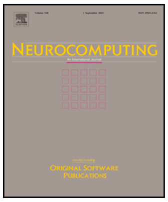
Neurocomputing 601 (2024) 128210

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



A dual Laplacian framework with effective graph learning for unified fair spectral clustering

Xiang Zhang, Qiao Wang [*]

School of Information Science and Engineering, Southeast University, Nanjing, 210096, China

ARTICLE INFO

Communicated by L. Oneto

Keywords:
Spectral clustering
Graph learning
Joint optimization
Fairness constraints
Spectral embedding

ABSTRACT

We consider the problem of spectral clustering under group fairness constraints, where samples from each sensitive group are approximately proportionally represented in each cluster. Traditional fair spectral clustering (FSC) methods consist of two consecutive stages, i.e., performing fair spectral embedding on a *given* graph and conducting $k$means to obtain discrete cluster labels. However, in practice, the graph is usually unknown, and we need to construct the underlying graph from potentially noisy data, the quality of which inevitably affects subsequent fair clustering performance. Furthermore, performing FSC through separate steps breaks the connections among these steps, leading to suboptimal results. To this end, we first theoretically analyze the effect of the constructed graph on FSC. Motivated by the analysis, we propose a novel graph construction method with a node-adaptive graph filter to learn graphs from noisy data. Then, all independent stages are integrated into a single objective function via a dual Laplacian framework, forming an end-to-end model that inputs raw data and outputs discrete cluster labels. An algorithm is developed to jointly and alternately update the variables in each stage. Finally, we conduct extensive experiments on synthetic, benchmark, and real data, which show that our model is superior to state-of-the-art fair clustering methods.

1. Introduction

Clustering is an unsupervised task that aims to group samples with common attributes and separate dissimilar samples. It has numerous practical applications, e.g., image processing [1], remote sensing [2], and bioinformatics [3]. Existing clustering methods include $k$means [4], spectral clustering (SC) [5], hierarchical clustering [6], and numerous methods in multi-view clustering [7]. Among these methods, SC is a graph-based method utilizing topological information of data and has achieved impressive performance in various applications [5].

Recently, many concerns have arisen regarding fairness when performing clustering algorithms. For example, in loan applications, applicants are grouped into several clusters to support cluster-specific loan policies. However, clustering results could be affected by sensitive factors such as race and gender [8], even if the clustering algorithms do not consider sensitive attributes. Unfair clustering can lead to discriminatory outcomes, such as a specific group being more likely to be denied a loan. Therefore, there is a growing need for fair clustering methods unbiased by sensitive attributes. In the literature, [9] first introduces the notion of group fairness into clustering. As illustrated in Fig. 1, given data with sensitive attributes, group fairness aims to find a data partition where samples in each sensitive group are approximately proportionally represented in each cluster [9]. In this way, every sensitive group is treated fairly. Following [9,10] generalizes the definition of fair clustering, [11] proposes a scalable fair clustering algorithm, and [12] applies the variational method to fair clustering. Furthermore, fairness constraints are also incorporated into deep clustering methods that leverage deep neural networks to partition data [13,14].

Here, we consider the problem of fair spectral clustering (FSC). The first work exploring FSC is [15], which designs a fairness constraint for SC according to the definition of group fairness in [9]. Then, a scalable algorithm is proposed in [16] to solve the model in [15], and [17] considers group fairness of normalized-cut graph partitioning. In [18], individual fairness is considered in SC, which utilizes a representation graph to encode sensitive attributes and requires the neighbors of a node in the graph to be approximately proportionally represented in the clusters. More recently, [19] proposes a fair multi-view SC method. However, existing FSC models are built on a *given* similarity graph, which may not be available in practice. Before proceeding with FSC algorithms, it is necessary to construct a graph from raw data. Thus, a complete FSC method typically consists of three subtasks. First, a similarity graph is constructed from raw data. Second, spectral embedding under fairness constraints is performed on the similarity graph to

[*] Corresponding author.
E-mail addresses: xiangzhang369@seu.edu.cn (X. Zhang), qiaowang@seu.edu.cn (Q. Wang).

https://doi.org/10.1016/j.neucom.2024.128210
Received 5 February 2024; Received in revised form 6 May 2024; Accepted 14 July 2024
Available online 18 July 2024

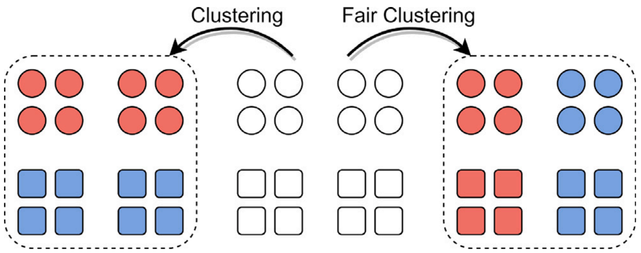


Fig. 1. The illustration of fair clustering. Given data points of two sensitive groups (represented by squares and circles), fair clustering partitions them into two clusters (represented by blue and red), where samples of each group are proportionally represented in each cluster.

obtain low-dimensional embeddings. Third, conducting post-processing like $k$means on the embeddings to obtain discrete cluster labels.

Although feasible, the traditional FSC paradigm still has the following problems to be addressed. (i) The quality of the constructed graph inevitably affects subsequent fair clustering performance, but this has not been explored theoretically. Additionally, noisy observations make it more difficult to construct accurate graphs that captures intrinsic topological relationships behind the data. (ii) The post-processing $k$means is sensitive to the initial cluster centers and could cause far deviation from the true discrete results [20]. (iii) Performing three subtasks separately breaks the connections among graph construction, fair spectral embedding, and discretization, leading to suboptimal clustering results. For example, the downstream fair clustering task imposes fair cluster structure constraints on the learned graph. However, independent graph construction does not consider the effect of subsequent tasks, which may fail to find the optimal graph for fair clustering [21]. Besides, [22] shows that independent spectral embedding is inferior to joint optimization of graph construction and spectral embedding.

To address the above issues, we propose a novel graph construction method, based on which a unified FSC model is built. The contributions of this study are summarized below.

- We theoretically analyze the impact of the constructed graph on fair clustering, justifying the necessity of an accurate graph to improve FSC performance. Motivated by the analysis, we propose a graph construction method equipped with a node-adaptive graph filter to learn accurate graphs from potentially noisy data as inputs to FSC.
- We propose a unified FSC model integrating denoising, graph construction, fair spectral embedding, and discretization into a single objective function via a dual Laplacian framework. Our model is an end-to-end framework that inputs observed data and outputs discrete fair clustering results and a similarity graph that captures the fair community structures underlying the data.
- We develop an algorithm to solve the objective function of our model. Compared with separate optimization, our algorithm updates all variables jointly and alternately, leading to an overall optimal solution for all subtasks.
- We conduct extensive experiments on synthetic, benchmark, and real data to test the proposed method. Experimental results demonstrate that our model outperforms state-of-the-art fair clustering methods.

Organization: The rest of this paper is organized as follows. Section 2 presents some related works, including graph construction methods for (fair) SC and unified SC models. For completeness, we present preliminaries for FSC in Section 3. Then, our FSC framework is proposed in Section 4. To solve the proposed model, we propose the algorithm in Section 5. Next, we conduct extensive experiments to test the proposed method in Section 6. Finally, concluding remarks are presented in Section 7.

Notations: Throughout this paper, vectors, matrices, and sets are written in bold lowercase, bold uppercase letters, and calligraphic uppercase letters, respectively. Given a matrix $\mathbf{B}$, $\mathbf{B}_{[i,:]}$, $\mathbf{B}_{[:,j]}$, and $\mathbf{B}_{[ij]}$ denote the ith row, the ith column, and the (i,j) entry of $\mathbf{B}$, respectively. $\mathbf{B} \geq 0$ means all elements of $\mathbf{B}$ are non-negative. Furthermore, $\mathrm{diag}(\mathbf{B})$ and $\mathrm{diag}_0(\mathbf{B})$ mean converting the diagonal elements of $\mathbf{B}$ to a vector and setting the diagonal entries of $\mathbf{B}$ to zeros. The vectors $\mathbf{1}$, $\mathbf{0}$, and matrix $\mathbf{I}$ represent all-one vectors, all-zero vectors, and identity matrices, respectively. Moreover, $\|\cdot\|_F$, $\|\cdot\|_{1,1}$, and $\|\cdot\|_q$ are the Frobenius norm, element-wise ℓ_1 norm, and ℓ_q norm of a vector (matrix), respectively. The notations $\dagger$, $\circ$, and $\mathrm{Tr}(\cdot)$ are pseudo inverse, Hadamard product, and trace operator, respectively. Given a set $\mathcal{B}$, $|\mathcal{B}|$ is the number of elements in $\mathcal{B}$. Finally, $\mathbb{R}$ and $\mathbb{S}$ are the domain of real values and symmetric matrices whose dimensions depend on the context.

2. Related work

2.1. Graph learning methods for (fair) SC

Graph learning (GL) aims to infer the graph topology behind observed data, a prerequisite step for (fair) SC when similarity graphs are unavailable. Traditionally, graphs are constructed via some direct rules, such as k-nearest-neighborhood (k-NN), ϵ-nearest-neighborhood (ϵ-NN) [23], and sample correlation methods like Pearson correlation (PC). These methods may be limited in capturing similarity relationships between data pairs [24]. Thus, many works attempt to learn graphs from data adaptively, including the sparse representation (SR) method [25] and the low-rank representation method [26]. Besides, [27] proposes a self-expression strategy with robust logarithmic loss and doubly stochastic constraint to learn graphs for clustering. The emergence of adaptive neighborhood graph learning (ANGL) [28,29] provides a new way that uses the probability of two samples being adjacent to measure the similarity between them. Based on ANGL, [30] proposes a robust method to learn graphs for clustering from noisy data, and [31] integrates the consistency propagation theory into graph learning to capture the topological structure more comprehensively. In [32], a possibilistic neighborhood graph is proposed, an improved version of [28]. Recently, with the rise of graph signal processing (GSP) [33], many works attempt to learn graphs from the perspective of signal processing. One of the widely-used GSP-based GL methods postulates that signals are smooth over the corresponding graphs [34]. Intuitively, a smooth graph signal means the signal values of two connected nodes are similar [35], which is also a fundamental principle of SC [5]. Many methods are dedicated to learning graphs from smooth signals [36,37]. However, limited to our understanding, applying smoothness-based GL to SC has yet to be thoroughly explored, let alone FSC.

2.2. Unified SC models

Many works focus on establishing a unified model for SC, which can be roughly divided into three categories. The first one integrates graph construction and spectral embedding [22,28,38]. They use an independent discretization step as post-processing. The second one is based on a given similarity graph. They integrate spectral embedding and discretization [39–41]. The third category integrates all three stages into a single objective function [21,24,42–44]. Our model differs from these models in two main ways. (i) Our framework utilizes a new graph construction method. (ii) We further consider fairness issues in clustering tasks.

3. Background

This section presents background information on SC under group fairness. Given an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ of D vertices, where $\mathcal{V}$

and $\mathcal{E}$ are the sets of vertices and edges of $\mathcal{G}$, respectively, its adjacency matrix $\mathbf{W} \in \mathbb{S}^{D\times D}$ is a symmetric matrix with zero diagonal entries and non-negative off-diagonal elements if the graph has non-negative edge weights and no self-loops. The Laplacian matrix of $\mathcal{G}$ is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D} \in \mathbb{S}^{D\times D}$ is a diagonal matrix satisfying $\mathbf{D}_{[ii]} = \sum_{j=1}^{D} \mathbf{W}_{[ij]}$. Unnormalized SC aims to partition D nodes into K disjoint clusters $\mathcal{C}_1, \dots, \mathcal{C}_K$, where $\mathcal{V} = \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_K$, and $\mathcal{C}_k$ is the set containing nodes in the k–th cluster. This problem is equivalent to minimizing the RatioCut objective function [5], i.e.,

$$\mathrm{RatioCut}(\mathcal{C}_1, \dots, \mathcal{C}_K) = \sum_{k=1}^{K} \frac{\mathrm{Cut}(\mathcal{C}_k, \mathcal{V} \setminus \mathcal{C}_k)}{|\mathcal{C}_k|}, \tag{1}$$

where $\mathcal{V} \setminus \mathcal{C}_k$ contains all nodes in $\mathcal{V}$ except those in $\mathcal{C}_k$, and

$$\mathrm{Cut}(\mathcal{C}_k, \mathcal{V} \setminus \mathcal{C}_k) = \sum_{i\in\mathcal{C}_k,\ j\in\mathcal{V}\setminus\mathcal{C}_k} \mathbf{W}_{[ij]}. \tag{2}$$

In practice, minimizing RatioCut (2) is usually relaxed to

$$\min_{\mathbf{U}} \mathrm{Tr}(\mathbf{U}^\top \mathbf{L}\mathbf{U}), \quad \text{s.t. } \mathbf{U}^\top\mathbf{U} = \mathbf{I}, \tag{3}$$

where $\mathbf{U} \in \mathbb{R}^{D\times K}$ is a relaxed continuous clustering label matrix, and $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$ is adopted to avoid trivial solutions. The process of solving (3) is called spectral embedding. After obtaining $\mathbf{U}^*$, a common practice is to apply $k$means to the rows of $\mathbf{U}^*$ to yield final discrete clustering labels $\mathbf{Q}$, where $\mathbf{Q} \in \{0,1\}^{D\times K}$ is a binary cluster indicator matrix. The only non-zero element of the ith row of $\mathbf{Q}$ indicates the cluster membership of the ith node of $\mathcal{G}$.

