{k} g^S_{\mathbf{a}_j}(\mathbf{a}_j;\mathbf{A}^l) + C(\mathbf{A}^l), \tag{13}$$

where

$$\begin{aligned}g^S_{\mathbf{a}_j}(\mathbf{a}_j;\mathbf{A}^l) &= b_j^{l+1}\mathbf{a}_j^\top\tilde{\mathbf{S}}^l\mathbf{a}_j - 2\mathbf{b}_j^{l+1\top}\tilde{\mathbf{S}}^l\mathbf{a}_j,\\ \tilde{\mathbf{S}}^l &= \frac{p+\nu}{n}\sum_i\frac{\tilde{\mathbf{S}}_i}{h_i + \operatorname{tr}(\mathbf{B}^{l+1}\mathbf{G}_i(\mathbf{A}^l)) + \nu},\end{aligned} \tag{14}$$

$b_j^{l+1} = \langle\mathbf{b}_j^{l+1},\mathbf{1}\rangle$, *and* $\mathbf{b}_j^{l+1}$ *denotes the j-column of* $\mathbf{B}^{l+1}$.

Proof. Here the proof is again based on the logarithmic inequality $\log(x) \leq x - 1,\ \forall x > 0$. We also use (2) to restate $f_{\mathbf{A}}(\mathbf{A})$ as a sum of functions of columns of $\mathbf{A}$. □

Next, the update step for $\mathbf{A}$ is obtained by solving the following equation:

$$\begin{array}{rll}\mathbf{A}^{l+1} = & \mathsf{argmin} & f^S_{\mathbf{A}}(\mathbf{A};\mathbf{A}^l)\\ & \mathsf{s.t.} & \mathbf{A} \geq \mathbf{0},\ \mathbf{A}^\top\mathbf{1}_r = \mathbf{1}_k,\\ & & \mathbf{A}\odot\mathbb{1}(\mathbf{B}^{l+1} = 0) = \mathbf{0}\end{array} \tag{15}$$

This problem can iteratively be solved for each $\mathbf{a}_j$ via the PGD as

$$\begin{aligned}\mathbf{a}_j^{m+1} &= \underset{\substack{\mathbf{a}_j\geq\mathbf{0},\ \mathbf{a}_j^\top\mathbf{1}=1\\ \mathbf{a}_j\odot\mathbb{1}(\mathbf{b}_j^{l+1}=0)=\mathbf{0}}}{\mathsf{argmin}}\quad g^S_{\mathbf{a}_j}(\mathbf{a}_j,\mathbf{A}^l)\\ &= \mathcal{P}_{\mathcal{S}}\left(\mathbf{a}_j^m - 2\eta\,\tilde{\mathbf{S}}^l(b_j^{l+1}\mathbf{a}_j^m - \mathbf{b}_j^{l+1})\right)\odot\mathbb{1}(\mathbf{b}_j^{l+1} > 0)\end{aligned} \tag{16}$$

where η is the step-size and $\mathcal{P}_{\mathcal{S}}$ refers to the projection operator that maps a vector onto the simplex $\{\mathbf{x} \geq \mathbf{0},\, \mathbf{x}^\top\mathbf{1} = 1\}$.

Algorithm 1 Proposed algorithm for bipartite k-component graph learning

```
Input: $\tilde{\mathbf{X}} \in \mathbb{R}^{r\times n}$    Parameters: $k$, $\nu$, $\rho$, $\mu$, and $\eta$
Output: $\mathbf{B}^l$
Initialization: $\mathbf{A}^0$, $\mathbf{B}^0$, $l = 0$
repeat
    Update $\mathbf{L}^{l+1}$ using (5).
    Update $\mathbf{B}^{l+1}$ by iterating (10) (starting from $\mathbf{B}^l$).
    Update $\mathbf{A}^{l+1}$ by iterating (16) (starting from $\mathbf{A}^l$).
    Update the dual variable using (17).
    Set $l \leftarrow l + 1$.
until a stopping criterion is satisfied
```

D. Dual variable update step:

Finally we have the update step for the dual variable as

$$\mathbf{Y}^{l+1} = \mathbf{Y}^l + \rho\left(\mathbf{L}^{l+1} - \begin{bmatrix}\mathbf{I}_r & -\mathbf{B}^{l+1}\\ -\mathbf{B}^{l+1\top} & \operatorname{Diag}(\mathbf{B}^{l+1\top}\mathbf{1}_r)\end{bmatrix}\right). \tag{17}$$