However, $k$means usually exhibits poor performance and depends heavily on initial cluster centers. Spectral rotation [20] is an alternative to $k$means, which is formulated as

$$\min_{\mathbf{Q},\mathbf{R}} \|\mathbf{Q} - \mathbf{U}\mathbf{R}\|_{\mathrm{F}}^2, \ \text{s.t. } \mathbf{R}^\top\mathbf{R} = \mathbf{I}, \mathbf{Q} \in \mathcal{I}, \tag{4}$$

where the set $\mathcal{I}$ contains all discrete cluster indicator matrices, and $\mathbf{R} \in \mathbb{R}^{K\times K}$ is an orthonormal matrix. According to the spectral solution invariance property [20], if $\mathbf{U}$ is a solution of (3), $\mathbf{U}\mathbf{R}$ is another solution. A suitable $\mathbf{R}$ could facilitate $\mathbf{U}\mathbf{R}$ as close to $\mathbf{Q}$ as possible. In contrast, $k$means is performed directly on $\mathbf{U}$, which may far deviate from the real discrete results. Thus, spectral rotation achieves superior performance than $k$means [20].

Fair spectral clustering groups the nodes of $\mathcal{G}$ by considering fairness. If the nodes belong to S sensitive groups $\mathcal{D}_1, \dots, \mathcal{D}_S$, where $\mathcal{D}_s$ contains the nodes of the sth sensitive group, we define the Balance of cluster $\mathcal{C}_k$ as [9]

$$\mathrm{Balance}(\mathcal{C}_k) = \min_{s\neq s'\in[S]} \frac{|\mathcal{D}_s \cap \mathcal{C}_k|}{|\mathcal{D}_{s'} \cap \mathcal{C}_k|} \in [0,1], \tag{5}$$

where $[S] := \{1, \dots, S\}$. The higher the Balance of each cluster, the fairer the clustering [9]. Eq. (5) indicates that the fairness is asking for a clustering where *the fraction of different sensitive groups in each cluster is approximately the same as that of the entire dataset* $\mathcal{V}$ [15], which is also called group fairness. To incorporate this fairness notion into SC, a group-membership vector $\mathbf{f}_s \in \{0,1\}^D$ of $\mathcal{D}_s$ is defined, where $(\mathbf{f}_s)_{[i]} = 1$ if $i \in \mathcal{D}_s$ and $(\mathbf{f}_s)_{[i]} = 0$ otherwise, for $s \in [S]$ and $i \in [D]$. Then, we have the following lemma.

Lemma 1 (*Fairness Constraint as Linear Constraint [15]*). *Let $\mathcal{V} = \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_K$ be a clustering . We have, for every $k \in [K]$*

$$\forall s \in [S]: \frac{|\mathcal{D}_s \cap \mathcal{C}_k|}{|\mathcal{C}_k|} = \frac{|\mathcal{D}_s|}{D} \approx \mathbf{F}^\top\mathbf{U} = \mathbf{0}, \tag{6}$$

where $\mathbf{F} \in \mathbb{R}^{D\times(S-1)}$ is a matrix satisfying $\mathbf{F}_{[:,s]} = \mathbf{f}_s - (|\mathcal{D}_s|/D) \cdot \mathbf{1}, s \in [S-1]$.

Lemma 1 states that the proportional representation of all sensitive attribute samples in each cluster can be approximated by a linear constraint $\mathbf{F}^\top\mathbf{U} = \mathbf{0}$. Under this fairness notion, unnormalized SC is equivalent to the following problem

$$\min_{\mathbf{U}} \mathrm{Tr}(\mathbf{U}^\top\mathbf{L}\mathbf{U}), \quad \text{s.t. } \mathbf{U}^\top\mathbf{U} = \mathbf{I},\ \mathbf{F}^\top\mathbf{U} = \mathbf{0}. \tag{7}$$

After solving (7), we can discrete $\mathbf{U}$ to obtain the cluster labels $\mathbf{Q}$.

4. Model formulation

In this section, we first theoretically analyze the impact of the constructed graph on FSC, which justifies an accurate graph for improving FSC performance. Then, we propose a novel graph construction method to learn graphs from potentially noisy observed data. Next, based on the graph construction method, we propose an end-to-end FSC framework. Finally, we analyze the connections between our model and existing works.

4.1. Why we need an accurate graph?

We first introduce a variant of the stochastic block model [45] (vSBM) to generate random graphs with cluster structures and sensitive attributes [15]. This model assumes that there are two or more meaningful ground-truth clusterings of the observed data, and only one of them is fair. Assume that $\mathcal{V}$ comprises S sensitive groups and is partitioned into K clusters such that $|\mathcal{D}_s \cap \mathcal{C}_k|/|\mathcal{C}_k| = \zeta_s, s \in [S], k \in [K]$, for $\zeta_s \in (0,1)$ with $\sum_{s=1}^{S} \zeta_s = 1$. Based on the clusters and sensitive groups, we construct a random graph by connecting two vertices i and j with a probability $\Pr(i,j)$ that depends on the clusters and sensitive groups of i and j. We define

$$\Pr(i,j) = \begin{cases} a, & \pi_C(i) = \pi_C(j),\ \pi_S(i) = \pi_S(j) \\ b, & \pi_C(i) \neq \pi_C(j),\ \pi_S(i) = \pi_S(j) \\ c, & \pi_C(i) = \pi_C(j),\ \pi_S(i) \neq \pi_S(j) \\ d, & \pi_C(i) \neq \pi_C(j),\ \pi_S(i) \neq \pi_S(j), \end{cases} \tag{8}$$

where $\pi_C : [D] \to [K]$ and $\pi_S : [D] \to [S]$ are two functions that assign a node $i \in \mathcal{V}$ to one of the clusters and sensitive groups, respectively. Let $\mathbf{L}^*$ be the real graph Laplacian matrix generated by the vSBM method and $\hat{\mathbf{L}}$ be the Laplacian matrix estimated by any graph construction method. The matrix $\hat{\mathbf{L}}$ is used as the input to fair spectral embedding in (7), and spectral rotation is utilized to obtain discrete cluster labels. Our goal is to derive a fair clustering error bound related to the estimation error between $\hat{\mathbf{L}}$ and $\mathbf{L}^*$. Let us make some assumptions.

Assumption 1. Let $\hat{\mathbf{U}}$ be a continuous cluster indicator matrix estimated from $\hat{\mathbf{L}}$ via (7). For a given constant $\epsilon_S > 0$, the $\hat{\mathbf{Q}}$ and $\hat{\mathbf{R}}$ estimated by spectral rotation satisfies

$$\|\hat{\mathbf{Q}} - \hat{\mathbf{U}}\hat{\mathbf{R}}\|_{\mathrm{F}}^2 \leq (1+\epsilon_S) \min_{\mathbf{Q}\in\mathcal{I},\mathbf{R}^\top\mathbf{R}=\mathbf{I}} \|\mathbf{Q} - \hat{\mathbf{U}}\mathbf{R}\|_{\mathrm{F}}^2. \tag{9}$$

Assumption 2. The ground-truth clustering and sensitive partitions of $\mathcal{V}$ satisfy

$$|\mathcal{D}_s| = \frac{D}{S},\ |\mathcal{C}_k| = \frac{D}{K},\ \frac{|\mathcal{D}_s \cap \mathcal{C}_k|}{|\mathcal{C}_k|} = \frac{1}{S}. \tag{10}$$

Assumption 1 is similar to the $(1+\epsilon_S)$–approximation of $k$means [46], which provides the estimation accuracy of spectral rotation. Assumption 2 is the same as that in Theorem 1 of [15], which is made to facilitate theoretical analysis. In practice, Assumption 2 may be violated, which, however, does not affect the effectiveness of FSC algorithms on the graph generated by the vSBM [15]. Based on the two assumptions, we have the following proposition.

Proposition 1. *Let $\mathbf{L}^*$ be the real Laplacian matrix of the random graph generated by the vSBM method with $a > b > c > d$ satisfying $a > r_1 \ln D/D$ for some $r_1 > 0$, and $\hat{\mathbf{L}}$ be the estimated Laplacian matrix from* *observed data whose estimation error is* $\epsilon_L = \|\mathbf{L}^* - \widehat{\mathbf{L}}\|_{\mathrm{F}}$*. Assume that we run fair spectral embedding (7) on* $\widehat{\mathbf{L}}$ *and perform* $(1+\epsilon_S)$ *spectral rotation (4) to obtain discrete cluster labels. Besides, let* $\widehat{\pi}_C(i)$ *be the assigned cluster label (after proper permutation) of node* i*, and define* $\mathcal{M}_k := \left\{i \in C_k : \widehat{\pi}_C(i) \neq k\right\}$ *as the set of misclassified vertices of cluster* k*. Under Assumptions 1–2, for every* $r_2 > 0$*, there exist constants* $\widehat{C} = \widehat{C}(r_1, r_2)$*,* $\widetilde{C} = \widetilde{C}(r_1, r_2)$*, and* $\bar{C} > 0$ *such that if*

$$\frac{aK^3 \ln D}{D(c-d)^2} \leq \frac{\widehat{C}}{1+\epsilon_S}, \tag{11}$$

then with probability at least $1 - D^{-r_2}$*, the number of misclassified vertices,* $\sum_{k=1}^{K} |\mathcal{M}_k|$*, is at most*

$$\underbrace{\frac{\widetilde{C}(1+\epsilon_S)aK^2 \ln D}{(c-d)^2}}_{\text{related to the vSBM}} + \underbrace{\frac{\bar{C}(1+\epsilon_S)\epsilon_L^2 K^2}{(c-d)^2}}_{\text{related to graph estimation}} . \tag{12}$$

Proof. The proof is inspired by [15] but has two main differences. First, spectral rotation instead of $k$means is used to obtain discrete labels. Second, [15] utilizes a known graph generated by the vSBM method to perform fair spectral embedding, while our method requires an estimated graph from raw data. Thus, Eq. (12) has an additional term related to graph estimation compared with [15]. See Appendix for details.

According to [15], the meaning of "the number of misclassified vertices is at most D_m" is that there exists a permutation of cluster indices such that the clustering results up to this permutation successfully predict all cluster labels but D_m many vertices. Note that the error bound consists of two parts. The first one is caused by the difference between the expected and real graph produced by the vSBM method, which is similar to [15]. The second part is related to the estimation error ϵ_L of graph construction methods. The fair constraint affects clustering performance via $\mathbf{Z}$, which is a matrix determined by sensitive group-membership matrix $\mathbf{F}$. Generally, the error bound in (12) depends on K, D and ϵ_S. If we divide (12) by D, we obtain the bound for the misclassification rate. The first part of the misclassification rate bound tends to zero as D goes to infinity, meaning that if $\mathbf{L}^*$ is exactly estimated (the second part equals to zero), performing FSC via (7) and spectral rotation is weakly consistent [15]. However, $\mathbf{L}^*$ usually cannot be estimated exactly, causing an additional error for subsequent fair clustering results. Proposition 1 illustrates that a well-estimated graph $\widehat{\mathbf{L}}$, which is close to $\mathbf{L}^*$, brings a small misclassification error bound. Thus, it motivates us to seek an effective method to construct accurate graphs from observed data.

4.2. The proposed graph construction method

Given N observed data $\mathbf{X}_o \in \mathbb{R}^{D \times N}$, we need to infer the underlying similarity graph topology as the input to FSC algorithms. However, contaminated data may lead to poor graph estimation performance, as indicated in Proposition 1, which degrades subsequent fair clustering performance. Therefore, we propose a method to learn graphs from potentially noisy data $\mathbf{X}_o$, which is formulated as

$$\min_{\mathbf{L}\in\mathcal{L},\mathbf{X},\boldsymbol{\upsilon}>0} \frac{1}{N}\|\boldsymbol{\Upsilon}(\mathbf{X}_o - \mathbf{X})\|_{\mathrm{F}}^2 + \frac{\xi}{N}\mathrm{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) + \sum_{i=1}^{D}\frac{1}{\boldsymbol{\upsilon}_{[i]}} \\ \underbrace{-\mathbf{1}^\top \log\left(\mathrm{diag}(\mathbf{L})\right) + \beta\|\mathrm{diag}_0(\mathbf{L})\|_{\mathrm{F}}^2}_{Reg(\mathbf{L})}, \tag{13}$$

where $\mathcal{L} := \left\{\mathbf{L} : \mathbf{L} \in \mathbb{S}^{D\times D}, \mathbf{L}\mathbf{1} = \mathbf{0}, \mathbf{L}_{[ij]} \leq 0,\ i \neq j\right\}$ contains all Laplacian matrices. Moreover, ξ and β are parameters, and $\boldsymbol{\upsilon} \in \mathbb{R}^D$ is a vector of adaptive weights. We let $\boldsymbol{\Upsilon} := \mathrm{diag}(\sqrt{\boldsymbol{\upsilon}})$, where $\sqrt{\boldsymbol{\upsilon}} = (\sqrt{\boldsymbol{\upsilon}_{[1]}}, \ldots, \sqrt{\boldsymbol{\upsilon}_{[D]}})^\top$. Eq. (13) is a joint model of denoising and smoothness-based GL [35], which will be explained in detail next.

(1) Denoising: If $\mathbf{L}$ is fixed, the problem (13) becomes

$$\min_{\mathbf{X},\boldsymbol{\upsilon}} \frac{1}{N}\|\boldsymbol{\Upsilon}(\mathbf{X}_o - \mathbf{X})\|_{\mathrm{F}}^2 + \frac{\xi}{N}\mathrm{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) + \sum_{i=1}^{D}\frac{1}{\boldsymbol{\upsilon}_{[i]}}. \tag{14}$$

The model is a node-adaptive graph filter, and $\boldsymbol{\upsilon}$ represents node weights. Specifically, given node weights $\boldsymbol{\upsilon}$, we have

$$\min_{\mathbf{X}} \|\boldsymbol{\Upsilon}(\mathbf{X}_o - \mathbf{X})\|_{\mathrm{F}}^2 + \xi\mathrm{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}). \tag{15}$$

Taking the derivative of (15) and setting it to zero, we obtain

$$\mathbf{X} = \left(\boldsymbol{\Upsilon}^\top\boldsymbol{\Upsilon} + \xi\mathbf{L}\right)^{-1}\boldsymbol{\Upsilon}^\top\boldsymbol{\Upsilon}\mathbf{X}_o = \left(\mathbf{I} + \xi(\boldsymbol{\Upsilon}^\top\boldsymbol{\Upsilon})^{-1}\mathbf{L}\right)^{-1}\mathbf{X}_o. \tag{16}$$

We let $\mathbf{K} := \left(\mathbf{I} + \xi(\boldsymbol{\Upsilon}^\top\boldsymbol{\Upsilon})^{-1}\mathbf{L}\right)^{-1}$, which is positive definite and has eigen-decomposition $\mathbf{K} = \boldsymbol{\Theta}\boldsymbol{\Lambda}\boldsymbol{\Theta}^\top$ with eigenvalues matrix $\boldsymbol{\Lambda}$ and eigenvectors matrix $\boldsymbol{\Theta}$. Moreover, $\boldsymbol{\Lambda} = \mathrm{diag}\left(\frac{1}{1+\xi\lambda_1}, \ldots., \frac{1}{1+\xi\lambda_D}\right)$, where $0 = \lambda_1 \leq, \ldots., \leq \lambda_D$ are the eigenvalues of $(\boldsymbol{\Upsilon}^\top\boldsymbol{\Upsilon})^{-1}\mathbf{L}$. From the perspective of graph spectral filtering (GFT) [33], $\mathbf{K}\mathbf{X}_o = \boldsymbol{\Theta}\boldsymbol{\Lambda}\boldsymbol{\Theta}^\top\mathbf{X}_o$ can be interpreted as that the observed graph signals (columns of $\mathbf{X}_o$) are first transformed to the graph frequency domain via $\boldsymbol{\Theta}^\top$, attenuated GFT coefficients according to $\boldsymbol{\Lambda}$, and transformed back to the nodal domain via $\boldsymbol{\Theta}$. It is observed from $\boldsymbol{\Lambda}$ that the graph filter $\mathbf{K}$ is low-pass since the attenuation is stronger for larger eigenvalues. Thus, the graph filter can suppress the high-frequency component of raw data $\mathbf{X}_o$ corresponding to the noise.

Our graph filter $\mathbf{K}$ differs from the Auto-Regressive graph filter $(\mathbf{I} + \xi\mathbf{L})^{-1}$ [47] in that we assign each node an individual weight $\boldsymbol{\upsilon}_{[i]}, i = 1, \ldots, D$. The reason for using $\boldsymbol{\upsilon}$ is that the measurement noise of different nodes may be heterogeneous. If the ith node signal (the ith row of $\mathbf{X}_o$) has a small noise scale, a large $\boldsymbol{\upsilon}_{[i]}$ should be assigned to the fidelity term of node i in (14) to ensure $\mathbf{X}_{[i,:]}$ is close to the corresponding observation $(\mathbf{X}_o)_{[i,:]}$ [48]. When we cannot know the noise scale a priori, we can adaptively learn $\boldsymbol{\upsilon}$ from the data. Specifically, given $\mathbf{X}$, the problem (14) becomes

$$\min_{\boldsymbol{\upsilon}>0} \frac{1}{N}\sum_{i=1}^{D}\boldsymbol{\upsilon}_{[i]}\|(\mathbf{X}_o)_{[i,:]} - \mathbf{X}_{[i,:]}\|_2^2 + \frac{1}{\boldsymbol{\upsilon}_{[i]}}. \tag{17}$$

The last term of (17) is used to avoid trivial solutions. Intuitively, solving (17) will assign a large $\boldsymbol{\upsilon}_{[i]}$ to node i if $\mathbf{X}_{[i,:]}$ is close to $(\mathbf{X}_o)_{[i,:]}$, as expected.

(2) Graph learning: If we have obtained the "noiseless" signals $\mathbf{X}$ via the graph filter $\mathbf{K}$, the problem (13) becomes

$$\min_{\mathbf{L}\in\mathcal{L}} \frac{\xi}{N}\mathrm{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) + Reg(\mathbf{L}). \tag{18}$$

The first Laplacian quadratic term of (18) is equivalent to

$$\frac{1}{N}\mathrm{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) = \frac{1}{N}\sum_{n=1}^{N}\sum_{i,j=1}^{D}\mathbf{W}_{[ij]}\left(\mathbf{X}_{[in]} - \mathbf{X}_{[jn]}\right)^2, \tag{19}$$

which measures the average smoothness of data $\mathbf{X}$ over the graph $\mathbf{L}$ [35]. The second term of (18) contains regularizers that endow the learned graphs with desired properties. The $\log$-degree term is to control node degree, and the Frobenius norm term is to control graph sparsity. Our model (18) can learn a graph suitable for graph-based clustering tasks for the following reasons. (i) It is observed from (19) that minimizing the smoothness is to seek a graph whose similar vertices (node signals) are closely connected, which is consistent with the fundamental principle of SC. (ii) The $\log$-degree term can avoid isolated nodes, which is crucial for SC, especially for normalized SC [5]. (iii) The Frobenius norm term can lead to a sparse graph, which may remove redundant and noisy edges.

The model (18) is similar to the ANGL method [28] since both construct graphs by minimizing the smoothness. The main differences lie in three aspects. (i) Our model removes the sum-to-one constraint in the ANGL method—the degree of each node is forced to be one—since the constraint makes the output graphs sensitive to noisy points [32]. Removing this constraint allows our model to capture more complex

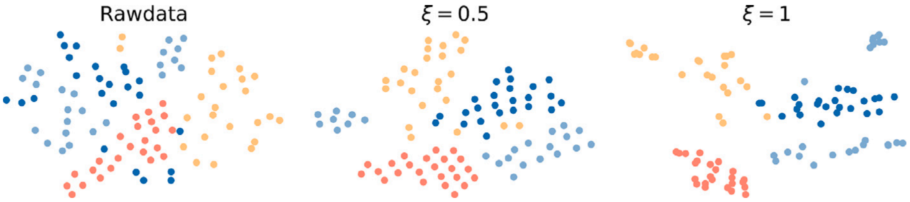


Fig. 2. The t-SNE results of MNIST with different ξ values.

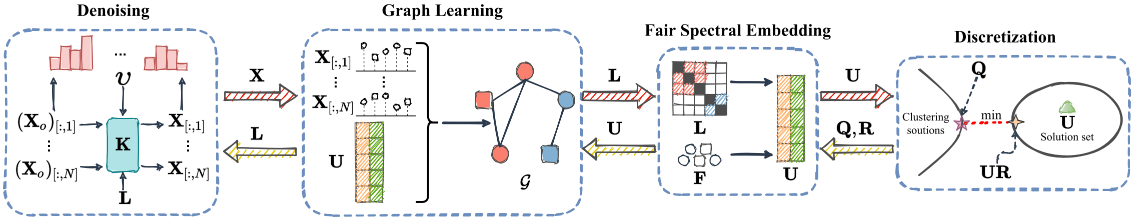


Fig. 3. The illustration of the proposed framework.

similarity relationships. (ii) We add a log-degree term to ensure the learned graph has no isolated nodes. (iii) The input data of (18) are those produced by a low-pass graph filter.

(3) Discussion: We try to explain why our method is effective in constructing graphs from observed data. If data $\mathbf{X}_o$ have a clustering structure, they should follow the assumption of cluster and manifold, i.e., the data in the same cluster are close to each other. According to [49], smooth signals containing low-frequency parts tend to follow the cluster and manifold assumption. Thus, if $\mathbf{L}$ accurately represents the graph behind observed data, the denoising part of our model has two functions. First, it filters out the high-frequency components of the observed graph signals that correspond to noises. Second, it produces smooth signals that have a clearer clustering structure, which could facilitate subsequent clustering. To better illustrate the effectiveness of the node-adaptive filter, Fig. 2 depicts the t-SNE [50] results of our methods on the MNIST dataset, where four clusters correspond to four randomly selected digits. We can see that raw data are entangled together. In contrast, the $\mathbf{X}$ output by the graph filter is clearly separated, meaning that the denoising part of our model can produce cluster-friendly signals. On the other hand, from the perspective of GL, our model (18) learns a graph minimizing the smoothness of data, i.e., the nodes corresponding to similar signals are closely connected. Therefore, the learned graph can effectively capture similarity relationships between data and preserve clustering structures, thereby improving the performance of graph filters. In summary, the denoising part can provide "noiseless" and smooth signals for graph construction, while graph learning provides accurate graphs to better denoise graph signals. The two steps can enhance mutually to bring a high-quality graph for subsequent fair clustering tasks. In the literature, there exist works like [51] which also simultaneously recover and cluster corrupted data. Our model differs from them in the proposed denoising and graph learning methods.

4.3. The unified FSC model

In this subsection, we build an end-to-end FSC framework that inputs observed data $\mathbf{X}_o$ and sensitive group membership matrix $\mathbf{F}$ and directly outputs discrete cluster labels. As shown in Fig. 3, our model consists of four tasks, i.e., denoising, graph learning, fair spectral embedding, and discretization. Integrating all the subtasks into a single objective function, we obtain

$$\begin{aligned}\min_{\mathbf{X},\mathbf{L},\boldsymbol{\upsilon},\mathbf{Y},\mathbf{R},\mathbf{Q}} \ & \frac{1}{N}\|\boldsymbol{\Upsilon}(\mathbf{X}_o-\mathbf{X})\|_{\mathrm{F}}^2+\frac{\xi}{N}\mathrm{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X})+Reg(\mathbf{L}) \\ & +\sum_{i=1}^{D}\frac{1}{\boldsymbol{\upsilon}_{[i]}}+\mu\mathrm{Tr}(\mathbf{U}^\top\mathbf{L}\mathbf{U})+\gamma\|\mathbf{Q}-\mathbf{U}\mathbf{R}\|_{\mathrm{F}}^2 \\ \text{s.t. } & \mathbf{L}\in\mathcal{L},\boldsymbol{\upsilon}>0,\mathbf{U}^\top\mathbf{U}=\mathbf{I},\mathbf{F}^\top\mathbf{U}=\mathbf{0},\mathbf{R}^\top\mathbf{R}=\mathbf{I},\mathbf{Q}\in\mathcal{I},\end{aligned} \tag{20}$$

where μ and γ are two parameters. In (20), the four subtasks are bridged by two Laplacian quadratic terms, i.e., $\frac{\xi}{N}\mathrm{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X})$ and $\mu\mathrm{Tr}(\mathbf{U}^\top\mathbf{L}\mathbf{U})$. Therefore, the model (20) is dubbed the dual Laplacian framework. Specifically, the first Laplacian term $\frac{\xi}{N}\mathrm{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X})$ is a bridge between denoising and graph construction. On one hand, it can be viewed as the graph Tikhonov regularizer of the denoising task (15) to output smooth signals [47]. On the other hand, it measures smoothness in the GL task to capture the similarity relationships between data. The second term $\mu\mathrm{Tr}(\mathbf{U}^\top\mathbf{L}\mathbf{U})$ connects graph construction, fair spectral embedding and discretization. It imposes structural constraints on the constructed graph, which will be discussed later, and provides low-dimensional embeddings for the discretization task. Here, we leverage spectral rotation (4) instead of $k$means to obtain discrete cluster labels due to its superior performance. The merit of the dual Laplacian framework is that all subtasks are jointly optimized and hence can be mutually enhanced. In the previous subsection, we discussed the role of the first Laplacian term in learning graphs suitable for clustering. Next, we explore how the second Laplacian term can further improve graph learning performance.