IV. Numerical Results

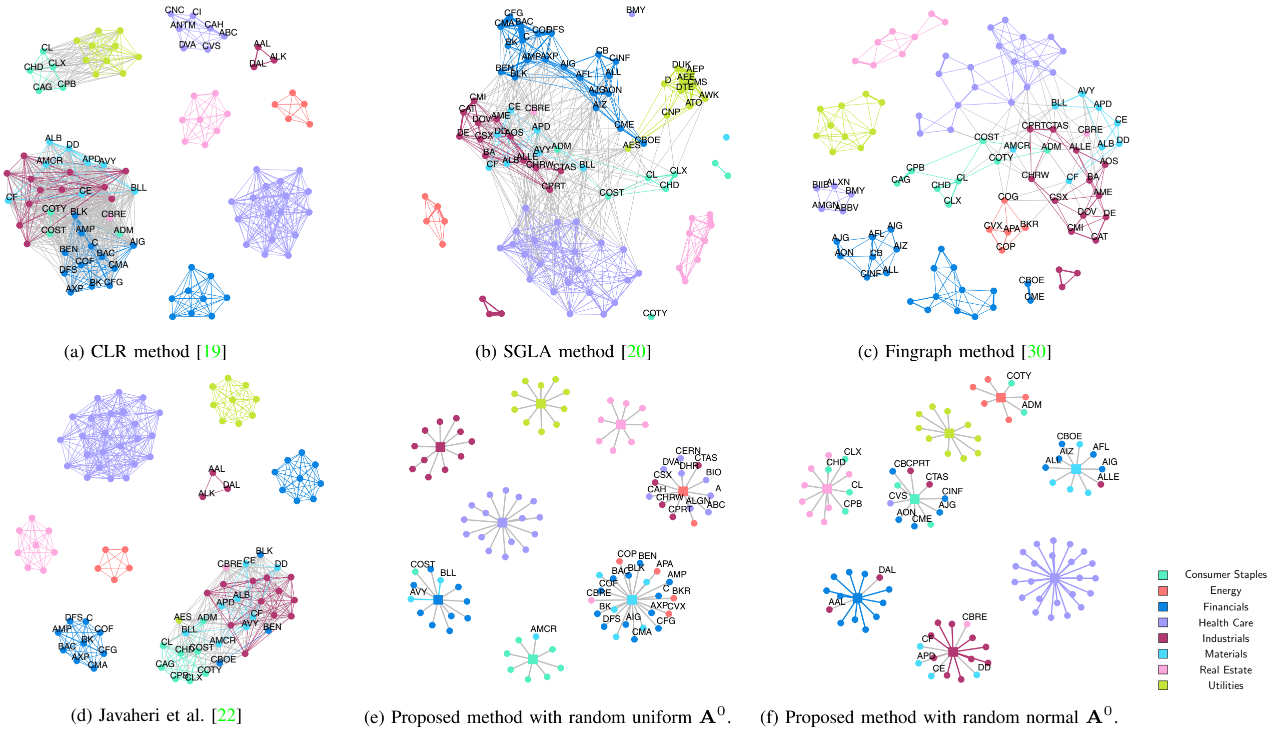
In this part, we present numerical results to evaluate the performance of our proposed method for clustering heavy-tailed data. For this purpose, we utilize real-world financial data, specifically the log-returns of S&P 500 stocks. Our experiment focuses on a subset of 100 stocks, divided into $k = 8$ clusters corresponding to financial sectors, with ground-truth cluster labels defined by the GICS classification standard[1]. The log-returns of these stocks are calculated over a 1000-day period from January 2016 to January 2020. The resulting data matrix $\tilde{\mathbf{X}} \in \mathbb{R}^{r\times n}$ consists of $r = 100$ rows (stocks) and $n = 1000$ columns (days).

To assess clustering performance, we employ accuracy (ACC), purity [31], modularity (MOD) [32], adjusted Rand index (ARI) [33], and Calinski-Harabasz index (CHI) [34]. Accuracy and purity measure the ratio of true-positive labels to p. Accuracy considers the optimal alignment of inferred cluster labels to ground truth across all $k!$ permutations, while purity considers the majority label within each cluster as the ground truth. ARI, on the other hand, quantifies the similarity between the true and inferred cluster labels. Modularity also measures how disjoint the nodes with different labels are. The Calinski-Harabasz index (CHI) is a reference-free criterion that measures the ratio of the between-cluster separation to the within-cluster separation.