We introduce a new variable matrix $\mathbf{Y}\in\mathbb{R}^{(D-S+1)\times K}$ and let $\mathbf{U}=\mathbf{Z}\mathbf{Y}$, where $\mathbf{Z}\in\mathbb{R}^{D\times(D-S+1)}$ is a matrix whose columns form the orthonormal basis of the nullspace of $\mathbf{F}^\top$. The matrix $\mathbf{F}$ encodes sensitive information, as does $\mathbf{Z}$. Then, (20) is rephrased in term of $\mathbf{Y}$ as

$$\begin{aligned}\min_{\mathbf{X},\mathbf{L},\boldsymbol{\upsilon},\mathbf{Y},\mathbf{R},\mathbf{Q}} \ & \frac{1}{N}\|\boldsymbol{\Upsilon}(\mathbf{X}_o-\mathbf{X})\|_{\mathrm{F}}^2+\frac{\xi}{N}\mathrm{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X})+Reg(\mathbf{L}) \\ & +\sum_{i=1}^{D}\frac{1}{\boldsymbol{\upsilon}_{[i]}}+\mu\mathrm{Tr}(\mathbf{Y}^\top\mathbf{Z}^\top\mathbf{L}\mathbf{Z}\mathbf{Y})+\gamma\|\mathbf{Q}-\mathbf{Z}\mathbf{Y}\mathbf{R}\|_{\mathrm{F}}^2 \\ \text{s.t. } & \mathbf{L}\in\mathcal{L},\boldsymbol{\upsilon}>0,\mathbf{Y}^\top\mathbf{Y}=\mathbf{I},\mathbf{R}^\top\mathbf{R}=\mathbf{I},\mathbf{Q}\in\mathcal{I}.\end{aligned} \tag{21}$$

In (21), the fairness constraint $\mathbf{F}^\top\mathbf{U}=\mathbf{0}$ is removed. We conduct spectral embedding on $\mathbf{Z}^\top\mathbf{L}\mathbf{Z}$, which is dubbed fair graph. The fair graph encodes graph topology and sensitive information simultaneously. If we only focus on GL and fair spectral embedding, problem (21) boils down to

$$\begin{aligned}\min_{\mathbf{L},\mathbf{Y}} \ & \frac{\xi}{N}\mathrm{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X})+Reg(\mathbf{L})+\mu\mathrm{Tr}(\mathbf{Y}^\top\mathbf{Z}^\top\mathbf{L}\mathbf{Z}\mathbf{Y}) \\ \text{s.t. } & \mathbf{L}\in\mathcal{L},\mathbf{Y}^\top\mathbf{Y}=\mathbf{I},\end{aligned} \tag{22}$$

where $\mathbf{X}$ is regarded as the "noiseless" data here. According to Ky Fan's theorem [52], we have $\min_{\mathbf{Y}^\top\mathbf{Y}=\mathbf{I}}\mathrm{Tr}(\mathbf{Y}^\top\mathbf{Z}^\top\mathbf{L}\mathbf{Z}\mathbf{Y})=\sum_{k=1}^{K}\widetilde{\lambda}_k$, where

$\widetilde{\lambda}_k$ is the k smallest eigenvalue of $\mathbf{Z}^\top\mathbf{L}\mathbf{Z}$. Thus, the problem (22) can be rephrased as

$$\min_{\mathbf{L}\in\mathcal{L}} \frac{\xi}{N}\mathrm{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) + Reg(\mathbf{L}) + \mu\sum_{k=1}^{K}\widetilde{\lambda}_k. \tag{23}$$

Note that $\mathbf{Z}^\top\mathbf{L}\mathbf{Z}$ is a semi-positive definite matrix, i.e., $\widetilde{\lambda}_k \geq 0$. Minimizing (23) is equivalent to forcing $\sum_{k=1}^{K}\widetilde{\lambda}_k \to 0$ if μ is large enough. That is, (23) encourages the fair graph to have K connected components, which provides structural constraints for the graph construction task. Therefore, the second Laplacian term allows the learned graph to better capture fair clusters among data, which in turn improves the fair clustering performance.

4.4. Connections to existing works

(1) Connections to community-based GL models: If we only consider the graph construction task, (23) can be viewed as the widely studied community-based GL model, the key of which is to design the community structure constraint. Suppose λ_k is the k smallest eigenvalues of $\mathbf{L}$. The work [53] lets the first K smallest eigenvalues of Laplacian matrices be zero, i.e., $\lambda_k = 0, k = 1,\ldots,K$. Moreover, [28,54] force the graph satisfying $\mathrm{rank}(\mathbf{L}) = D - K$. The two constraints can be relaxed to minimizing $\sum_{k=1}^{K}\lambda_k$, which is similar to the last term in (23). Although closely related, (22) differs from existing community-based GL models in two key aspects. First, the basic GL models are different. Our model is based on the smoothness-based GL, while [53] are based on statistical GL models like Graphical Lasso. Besides, [28,54] are based on the ANGL method. Second, our model imposes the community constraint on the fair graph $\mathbf{Z}^\top\mathbf{L}\mathbf{Z}$ instead of on $\mathbf{L}$ like existing works. Therefore, our approach seeks a graph that can describe the community structure while satisfying fairness requirements. In the experimental section, we will show that the additional fairness constraint can change the cluster structure of the learned graph to ensure fairness requirements.

(2) Connections to unified SC models: If we remove the denoising module and the fairness constraint, our model (20) becomes

$$\begin{aligned}\min_{\mathbf{L},\mathbf{U},\mathbf{R},\mathbf{Q}}\ & \frac{\xi}{N}\mathrm{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) + Reg(\mathbf{L}) + \mu\mathrm{Tr}(\mathbf{U}^\top\mathbf{L}\mathbf{U})\\ & + \gamma\|\mathbf{Q}-\mathbf{U}\mathbf{R}\|_F^2\\ \text{s.t. } & \mathbf{L}\in\mathcal{L}, \mathbf{U}^\top\mathbf{U} = \mathbf{I}, \mathbf{R}^\top\mathbf{R} = \mathbf{I}, \mathbf{Q}\in\mathcal{I}.\end{aligned} \tag{24}$$

Again, $\mathbf{X}$ is treated as the observed data. The model (24) is an end-to-end SC model. Here, we discuss the connections between our model and those unified SC models integrating graph construction, spectral embedding, and discretization. As stated before, we focus on basic formulations without additional extensions. The first model we compare is [21]

$$\begin{aligned}\min_{\mathbf{W},\mathbf{U},\mathbf{Q},\mathbf{R}}\ & \|\mathbf{X}-\mathbf{W}^\top\mathbf{X}\|_F^2 + \alpha_U\|\mathbf{W}\|_{1,1} + \mu_U\mathrm{Tr}(\mathbf{U}^\top\mathbf{L}\mathbf{U})\\ & + \gamma_U\|\mathbf{Q}-\mathbf{U}\mathbf{R}\|_F^2\\ \text{s.t. } & \mathbf{W}\in\mathcal{W}, \mathbf{U}^\top\mathbf{U} = \mathbf{I}, \mathbf{R}^\top\mathbf{R} = \mathbf{I}, \mathbf{Q}\in\mathcal{I},\end{aligned} \tag{25}$$

where $\mathcal{W} = \left\{\mathbf{W} : \mathbf{W}\in\mathbb{S}^{D\times D}, \mathbf{W}\geq 0, \mathrm{diag}(\mathbf{W}) = \mathbf{0}\right\}$ is the set containing all adjacency matrices. Moreover, α_U, μ_U, and γ_U are constant parameters. This is a unified SC model that leverages the sparse representation method [25] to construct graphs, which is different from our GL method.

Another unified SC model [24,42] is concluded as

$$\begin{aligned}\min_{\mathbf{W},\mathbf{U},\mathbf{Q},\mathbf{R}}\ & \sum_{i,j=1}^{D}\|\mathbf{X}_{[i,:]}-\mathbf{X}_{[j,:]}\|_2^2\mathbf{W}_{[i,j]} + \beta_J\mathbf{W}_{[i,j]}^2\\ & + \mu_J\mathrm{Tr}(\mathbf{U}^\top\mathbf{L}\mathbf{U}) + \gamma_J\|\mathbf{Q}-\mathbf{U}\mathbf{R}\|_F^2\\ \text{s.t. } & \mathbf{W}\mathbf{1} = \mathbf{1}, \mathbf{W}\geq 0, \mathbf{L} = \mathbf{D}-\mathbf{W}, \mathbf{U}^\top\mathbf{U} = \mathbf{I},\\ & \mathbf{R}^\top\mathbf{R} = \mathbf{I}, \mathbf{Q}\in\mathcal{I},\end{aligned} \tag{26}$$

where β_J, μ_J, and γ_J are constant parameters. The graph construction method in (26) is the ANGL method [28]. We have discussed the difference between our graph construction method and the ANGL in the previous subsection.

Algorithm 1 The algorithm for problem (30)

```
Require: $\beta, \mathbf{p}$, set $L = \frac{D-1}{2\beta}$
Ensure: The learned graph $\mathbf{w}$
Initialize $\eta^{(1)} = 1$ and $\boldsymbol{\omega}^{(1)} = \mathbf{r}^{(0)} \in \mathbb{R}^D$ at random
for $t = 1, 2, ...,$ do
    $\bar{\mathbf{w}}^{(t)} = \max\left(\frac{\mathbf{S}^\top\boldsymbol{\omega}^{(t)}-2\mathbf{p}}{4\beta}, 0\right)$
    $\mathbf{v}^{(t)} = \frac{\mathbf{S}\bar{\mathbf{w}}^{(t)}-L\boldsymbol{\omega}^{(t)}+\sqrt{(\mathbf{S}\bar{\mathbf{w}}^{(t)}-L\boldsymbol{\omega}^{(t)})^2+4L\mathbf{1}}}{2}$
    $\mathbf{r}^{(t)} = \boldsymbol{\omega}^{(t)} - L^{-1}\left(\mathbf{S}\bar{\mathbf{w}}^{(t)} - \mathbf{v}^{(t)}\right)$
    $\eta^{(t+1)} = \frac{1+\sqrt{1+4(\eta^{(t)})^2}}{2}$
    $\boldsymbol{\omega}^{(t+1)} = \mathbf{r}^{(t)} + \left(\frac{\eta^{(t)}-1}{\eta^{(t+1)}}\right)\left(\mathbf{r}^{(t)} - \mathbf{r}^{(t-1)}\right)$
end for
return $\mathbf{w} = \max\left(\frac{\mathbf{S}^\top\mathbf{r}^{(t)}-2\mathbf{p}}{4\beta}, 0\right)$
```

In summary, our model (24) differs from the existing unified SC models (25)–(26) mainly in the graph construction method. As Proposition 1 states, an accurate GL method can boost fair clustering performance. In Section 6, we develop fair versions of (25)–(26) and compare them with our model (20) to illustrate the superiority of our model.

5. Model optimization

In this section, we first propose an algorithm for solving (20), followed by convergence and complexity analyses.

5.1. Optimization algorithm

Our algorithm alternately updates $\mathbf{L}, \mathbf{U}, \mathbf{R}$, $\mathbf{Q}$, $\mathbf{X}$, and $\boldsymbol{v}$ in (20), i.e., updating one with the others fixed. For clarity, we omit the iteration index here. The following derivations are the updates in one iteration.

(1) Update $\mathbf{L}$*:* The sub-problem of updating $\mathbf{L}$ is

$$\min_{\mathbf{L}\in\mathcal{L}} \frac{\xi}{N}\mathrm{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) + Reg(\mathbf{L}) + \mu\mathrm{Tr}(\mathbf{U}^\top\mathbf{L}\mathbf{U}). \tag{27}$$

The problem can be rewritten in terms of $\mathbf{W}$

$$\min_{\mathbf{W}\in\mathcal{W}} \frac{1}{2}\|\mathbf{W}\circ\mathbf{P}\|_{1,1} + Reg_W(\mathbf{W}), \tag{28}$$

where

$$\mathbf{P}_{[ij]} = \frac{\xi}{N}\|\mathbf{X}_{[i,:]}-\mathbf{X}_{[j,:]}\|_2^2 + \mu\|\mathbf{U}_{[i,:]}-\mathbf{U}_{[j,:]}\|_2^2, \tag{29}$$

and $Reg_W(\mathbf{W}) = -\mathbf{1}^\top\log(\mathbf{W}\mathbf{1}) + \beta\|\mathbf{W}\|_F^2$. By the definition of $\mathcal{W}$, the free variables of $\mathbf{W}$ are the upper triangle elements. Thus, we define a vector $\mathbf{w}\in\mathbb{R}^P, P := \frac{D(D-1)}{2}$, satisfying that $\mathbf{w} = \mathrm{Triu}(\mathbf{W})$, where $\mathrm{Triu}(\cdot) : \mathbb{R}^{D\times D} \to \mathbb{R}^P$ is a function that converts the upper triangular elements of a matrix into a vector. Then, problem (28) is equivalent to

$$\min_{\mathbf{w}\geq 0} \mathbf{p}^\top\mathbf{w} - \mathbf{1}^\top\log(\mathbf{S}\mathbf{w}) + 2\beta\|\mathbf{w}\|_2^2, \tag{30}$$

where $\mathbf{p} = \mathrm{Triu}(\mathbf{P})$, $\mathbf{S}\in\mathbb{R}^{D\times P}$ is a linear operator satisfying $\mathbf{S}\mathbf{w} = \mathbf{W}\mathbf{1}$ [35]. The problem (30) is convex, and we employ the algorithm in [55] to solve the problem. The complete algorithm flow is presented in Algorithm 1. After obtaining the estimated $\mathbf{w}$, we let $\mathbf{W} = \mathrm{iTriu}(\mathbf{w})$, where $\mathrm{iTriu}(\cdot) : \mathbb{R}^P \to \mathbb{R}^{D\times D}$ is the inverse Triu operation. The operation $\mathrm{iTriu}(\mathbf{w})$ converts $\mathbf{w}$ into an adjacency matrix, where $\mathbf{w}$ corresponds to the upper triangle elements of $\mathbf{W}$. Finally, we calculate the Laplacian matrix from $\mathbf{W}$ and feed it into subsequent updates of other variables.

(2) Update $\mathbf{U}$*:* The sub-problem of updating $\mathbf{U}$ is

$$\begin{aligned}&\min_{\mathbf{U}} \mu \mathrm{Tr}\left(\mathbf{U}^\top \mathbf{L}\mathbf{U}\right) + \gamma \|\mathbf{Q}-\mathbf{U}\mathbf{R}\|_F^2 \\ &\text{s.t. } \mathbf{U}^\top\mathbf{U}=\mathbf{I}, \mathbf{F}^\top\mathbf{U}=\mathbf{0}.\end{aligned} \tag{31}$$

Like (21), (31) can be cast into a problem of variable $\mathbf{Y}$

$$\begin{aligned}&\min_{\mathbf{Y}^\top\mathbf{Y}=\mathbf{I}} \mu \mathrm{Tr}\left(\mathbf{Y}^\top\mathbf{Z}^\top\mathbf{L}\mathbf{Z}\mathbf{Y}\right) + \gamma\|\mathbf{Q}-\mathbf{Z}\mathbf{Y}\mathbf{R}\|_F^2 \\ \Leftrightarrow &\min_{\mathbf{Y}^\top\mathbf{Y}=\mathbf{I}} \mu \mathrm{Tr}\left(\mathbf{Y}^\top\mathbf{Z}^\top\mathbf{L}\mathbf{Z}\mathbf{Y}\right) - 2\gamma \mathrm{Tr}(\mathbf{R}\mathbf{Q}^\top\mathbf{Z}\mathbf{Y}).\end{aligned} \tag{32}$$

This is a typical quadratic optimization problem with orthogonal constraints. Let $\phi(\mathbf{Y})$ be the objective function of (32). We have that $\phi(\mathbf{Y})$ is differential, and $\nabla_{\mathbf{Y}}\phi(\mathbf{Y}) = 2\mu\mathbf{Z}^\top\mathbf{L}\mathbf{Z}\mathbf{Y} - 2\gamma\mathbf{Z}^\top\mathbf{Q}\mathbf{R}^\top$. Thus, the problem can be efficiently solved via the algorithm in [56]. After obtaining $\mathbf{Y}$, we let $\mathbf{U}=\mathbf{Z}\mathbf{Y}$.

(3) Update $\mathbf{R}$*:* The sub-problem of updating $\mathbf{R}$ is

$$\min_{\mathbf{R}^\top\mathbf{R}=\mathbf{I}} \gamma\|\mathbf{Q}-\mathbf{U}\mathbf{R}\|_F^2 \Leftrightarrow \max_{\mathbf{R}^\top\mathbf{R}=\mathbf{I}} \mathrm{Tr}(\mathbf{Q}^\top\mathbf{U}\mathbf{R}). \tag{33}$$

It is the orthogonal Procrustes problem with a closed-form solution [57]. Assuming that $\mathbf{\Theta}_L$ and $\mathbf{\Theta}_R$ are the left and right matrices of SVD of $\mathbf{Q}^\top\mathbf{U}$, the solution to (33) is $\mathbf{R} = \mathbf{\Theta}_R\mathbf{\Theta}_L^\top$ [57].

(4) Update $\mathbf{Q}$*:* The sub-problem of updating $\mathbf{Q}$ is

$$\begin{aligned}&\min_{\mathbf{Q}\in\mathcal{I}} \gamma\|\mathbf{Q}-\mathbf{U}\mathbf{R}\|_F^2 \\ \Leftrightarrow &\min_{\mathbf{Q}\in\mathcal{I}} \gamma\mathrm{Tr}(\mathbf{Q}^\top\mathbf{Q}) - 2\gamma\mathrm{Tr}(\mathbf{Q}^\top\mathbf{U}\mathbf{R}) \\ \Leftrightarrow &\max_{\mathbf{Q}\in\mathcal{I}} \mathrm{Tr}(\mathbf{Q}^\top\mathbf{U}\mathbf{R}).\end{aligned} \tag{34}$$

The optimal solution to (34) is as follows:

$$\mathbf{Q}_{[ik]} = \begin{cases} 1, & k = \mathrm{argmax}_{j\in[K]} (\mathbf{U}\mathbf{R})_{[ij]}, \\ 0, & \text{others}. \end{cases} \tag{35}$$

(5) Update $\mathbf{X}$*:* The sub-problem of updating $\mathbf{X}$ is (14), which has a closed solution (16). However, matrix inversion is computationally expensive with complexity $\mathcal{O}(D^3)$. Fortunately, $\left(\mathbf{\Upsilon}^\top\mathbf{\Upsilon}+\xi\mathbf{L}\right)^{-1}$ is symmetric, sparse, and positive definite. We can hence solve (14) efficiently using conjugate gradient (CG) algorithm without matrix inverse [58].

(6) Update $\boldsymbol{\upsilon}$*:* The sub-problem of updating $\boldsymbol{\upsilon}$ is (17). Taking the derivative of (17) and setting it to zero, we have

$$\boldsymbol{\upsilon}_{[i]} = \frac{\sqrt{N}}{\|(\mathbf{X}_o)_{[i,:]} - \mathbf{X}_{[i,:]}\|_2}, \quad i = 1,\dots,D. \tag{36}$$

It is observed that the updates of $\mathbf{L}$, $\mathbf{U}$, $\mathbf{R}$, $\mathbf{Q}$, $\mathbf{X}$, and $\boldsymbol{\upsilon}$ are coupled with each other. Updating one variable depends on the other variables, leading to an overall optimal solution. The complete procedure is presented in Algorithm 2.

5.2. Convergence and complexity analysis

(1) Convergence analysis: It is challenging to obtain a globally optimal solution to (20) since it is not jointly convex for all variables. However, our algorithm for solving each sub-problem can reach its optimal solution. Specifically, when we update $\mathbf{L}$, the problem (29) is convex, and the corresponding algorithm is guaranteed to converge to the global optimum [55]. When updating $\mathbf{U}$, we use the algorithm in [56] to solve the problem (32), which can converge to the global optimum [56]. The updates of $\mathbf{Q}$, $\mathbf{R}$ and $\boldsymbol{\upsilon}$ have closed-form solutions. Despite updating $\mathbf{X}$ via (15) has a closed solution, we update $\mathbf{X}$ using CG, which is guaranteed to converge [58]. In summary, the update of each variable converges in our algorithm. In reality, the whole algorithm converges well, which is verified in Section 6.

(2) Complexity analysis: In one iteration, our algorithm consists of six parts, which we analyze one by one below. As stated in [55], the update of $\mathbf{L}$ requires $\mathcal{O}(T_1 D^2)$ costs, where T_1 is the average number of iterations of updating $\mathbf{w}$. The computational cost can be further reduced if the average number of neighbors per node is fixed; see [59] and analysis therein. The computational complexity of our algorithm for updating $\mathbf{U}$ is $\mathcal{O}(T_2(DK^2+K^3))$ according to [56], where T_2 is the average number of iterations of the algorithm in [56]. When updating $\mathbf{R}$, we perform SVD on $\mathbf{Q}^\top\mathbf{U}\in\mathbb{R}^{K\times K}$, which costs $\mathcal{O}(K^3)$. The updates of $\mathbf{Q}$ and $\boldsymbol{\upsilon}$ require $\mathcal{O}(DK^2)$ and $\mathcal{O}(DN)$, respectively. Finally, the complexity of using CG to update $\mathbf{X}$ is $\mathcal{O}(T_3 DN)$, where T_3 is the average number of iterations of the CG algorithm.

Algorithm 2 The algorithm for problem (20).

```
Require: Data matrix $\mathbf{X}_o \in \mathbb{R}^{D\times N}$, sensitive attributes related matrix $\mathbf{F}$
    or $\mathbf{Z}$, the number of clusters $K$, model parameters $\xi, \beta, \mu$, and $\gamma$
Ensure: The learned $\mathbf{L}$ and discrete cluster labels $\mathbf{Q}$
Initialize $\mathbf{L}$, $\mathbf{U}$, $\mathbf{Q}$, and $\mathbf{R}$ randomly, $\mathbf{X}=\mathbf{X}_o$, and $\boldsymbol{\upsilon}=\mathbf{1}$
while not converged do
    Calculate $\mathbf{P}$ by (29) and let $\mathbf{p} = \mathrm{Triu}(\mathbf{P})$
    Update $\mathbf{w}$ by solving (30)
    Convert $\mathbf{W} = \mathrm{iTriu}(\mathbf{w})$ and calculate $\mathbf{L} = \mathbf{D} - \mathbf{W}$
    Update $\mathbf{Y}$ by solving problem (32) using the algorithm in [56],
        and let $\mathbf{U} = \mathbf{Z}\mathbf{Y}$
    Update $\mathbf{R} = \mathbf{\Theta}_R\mathbf{\Theta}_L^\top$, where $\mathbf{\Theta}_L$ and $\mathbf{\Theta}_R$ are the left and right
        matrices of SVD of $\mathbf{Q}^\top\mathbf{U}$
    Update $\mathbf{Q}$ via (35)
    Update $\mathbf{X}$ by solving (14)
    Update $\boldsymbol{\upsilon}$ using (36)
end while
```

Table 1
Comparison baselines.

Index	Models	Graph-based	Fair	End-to-End	GL method
1	$k$means	✗	✗	—	—
2	Fairlets	✗	✓	✗	—
3	CorrFSC	✓	✓	✗	PC
4	KNNFSC	✓	✓	✗	k-NN
5	EpsNNFSC	✓	✓	✗	ϵ-NN
6	FGLASSO	✓	✓	✗	GLASSO
7	FJGSED	✓	✓	✓	ANGL
8	FSRSC	✓	✓	✓	SR

6. Experiments

In this section, we test our proposed model using synthetic, benchmark, and real-world data. First, some experimental setups are introduced. The code is available at https://github.com/kalman36912/UFSC.

6.1. Experimental setups

(1) Graph generation: For synthetic data, we leverage the vSBM method to generate random graphs with sensitive attributes. Specifically, we let $\zeta_s = \frac{1}{S}, a = 0.8, b = 0.2, c = 0.15$, and $d = 0.05$. After obtaining connections among nodes, we assign each edge a random weight between $[0.1, 2]$. Finally, we normalize the edge weights to satisfy $\mathrm{Tr}(\mathbf{L}^*) = D$.

(2) Signal generation: We generate N observed signals from the following Gaussian distribution [34]

$$(\mathbf{X}_o)_{[:,n]} \sim \mathcal{N}\left(\mathbf{0}, (\mathbf{L}^*)^\dagger + \mathbf{\Sigma}_e\right), \quad n = 1,\dots,N, \tag{37}$$

where $\mathbf{\Sigma}_e = \mathrm{diag}(\sigma_1^2,\dots,\sigma_D^2)$ and σ_i is the noise scale of the ith node. As stated in [34], signals generated in this way are smooth over the corresponding graph.

(3) Evaluation metrics: In topology inference, determining whether two vertices are connected can be regarded as a binary classification

Table 2
The results of our model and the compared baselines under different cases.