To run our method, we first obtain ν by fitting a multivariate Student's t-distribution to the data using the `fitHeavyTail` R package[2]. The parameter k in our method corresponds to the number of desired clusters ($k = 8$ in this experiment). We also set $\rho = 1$, $\mu = 0.5$, and $\eta = 0.005$ (determined via cross-validation). We then consider two cases for initialization of $\mathbf{A}$. In the first case, $\mathbf{A}^0$ is sampled from a random uniform $U[0, 1]$ distribution. In the other case, the entries of $\mathbf{A}^0$ are drawn from

[1] https://www.msci.com/our-solutions/indexes/gics

[2] https://CRAN.R-project.org/package/fitHeavyTail



(a) CLR method [19]

(b) SGLA method [20]

(c) Fingraph method [30]

(d) Javaheri et al. [22]

(e) Proposed method with random uniform $\mathbf{A}^0$.

(f) Proposed method with random normal $\mathbf{A}^0$.

Fig. 1: The graphs learned from financial data corresponding to the log-returns of 100 stocks in S&P500 index (including $k = 8$ sectors).

the normal distribution $\mathcal{N}(0,1)$. We later normalize $\mathbf{A}^0$ so that each column has unit sum. The augmented data matrix is then constructed as $\mathbf{X} = \begin{bmatrix} \tilde{\mathbf{X}}^\top & \tilde{\mathbf{X}}^\top \mathbf{A} \end{bmatrix}^\top$. Using this, we compute the initial value of $\mathbf{B}^0 = \mathcal{P}_{\Omega_\mathbf{B}}\left([(\mathbf{X}^\top \mathbf{X}/n)^\dagger]_{1:r,(r+1):p}\right)$, where $\Omega_\mathbf{B}$ denotes the set of feasible $\mathbf{B}$ matrices. Once the graph is learned, we assign the argument of the maximum entry of the i-th row of $\mathbf{B}^l$ as the cluster label for node i.

We compare our proposed method against several benchmark algorithms for k-component graph learning, including the constrained Laplacian rank (CLR) method [19], the SGLA method [20], the Fingraph method [21], and the method proposed by Javaheri et al. [22] Note that under the incomplete data scenario considered here, conventional bipartite graph learning benchmarks (e.g., [24], [25]) cannot be employed, as they require complete data statistics for center nodes, which are missing in our setup.

Figure 1 illustrates the learned graphs with node colors representing the ground-truth clusters (sector indices). The clustering performance associated with these graphs are also given in Table I. As shown in the table, the graph learned by the proposed method with random normal initialization achieves the highest accuracy and the highest ARI, while the result with the uniform initialization gives the highest purity. Overall, the proposed method is shown to have superior performance for financial data clustering.

TABLE I: Clustering performance of the graphs shown in Fig. 1

	ACC	Purity	MOD	ARI	CHI
CLR [19]	0.61	0.71	0.34	0.43	4.73
SGLA [20]	0.42	0.44	0.41	0.12	3.08
Fingraph [30]	0.51	0.66	0.64	0.26	3.48
Javaheri et al. [22]	0.67	0.79	0.52	0.57	4.34
Proposed (uniform $\mathbf{A}^0$)	0.67	**0.81**	**0.85**	0.53	**5.05**
Proposed (normal $\mathbf{A}^0$)	**0.73**	0.77	**0.85**	**0.63**	**5.05**

V. Conclusion

In this paper, we addressed the problem of learning bipartite k-component graphs for clustering data with heavy-tailed distributions. We model the heavy-tailed behavior using Student's t-distribution, which includes the Gaussian distribution as a special case when the degrees-of-freedom parameter is sufficiently large. Unlike existing methods that rely on access to data for both cluster centers and members in a bipartite graph model, our proposed approach addresses this limitation by jointly inferring the topology and the center nodes of the graph. Numerical experiments highlight the efficiency of the proposed method in clustering heavy-tailed data, particularly data from financial markets.