	$\sigma_i \sim \mathcal{U}(0, 0.2), N = 1000$				$\sigma_i \sim \mathcal{U}(0.4, 0.6), N = 1000$			
	FS ↑	EE ↓	CE ↓	Bal ↑	FS ↑	EE ↓	CE ↓	Bal ↑
$k$means	—	—	0.671	0.191	—	—	0.687	0.149
Fairlets	—	—	0.658	0.485	—	—	0.665	0.457
CorrFSC	0.472	2.858	0.567	0.482	0.441	3.016	0.578	0.705
KNNFSC	0.105	—	0.687	0.829	0.103	—	0.703	0.626
EpsNNFSC	0.086	—	0.729	0.380	0.094	—	0.739	0.333
FGLASSO	0.482	3.902	0.411	0.616	0.450	3.724	0.406	0.646
FJGSED	0.271	28.159	0.724	0.359	0.263	22.626	0.734	0.240
FSRSC	0.374	5.222	0.724	0.619	0.355	9.671	0.733	0.607
Ours	**0.501**	**2.375**	**0.286**	**0.845**	**0.474**	**2.414**	**0.390**	**0.801**
	$\sigma_i \sim \mathcal{U}(0, 0.2), N = 5000$				$\sigma_i \sim \mathcal{U}(0.4, 0.6), N = 5000$			
	FS ↑	EE ↓	CE ↓	Bal ↑	FS ↑	EE ↓	CE ↓	Bal ↑
$k$means	—	—	0.635	0.161	—	—	0.667	0.145
Fairlets	—	—	0.611	0.355	—	—	0.623	0.348
CorrFSC	0.630	2.529	0.104	0.874	0.596	2.511	0.156	0.859
KNNFSC	0.113	—	0.729	0.628	0.098	—	0.682	0.731
EpsNNFSC	0.091	—	0.718	0.652	0.065	—	0.724	0.369
FGLASSO	0.587	3.971	0.291	0.657	0.574	3.533	0.271	**0.908**
FJGSED	0.325	23.576	0.604	0.579	0.293	31.552	0.734	0.247
FSRSC	0.345	5.049	0.729	0.766	0.512	10.024	0.739	0.663
Ours	**0.715**	**1.691**	**0.052**	**0.960**	**0.674**	**2.174**	**0.142**	0.870

↑ means that higher value is better, and ↓ means that lower value is better.
$\sigma_i \sim \mathcal{U}(a1, a2), i = 1, \ldots, D$, means that the noise scale of the ith node is from the uniform distribution $\mathcal{U}(a1, a2)$.

problem. Thus, we employ the F1-score (FS) to evaluate classification results

$$\mathrm{FS} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FN} + \mathrm{FP}}, \tag{38}$$

where TP is true positive rate, TN is true negative rate, FP is false positive rate, and FN is false negative rate. We also use the estimation error (EE) of fair graph to evaluate the learned graph

$$\mathrm{EE} = \|\mathbf{Z}^\top \hat{\mathbf{L}} \mathbf{Z} - \mathbf{Z}^\top \mathbf{L}^* \mathbf{Z}\|_\mathrm{F}. \tag{39}$$

The reason we use the estimation error of fair graph instead of ϵ_L is that EE simultaneously encodes graph construction and sensitive information. For a fair comparison of EE, we normalize the learned graphs to $\mathrm{Tr}(\hat{\mathbf{L}}) = D$. As for the metrics used to evaluate fair clustering, we use the same two metrics as in [15]: clustering error (CE) and Balance (Bal)

$$\mathrm{CE} = \frac{1}{D}\left|\{i : \hat{\pi}_C(i) \neq \pi_C(i), i = 1...., D\}\right|,$$
$$\text{Balance (Bal)} = \frac{1}{K}\sum_{k=1}^{K} \mathrm{Balance}(C_k), \tag{40}$$

where $\hat{\pi}_C(i)$ is the estimated cluster label of node i (after proper permutation), and $\pi_C(i)$ is the ground-truth. The metric Balance measures the average balance of all clusters.

(4) Baselines: The comparison baselines are list in Table 1. The model Fairlets is the fair version of $k$median [9]. Models 3–5 are the implementations of [15] using different graph construction methods. FGLASSO [60] is the only model that jointly performs graph construction and fair spectral embedding. FSRSC and FJGSED are the fair versions of unified SC models (25) and (26). Specifically, we add the fairness constraints $\mathbf{F}^\top \mathbf{U} = \mathbf{0}$ to the models to (25)–(26) and solve the corresponding problem to perform fair clustering.

(5) Determination of parameters: For our model, we first grid-search ξ and β corresponding to the best FS from the set $\{0.001, 0.005, 0.01..., 0.1\}$ for the graph learning task. Then, parameters μ and γ are selected as those achieving the best CE from the set $\{0.001, 0.005, 0.01...1\}$. All parameters of baselines are also selected as those achieving the best CE values.

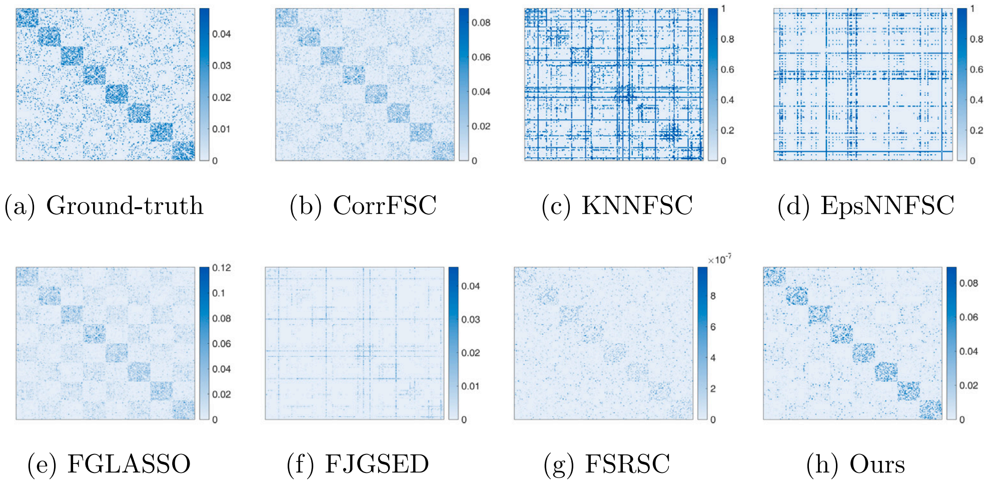
6.2. Synthetic data

(1) Model performance: We first compare our model with all baselines in four cases. We let $D = 192, K = 4$, and $S = 2$. As listed in Table 2, our model outperforms $k$means and Fairlets on clustering metrics because it exploits structured information behind raw data. The graphs established by KNNFSC and EpsNNFSC methods are not evaluated by EE since no edge weights are assigned. Among Models 3–5, CorrFSC achieves the best GL performance (FS) as well as the best CE clustering performance (CE). However, the graph construction performance of the three methods is inferior to our model, leading to unsatisfactory fair clustering results. Furthermore, compared with the three methods, our model unifies all separate stages into a single optimization objective, avoiding suboptimality caused by separate optimization. The reason why our model outperforms FGLASSO could be that FGLASSO separately uses $k$means to obtain final cluster labels. Besides, our method could learn better graphs than FGLASSO. Although FJGSED and FSRSC also perform fair clustering in an end-to-end manner, our model obtains superior fair clustering performance due to more accurate graphs constructed by our method. Finally, our model has a node-adaptive graph filter to denoise observed signals. Thus, our model obtains the best graph construction performance under different levels of noise contamination.

We visualize the learned graphs in Fig. 4. We see that EpsNNFSC fails to capture the clustering structure, resulting in the worst fair clustering performance. The graph of KNNFSC tends to have imbalanced node degrees, and the graph of FSRSC has small edge weights. Compared with CorrFSC, FGLASSO, and FJGSED, the graph of our model has fewer noisy edges and clearer clusters.

(2) The effect of K and S: We set $d = 192, N = 5000, \sigma_i \sim \mathcal{U}(0.4, 0.6)$. In the first case, we fix $S = 2$ and vary K from 2 to 6. In the second case, we fix $K = 2$ and vary S from 2 to 6. Fig. 5 displays that the fair clustering performance degrades with the increase of K (CE increases and Balance decreases), which is consistent with Proposition 1. On the other hand, the fair clustering performance is less affected by S. As S changes, CE remains around 0.05 and Balance remains around 0.9, indicating the stable and high performance of our method in terms of different S.

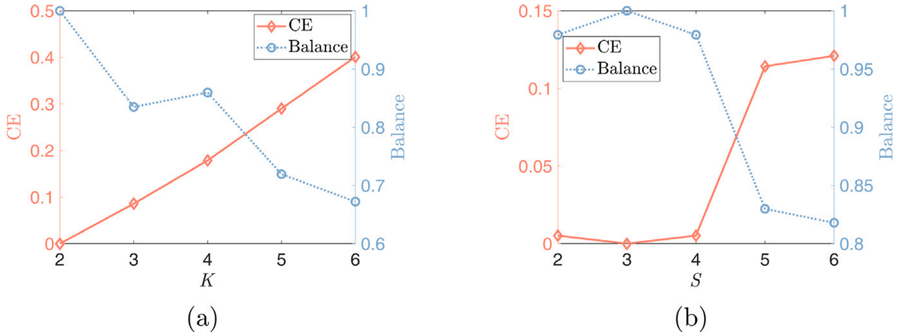
(3) The effect of D: We let $N = 10^4, \sigma_i \sim \mathcal{U}(0.4, 0.6), K = 4$ and $S = 2$. As depicted in Fig. 6, for a fixed data size, CE first decreases to close to 0 and then increases as D. Besides, Balance first increases to close to 1 and then decreases. The reason may be that, as stated in Proposition 1, the misclassification rate of FSC algorithms on the graph generated by the vSBM method decreases as D if the underlying graph is exactly estimated. However, the quality of the estimated graph declines for



(a) Ground-truth (b) CorrFSC (c) KNNFSC (d) EpsNNFSC

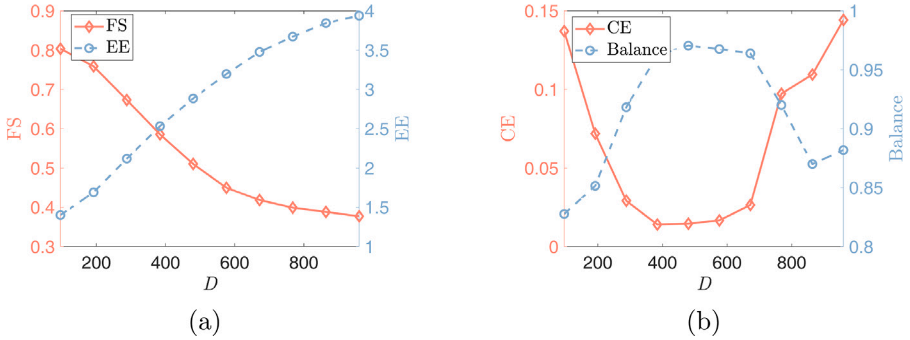
(e) FGLASSO (f) FJGSED (g) FSRSC (h) Ours

Fig. 4. The visualization of the learned graphs (unnormalized weights) when $N = 5000$ and $\sigma_i \sim \mathcal{U}(0.4, 0.6)$.



(a) (b)

Fig. 5. The effect of (a) K and (b) S on fair clustering performance.



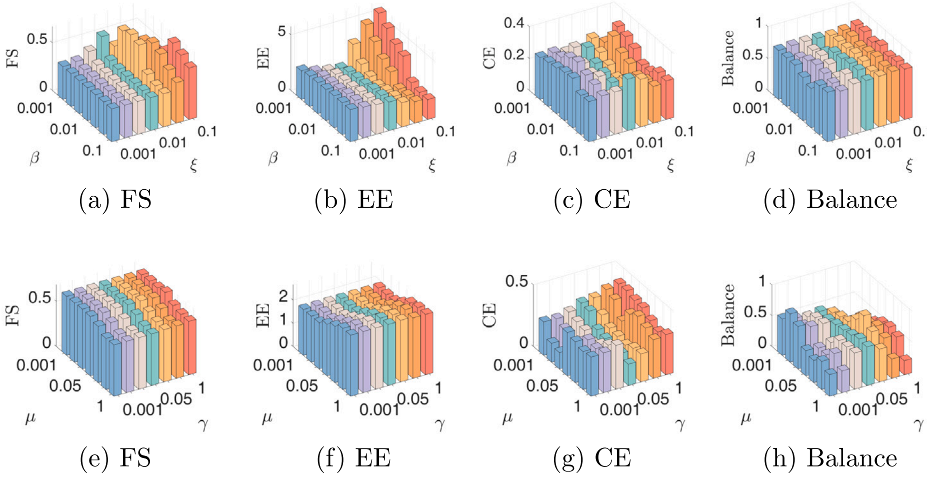
(a) (b)

Fig. 6. The effect of D on (a) graph learning and (b) fair clustering.

large D if N is fixed. Thus, the second part of the error bound in Proposition 1 worsens. If the performance improvement brought by increasing D is smaller than the degradation caused by the graph estimation error, fair clustering performance decreases when D is large.

(4) The sensitivity of parameters: We let $D = 196$, $K = 4$, $S = 2$, $N = 5000$, and , $\sigma_i \sim \mathcal{U}(0.4, 0.6)$. First, we fix $\mu = 0.01$ and $\gamma = 0.01$ and vary β and ξ from 0.001 to 0.1. We then fix $\beta = 0.01$ and $\xi = 0.05$ and vary μ and γ from 0.001 to 1. As shown in Fig. 7, our model can achieve consistent GL and fair clustering performance except when β is too small and ξ is too large. Moreover, GL performance is more sensitive to μ than γ. There exist combinations of μ and γ that achieve satisfactory CE and Balance simultaneously.

(5) The effect of the fairness constraint: We consider a special case where the real graph contains two clusters, each consisting of samples from the same sensitive group. In this case, the Balance of real clustering is zero. This is an extreme case we used to test the effect of the fairness constraint, where there is only one ground-truth clustering. We then perform clustering using our FSC model and a variant where the fairness constraint is removed. As shown in Fig. 8, if we remove the fairness constraint, our model can exactly group all samples. However, some samples are misclassified to improve fairness in our model due to the effect of the fairness constraint. We list the corresponding model performance in Table 3. Our model achieves a significant increase in Balance value (from 0 to 0.867) at the cost of increased CE (from 0 to 0.484). The GL performance of our model is also degraded due to the fairness constraint. Thus, if the underlying graph has only one meaningful cluster that is highly unbalanced, fairness constraints may degrade GL and clustering performance. In the real world, there may be two or more meaningful ground-truth clustering [15], and the fairness constraint can help recover the fair one.



(a) FS (b) EE (c) CE (d) Balance

(e) FS (f) EE (g) CE (h) Balance

Fig. 7. The effect of parameter sensitivity. (a)–(d) The results of varying ξ and β. (e)–(h) The results of varying μ and γ.

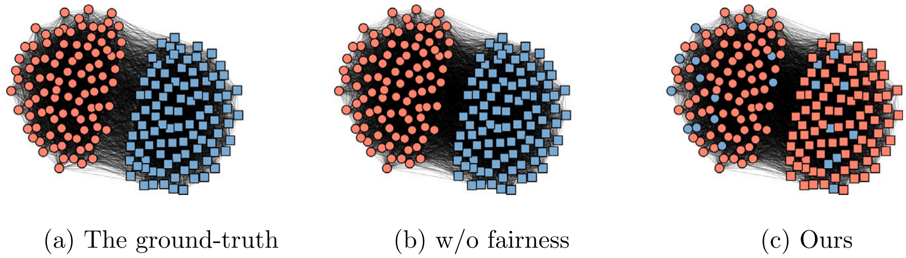

(a) The ground-truth (b) w/o fairness (c) Ours

Fig. 8. The effect of the fairness constraint. Colors represent clusters, while mark shapes represent sensitive groups.

Table 3
The results of removing fairness.

	FS	EE	CE	Bal
w/o fairness	0.734	1.211	0	0
Ours	0.719	1.353	0.484	0.867

Table 4
The results of ablation studies.

	FS	EE	CE	Bal
Ours-Sep	0.637	2.333	0.250	0.704
Ours-$k$means	0.635	2.320	0.549	0.353
Ours-noDN	0.623	2.354	0.276	0.694
Ours	**0.658**	**2.203**	**0.167**	**0.782**

(6) Ablation study: Three cases are taken into consideration. (i) We construct graphs using (13), conduct fair spectral embedding, and discretize using spectral rotation separately to test the benefit of a unified model (Ours-Sep). (ii) We construct graphs using (14) and conduct fair spectral embedding jointly. After obtaining continuous results, we exploit $k$means as the discretization step to test the benefit of spectral rotation (Ours-$k$means). (iii) We remove the denoising module in our model to test the benefit of the node-adaptive graph filter (Ours-noDN). We let $D = 196, K = 4, S = 2, N = 5000$, and $\sigma_i \sim \mathcal{U}(0.4, 0.6)$, and the results are listed in Table 4. Our model outperforms Ours-Sep in terms of all four metrics, demonstrating the benefit of a unified model. Although the graph of Ours-$k$means is well estimated, our method significantly improves the clustering performance CE by about 70%, since our method uses a unified framework and spectral rotation. Finally, our model outperforms Ours-noDN, especially in terms of graph construction metrics, where FS and EE improved by 5.6% and 6.4% respectively. The reason is that our method has a low-pass filter to enhance graph construction.

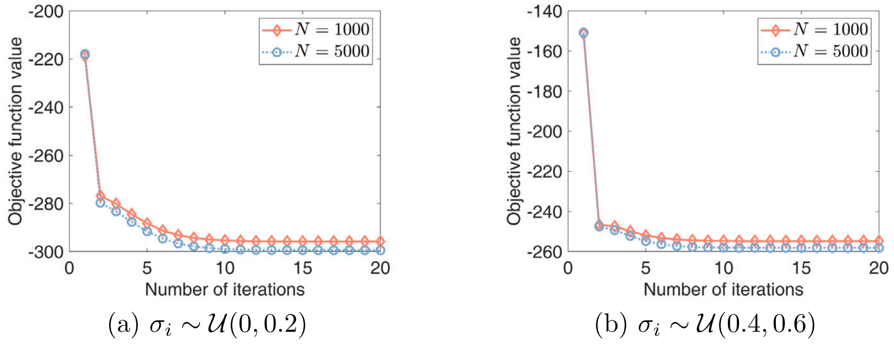
(7) Convergence: We let $D = 196, K = 4, S = 2, D = 192$. As shown in Fig. 9, the objective function values monotonically decrease as the number of iterations. Besides, our algorithm converges within a few iterations, indicating its fast convergence.

6.3. Benchmark data

In this section, we test the performance of our model on the commonly used benchmark datasets of FSC [15]. The first dataset is a high school friendship network named FACEBOOKNET.[1] The dataset contains a graph with vertices representing high school students and edges representing connections between students on Facebook. After data preprocessing, we obtain 155 students split into male and female groups. In this dataset, gender is considered a sensitive attribute. All vertices are divided into two groups, i.e., male and female. The second dataset, DRUGNET, is a network encoding acquaintanceship between drug users in Hartford.[2] After data preprocessing, we obtain 193 vertices. We use ethnicity as a sensitive attribute and split the vertices into three groups: African Americans, Latinos, and others. Note that previous FSC work [15] is based on a given graph, and the two datasets only contain ground-truth graphs and no observed signals. However, one of the primary advantages of our model is that we can group observed data without the real graph structures. Thus, we generate data via (37) based on the ground-truth networks. We then use our model to group vertices via the observed data. For comparison, we apply the FSC algorithm in [15] (FairSC) and unnormalized spectral clustering (SC) to the real networks to cluster vertices. We aim to demonstrate that our model can achieve competitive fair clustering performance even

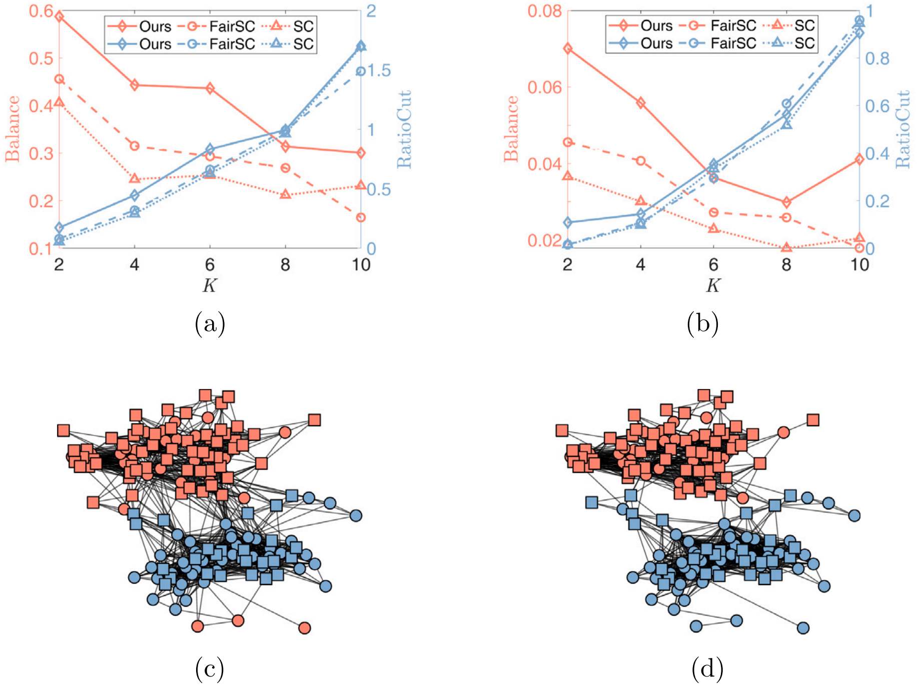
[1] http://www:sociopatterns:org/datasets/high-school-contact-and-friendshipnetworks/

[2] https://sites:google:com/site/ucinetsoftware/datasets/covert-networks/drugnet



(a) $\sigma_i \sim \mathcal{U}(0, 0.2)$ (b) $\sigma_i \sim \mathcal{U}(0.4, 0.6)$

Fig. 9. The convergence of our algorithm.



(a) (b)

(c) (d)

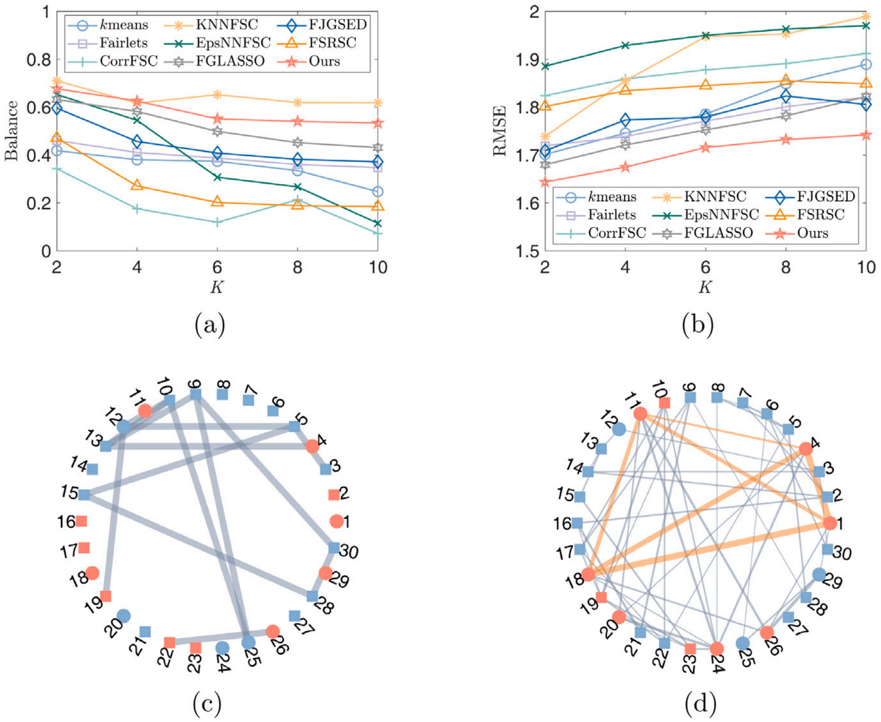
Fig. 10. The results of the benchmark datasets. (a)–(b) The fair clustering results of the FACEBOOK and DRUGNET datasets. (c)–(d) The real and the learned FACEBOOK network. Colors represent clusters, while mark shapes represent sensitive groups.

without real graphs. Referring to [15], we use Balance and RatioCut as evaluation metrics since we have no real labels. We let $N = 1000$ and $\sigma_i \sim \mathcal{U}(0, 0.2)$. As displayed in 10(a)–(b), for the two datasets, our model achieves almost the same RatioCut as FairSC and SC—which are based on the ground-truth networks—even though we do not know the underlying graphs. However, compared to the state-of-the-art model FairSC, our model can improve Balance by 60% and 41% on average over K on two benchmark datasets, meaning that our method can improve fairness in clustering at a moderate cost of RationCut. Figs. 10(c)–(d) depict the real FACEBOOKNET graph and the graph learned by our model when $K = 2$. Fewer edges are learned between two clusters, suggesting that our model may tend to learn a graph that is more suitable for clustering. Furthermore, due to the fairness constraints, our graph exhibits some changes compared to the real graph to obtain higher balanced clustering results. Two clusters are observed from our learned graph, meaning our model can partition the nodes from the observed data even if we have no real graphs.

6.4. Real data

(1) MovieLens 100K dataset: We employ MovieLens 100K dataset[3] to group movies by their ratings. This dataset contains ratings of 1682 movies by 943 users in the range $[1, 5]$, which is sparse as many movies have few ratings. To alleviate the impact of sparsity, we select the top 200 most-rated movies from 1682 movies. Therefore, we have a who-rated-what matrix $\mathbf{X} \in \mathbb{R}^{200\times943}$. The matrix can be used to construct a movie–movie similarity graph strongly correlated with how users explicitly rate items [61]. Therefore, we can perform clustering on the similarity graph to group movies with similar attributes. However, as stated in [60], old movies tend to obtain higher ratings because only masterpieces have survived. To obtain fair results unbiased by production time, we consider movie year as a sensitive attribute. Movies

[3] http://www.grouplens.org



(a)

(b)

(c)

(d)

Fig. 11. The results of the MovieLens dataset. (a)–(b) The fair clustering results of different models. (c)–(d) The learned sub-graphs of KNNFSC and our model when $K = 2$. Colors represent clusters, while mark shapes represent sensitive attributes.

made before 1991 are considered old, while others are considered new. To evaluate clustering results, we conduct traditional item-based collaborative filtering (CF) on each cluster, termed cluster CF, to predict user ratings of movies. As claimed in [60], if the obtained clusters accurately contain a set of similar items, cluster CF can better predict user ratings of movies. Therefore, we follow [60] and use root mean square error (RMSE) between the predicted and true ratings as an evaluation metric in addition to Balance [60,61]. Figs. 11(a)–(b) depict fair clustering results of different models. Our model obtains the highest Balance except KNNFSC. However, KNNFSC performs poorly on RMSE, indicating unsatisfactory clustering results. This may be caused by the fact that the graph constructed by KNNFSC hardly characterizes the similarity relationships between movies. In contrast, our model achieves the best RMSE since it better reveals similarity relationships behind observed data. In Fig. 11(c)–(d), we provide the learned sub-graphs and clustering results of the top 30 rated movies when $K = 2$. The graph learned by KNNFSC has isolated nodes since they are connected to the movies outside the top 30 rated movies. In our graph, nodes 1, 4, 11, and 18 are closely connected because they belong to the Star Wars series. However, in Fig. 11(c), they are not connected. Moreover, our model successfully groups the four nodes into the same cluster.