Appendix A
Proof of Lemma 1

For simplicity in presentation, let us omit the superscript of the variables. Let $\mathbf{M} = \mathbf{L} + \frac{1}{\rho}\mathbf{Y}$, $\mathbf{m} = \mathrm{diag}(\mathbf{M})$, $\mathbf{m}_r = \mathbf{m}_{1:r}$, and $\mathbf{m}_k = \mathbf{m}_{r+1:p}$. Then, we may write

$$\begin{aligned} &\left\langle \mathbf{M}, \begin{bmatrix} \mathbf{I}_r & -\mathbf{B} \\ -\mathbf{B}^\top & \mathrm{Diag}(\mathbf{B}^\top \mathbf{1}_r) \end{bmatrix} \right\rangle \\ &= \mathbf{m}_r^\top \mathbf{1}_r - \langle \mathbf{M}_{rk}, \mathbf{B} \rangle - \langle \mathbf{M}_{kr}, \mathbf{B}^\top \rangle + \mathbf{m}_k^\top \mathbf{B}^\top \mathbf{1}_r \\ &= \mathbf{m}_r^\top \mathbf{1}_r - \mathrm{tr}\left(\left(\mathbf{M}_{rk}^\top + \mathbf{M}_{kr} - \mathbf{m}_k \mathbf{1}_r^\top \right) \mathbf{B} \right), \end{aligned} \tag{18}$$

and

$$\left\| \begin{bmatrix} \mathbf{I}_r & -\mathbf{B} \\ -\mathbf{B}^\top & \mathrm{Diag}(\mathbf{B}^\top \mathbf{1}_r) \end{bmatrix} \right\|_F^2 = r + 2\left\|\mathbf{B}\right\|_F^2 + \mathbf{1}_r^\top \mathbf{B}\mathbf{B}^\top \mathbf{1}_r. \quad (19)$$

Thus, $f_{\mathbf{B}}(\mathbf{B})$ in problem (6), can be simplified as

$$f_{\mathbf{B}}(\mathbf{B}) = \mathrm{tr}(\mathbf{B}\mathbf{R}) + \rho\left\|\mathbf{B}\right\|_F^2 + \frac{\rho}{2}\mathbf{1}_r^\top \mathbf{B}\mathbf{B}^\top \mathbf{1}_r + \frac{p+\nu}{n}\sum_i \log\left(1 + \frac{h_i + \mathrm{tr}(\mathbf{B}\mathbf{G}_i(\mathbf{A}))}{\nu}\right),$$

where $\mathbf{R} = \rho\left(\mathbf{M}_{rk}^\top + \mathbf{M}_{kr} - \mathbf{m}_k \mathbf{1}_r^\top\right)$.

Now, using the logarithmic inequality, $\log(x) \leq x - 1,\ \forall x > 0$, one can find an upperbound as follows:

$$\frac{p+\nu}{n}\sum_i \log\left(1 + \frac{h_i + \mathrm{tr}(\mathbf{B}\mathbf{G}_i(\mathbf{A}))}{\nu}\right) \leq \mathrm{tr}\left(\mathbf{B}\mathbf{P}_0\right) + C(\mathbf{B}_0),$$

with

$$\mathbf{P}_0 = \frac{p+\nu}{n}\sum_i \frac{\mathbf{G}_i(\mathbf{A})}{h_i + \mathrm{tr}\left(\mathbf{B}_0\mathbf{G}_i(\mathbf{A})\right) + \nu}.$$

and $C(\mathbf{B}_0)$ being a constant term. Thus, we may propose a majorization function for $f_{\mathbf{B}}(\mathbf{B})$ as

$$f_{\mathbf{B}}^S(\mathbf{B}; \mathbf{B}_0) = \mathrm{tr}(\mathbf{B}\mathbf{H}) + \rho\left\|\mathbf{B}\right\|_F^2 + \frac{\rho}{2}\mathbf{1}_r^\top \mathbf{B}\mathbf{B}^\top \mathbf{1}_r + C(\mathbf{B}_0),$$

where $\mathbf{H}_0 = \mathbf{P}_0 + \mathbf{R}$. Thus, one can obtain (8) by choosing $\mathbf{B}_0 = \mathbf{B}^l$.

References

[1] G. J. Oyewole and G. A. Thopil, "Data clustering: application and trends," *Artificial Intelligence Review*, vol. 56, no. 7, pp. 6439–6475, July 2023.

[2] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences*, vol. 95, no. 25, pp. 14863–14868, 1998.

[3] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[4] Z. Li, Y. Zheng, L. Cao, L. Jiao, Y. Zhong, and C. Zhang, "A student's t-based density peaks clustering with superpixel segmentation (tdpcss) method for image color clustering," *Color Research & Application*, vol. 45, no. 4, pp. 656–670, 2020.

[5] K. Singh, H. K. Shakya, and B. Biswas, "Clustering of people in social network based on textual similarity," *Perspectives in Science*, vol. 8, pp. 570–573, Sept. 2016.

[6] G. Marti, F. Nielsen, M. Bińkowski, and P. Donnat, *A review of two decades of correlations, hierarchies, networks and clustering in financial markets*, 2021, arXiv:1703.00485.

[7] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, pp. 241–254, 1967.

[8] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. 1996, p. 226–231, AAAI Press.

[9] T. Massing, "Clustering using student t mixture copulas," *SN Comput. Sci.*, vol. 2, no. 2, Feb. 2021.

[10] G. J. McLachlan and D. Peel, *Finite Mixture Models*, John Wiley & Sons, Mar. 2004.

[11] J. Paratte, "Graph-based Methods for Visualization and Clustering," 2017, Publisher: Lausanne, EPFL.

[12] A. Ng, M. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2001, vol. 14.

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

[14] B. Lake and J. Tenenbaum, "Discovering structure by learning sparse graphs," in *Proceedings of the 32nd Annual Meeting of the Cognitive Science Society*, Portland, Oregon, United States, Aug. 2010, pp. 778–784.

[15] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph Learning From Data Under Laplacian and Structural Constraints," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 6, pp. 825–841, Sept. 2017.

[16] L. Zhao, Y. Wang, S. Kumar, and D. P. Palomar, "Optimization Algorithms for Graph Laplacian Estimation via ADMM and MM," *IEEE Transactions on Signal Processing*, vol. 67, no. 16, pp. 4231–4244, Aug. 2019.

[17] J. Ying, J. V. de M. Cardoso, and D. P. Palomar, "Nonconvex Sparse Graph Learning under Laplacian Constrained Graphical Model," in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 7101–7113.

[18] A. Javaheri, A. Amini, F. Marvasti, and D. P. Palomar, "Learning Spatiotemporal Graphical Models From Incomplete Observations," *IEEE Transactions on Signal Processing*, vol. 72, pp. 1361–1374, 2024.

[19] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The Constrained Laplacian Rank algorithm for graph-based clustering," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, Arizona, Feb. 2016, AAAI'16, pp. 1969–1976.

[20] S. Kumar, J. Ying, J. V. de M. Cardoso, and D. P. Palomar, "A Unified Framework for Structured Graph Learning via Spectral Constraints," *Journal of Machine Learning Research*, vol. 21, no. 22, pp. 1–60, 2020.

[21] J. V. M. Cardoso, J. Ying, and D. P. Palomar, "Graphical models in heavy-tailed markets," in *Advances in Neural Information Processing Systems*, 2021, vol. 34, pp. 19989–20001.

[22] A. Javaheri, J. V. de M. Cardoso, and D. P. Palomar, "Graph Learning for Balanced Clustering of Heavy-Tailed Data," in *2023 IEEE 9th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Herradura, Costa Rica, Dec. 2023, pp. 481–485.

[23] A. Javaheri and D. P. Palomar, "Learning time-varying graphs for heavy-tailed data clustering," in *2024 32nd European Signal Processing Conference (EUSIPCO)*, 2024, pp. 2472–2476.

[24] F. Nie, X. Wang, C. Deng, and H. Huang, "Learning a structured optimal bipartite graph for co-clustering," in *Neural Information Processing Systems*, 2017.

[25] J. V. de M. Cardoso, J. Ying, and D. P. Palomar, "Learning Bipartite Graphs: Heavy Tails and Multiple Components," *Advances in Neural Information Processing Systems*, vol. 35, pp. 14044–14057, Dec. 2022.

[26] S. Boyd, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[27] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-Minimization Algorithms in Signal Processing, Communications, and Machine Learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 794–816, Feb. 2017.

[28] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 1 edition, Mar. 2004.

[29] D.P. Palomar and J.R. Fonollosa, "Practical algorithms for a family of waterfilling solutions," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 686–695, 2005.

[30] J. V. de M. Cardoso, J. Ying, and D. P. Palomar, "Graphical Models in Heavy-Tailed Markets," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021, vol. 34, pp. 19989–20001.

[31] B. Everitt, Ed., *Cluster analysis*, Wiley series in probability and statistics. Wiley, Chichester, West Sussex, U.K, 5th ed edition, 2011.

[32] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, June 2006.

[33] W. M. Rand, "Objective Criteria for the Evaluation of Clustering Methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, Dec. 1971.

[34] T. Caliński and J Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics*, vol. 3, no. 1, pp. 1–27, 1974.