(2) MNIST-USPS dataset: The second dataset we employ is MNIST-USPS dataset, which contains two sub-datasets, i.e., MNIST[4] and USPS.[5] The two sub-datasets contain images of handwritten digits from 0 to 9. We cluster these images and use digits as the ground-truth cluster labels. Specifically, we randomly select 48 images from each sub-dataset, which contains four digits and twelve pictures for each digit. We finally obtain 96 images and resize each image to a 28 × 28 matrix. We take each image as a node in a graph and flatten the corresponding matrix as the node signals. Therefore, the observed data are $\mathbf{X} \in \mathbb{R}^{96\times784}$. We take the domain source of images—images from MNIST or USPS—as a sensitive attribute. Thus, we have $S = 2$ and $K = 4$. We use CE and Balance as evaluation metrics since we have real labels but no ground-truth graphs. As shown in Fig. 12, our model achieves the best fair clustering performance for both CE and Balance. Compared with the second-best model, our model improves CE by 21% and Balance by 12%, indicating its superiority. The reason for CorrFSC, KNNFSC, EpsFSC, and FSRSC achieving unsatisfactory CE may be that the corresponding graphs cannot reflect the real topological similarity.

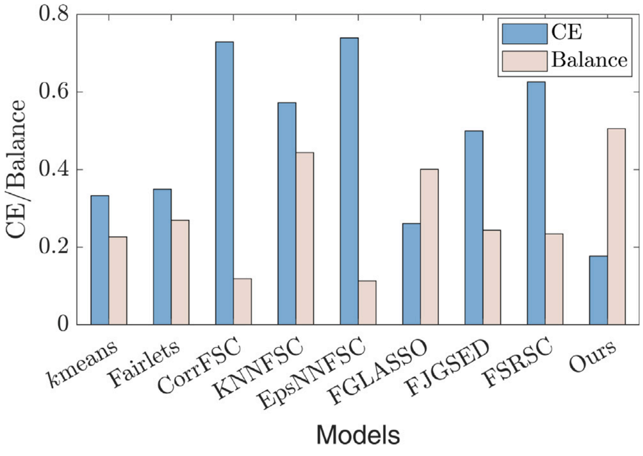


Fig. 12. The clustering results of the MNIST-USPS dataset.

7. Conclusion

In this paper, we theoretically analyzed the impact of similarity graphs on FSC performance. Motivated by the analysis, we proposed a graph construction method for FSC tasks as well as an end-to-end FSC framework. Then, we designed an efficient algorithm to alternately update the variables corresponding to each submodule in our model.

[4] http://yann.lecun.com/exdb/mnist

[5] https://www.kaggle.com/bistaumanga/usps-dataset

Extensive experiments showed that our approach is superior to state-of-the-art (fair) SC models. Future research directions may include developing more scalable FSC algorithms.

CRediT authorship contribution statement

Xiang Zhang: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Qiao Wang:** Writing – review & editing, Validation, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

This work is supported in part by Jiangsu provincial key research and development program: Technology demonstration of urban intelligent management platform based on big data (BE2023799).

Appendix. Proof of Proposition 1

We first provide the following lemma.

Lemma 2. *For any $\epsilon_S > 0$ and any two matrices $\mathbf{U}, \widehat{\mathbf{U}} \in \mathbb{R}^{D\times K}$ such that $\mathbf{U} = \mathbf{QR}$ with $\mathbf{Q} \in \mathcal{I}$ and $\mathbf{R}^\top\mathbf{R} = \mathbf{I}$, let $(\widehat{\mathbf{Q}}, \widehat{\mathbf{R}})$ be a $(1+\epsilon_S)$ approximation of $\widehat{\mathbf{U}}$ using spectral rotation as Assumption 1, and $\breve{\mathbf{U}} = \widehat{\mathbf{Q}}\widehat{\mathbf{R}}$. Then, for any $\delta_k \geq 0$, define $\widetilde{\mathcal{M}}_k = \left\{ i \in \mathcal{C}_k : \|\mathbf{U}_{[i,:]} - \breve{\mathbf{U}}_{[i,:]}\|_2 \geq \delta_k/2 \right\}$, $k = 1, \ldots, K$, and we have*

$$\sum_{k=1}^{K} |\widetilde{\mathcal{M}}_k| \delta_k^2 \leq 4(4+2\epsilon_S)\|\mathbf{U} - \widehat{\mathbf{U}}\|_{\mathrm{F}}^2, \tag{A.1}$$

Proof. First, by the procedure of spectral rotation, we have

$$\begin{aligned} &\widehat{\mathbf{Q}}, \widehat{\mathbf{R}} = \min_{\mathbf{Q},\mathbf{R}} \|\mathbf{Q} - \widehat{\mathbf{U}}\mathbf{R}\|_{\mathrm{F}}^2 \quad \text{s.t.} \ \ \mathbf{R}^\top\mathbf{R} = \mathbf{I},\ \mathbf{Q} \in \mathcal{I} \\ \Leftrightarrow &\widehat{\mathbf{Q}}, \widehat{\mathbf{R}} = \min_{\mathbf{Q},\mathbf{R}} \|\widehat{\mathbf{U}} - \mathbf{QR}\|_{\mathrm{F}}^2 \quad \text{s.t.} \ \ \mathbf{R}^\top\mathbf{R} = \mathbf{I},\ \mathbf{Q} \in \mathcal{I}. \end{aligned} \tag{A.2}$$

Then, based on Assumption 1, we can obtain that

$$\begin{aligned} &\|\widehat{\mathbf{U}} - \widehat{\mathbf{Q}}\widehat{\mathbf{R}}\|_{\mathrm{F}}^2 \leq (1+\epsilon_S) \min_{\mathbf{Q}\in\mathcal{I}, \mathbf{R}^\top\mathbf{R}=\mathbf{I}} \|\widehat{\mathbf{U}} - \mathbf{QR}\|_{\mathrm{F}}^2 \\ \Rightarrow &\|\widehat{\mathbf{U}} - \breve{\mathbf{U}}\|_{\mathrm{F}}^2 \leq (1+\epsilon_S)\|\widehat{\mathbf{U}} - \mathbf{U}\|_{\mathrm{F}}^2. \end{aligned} \tag{A.3}$$

It is not difficult to obtain the following inequalities

$$\begin{aligned} &\|\breve{\mathbf{U}} - \mathbf{U}\|_{\mathrm{F}}^2 \leq 2\|\breve{\mathbf{U}} - \widehat{\mathbf{U}}\|_{\mathrm{F}}^2 + 2\|\widehat{\mathbf{U}} - \mathbf{U}\|_{\mathrm{F}}^2 \\ &\leq (4+2\epsilon_S)\|\widehat{\mathbf{U}} - \mathbf{U}\|_{\mathrm{F}}^2. \end{aligned} \tag{A.4}$$

The first inequality holds due to the basic inequality, and the second one holds due to (A.3). Finally, according to the definition of δ_k, we can obtain the conclusion (A.1).

We start our proof of Proposition 1, which is inspired by [15]. To incorporate the fairness constraint into the objective function of (7), we let $\widehat{\mathbf{U}} = \mathbf{Z}\widehat{\mathbf{Y}}$, where $\widehat{\mathbf{Y}}$ contains the eigenvectors of $\mathbf{Z}^\top\widehat{\mathbf{L}}\mathbf{Z}$ corresponding to the K smallest eigenvalues. Suppose that $\bar{\mathbf{Y}}$ contains the eigenvectors of $\mathbf{Z}^\top\bar{\mathbf{L}}\mathbf{Z}$ corresponding to the K smallest eigenvalues, where $\bar{\mathbf{L}}$ is the expected Laplacian matrix of the graphs generated by the vSBM method. We apply spectral rotation on $\widehat{\mathbf{U}}$ estimated from $\widehat{\mathbf{L}}$ by solving (7). For any $\mathbf{V} \in \mathbb{R}^{K\times K}$ satisfying $\mathbf{V}^\top\mathbf{V} = \mathbf{I}, \mathbf{V}\mathbf{V}^\top = \mathbf{I}$, it is not difficult to obtain

$$\begin{aligned} &\|\mathbf{Z}\bar{\mathbf{Y}} - \mathbf{Z}\widehat{\mathbf{Y}}\mathbf{V}\|_{\mathrm{F}}^2 = \mathrm{Tr}\left((\bar{\mathbf{Y}} - \widehat{\mathbf{Y}}\mathbf{V})^\top\mathbf{Z}^\top\mathbf{Z}(\bar{\mathbf{Y}} - \widehat{\mathbf{Y}}\mathbf{V}) \right) \\ =&\|\bar{\mathbf{Y}} - \widehat{\mathbf{Y}}\mathbf{V}\|_{\mathrm{F}}^2. \end{aligned} \tag{A.5}$$

Therefore, we have

$$\begin{aligned} &\min_{\mathbf{V}^\top\mathbf{V}=\mathbf{I},\mathbf{V}\mathbf{V}^\top=\mathbf{I}} \|\mathbf{Z}\bar{\mathbf{Y}} - \mathbf{Z}\widehat{\mathbf{Y}}\mathbf{V}\|_{\mathrm{F}} = \min_{\mathbf{V}^\top\mathbf{V}=\mathbf{I},\mathbf{V}\mathbf{V}^\top=\mathbf{I}} \|\bar{\mathbf{Y}} - \widehat{\mathbf{Y}}\mathbf{V}\|_{\mathrm{F}} \\ \leq&\frac{8\sqrt{2K^3}}{D(c-d)}\|\mathbf{Z}^\top\bar{\mathbf{L}}\mathbf{Z} - \mathbf{Z}^\top\widehat{\mathbf{L}}\mathbf{Z}\|_2 \leq \frac{8\sqrt{2K^3}}{D(c-d)}\|\mathbf{Z}^\top\bar{\mathbf{L}}\mathbf{Z} - \mathbf{Z}^\top\widehat{\mathbf{L}}\mathbf{Z}\|_{\mathrm{F}}. \end{aligned} \tag{A.6}$$

The first inequality holds due to [15] and how we generate the ground-truth graph, and the second inequality holds due to norm inequality. On the other hand, we have

$$\|\mathbf{Z}\bar{\mathbf{Y}} - \mathbf{Z}\widehat{\mathbf{Y}}\mathbf{V}\|_{\mathrm{F}} = \|\mathbf{Z}\bar{\mathbf{Y}}\mathbf{V}^\top - \mathbf{Z}\widehat{\mathbf{Y}}\|_{\mathrm{F}}. \tag{A.7}$$

As in Lemma 6 of [15], we can choose $\bar{\mathbf{Y}}$ in such a way that $\mathbf{Z}\bar{\mathbf{Y}} = \mathbf{E}$, where $\mathbf{E}_{[i,:]} = \mathbf{E}_{[j,:]}$ if the vertices i and j are in the same cluster and $\|\mathbf{E}_{[i,:]} - \mathbf{E}_{[j,:]}\|_2 = \sqrt{2K/D}$ if the vertices i and j are not in the same cluster. Furthermore, multiplying $\mathbf{E}$ by $\mathbf{V}^\top$ will not change the properties of $\mathbf{E}$ since $\mathbf{V}^\top$ is a orthogonal matrix. Finally, according to Lemma 2, if we let $\delta_k = \sqrt{2K/D}$, then $\widetilde{\mathcal{M}}_k$ in Lemma 2 is equivalent to $\mathcal{M}_k$. Furthermore, according to Lemma 5.3 in [46], if $\frac{4(4+2\epsilon_S)}{\delta_k^2}\|\mathbf{E}\mathbf{V}^\top - \mathbf{Z}\widehat{\mathbf{Y}}\|_{\mathrm{F}}^2 \leq \frac{D}{K}$, we have

$$\begin{aligned} \sum_{k=1}^{K} |\mathcal{M}_k| &\leq \frac{4(4+2\epsilon_S)}{\delta_k^2}\|\mathbf{E}\mathbf{V}^\top - \mathbf{Z}\widehat{\mathbf{Y}}\|_{\mathrm{F}}^2 \\ &\leq \frac{256(4+2\epsilon_S)K^2}{D(c-d)^2}\|\mathbf{Z}^\top\bar{\mathbf{L}}\mathbf{Z} - \mathbf{Z}^\top\widehat{\mathbf{L}}\mathbf{Z}\|_{\mathrm{F}}^2. \end{aligned} \tag{A.8}$$

Let $C_1 = \frac{256(4+2\epsilon_S)K^2}{D(c-d)^2}$, we have

$$\begin{aligned} \sum_{k=1}^{K} |\mathcal{M}_k| \leq 2C_1 &\underbrace{\|\mathbf{Z}^\top\bar{\mathbf{L}}\mathbf{Z} - \mathbf{Z}^\top\mathbf{L}^*\mathbf{Z}\|_{\mathrm{F}}^2}_{\mathcal{T}_1} \\ + 2C_1 &\underbrace{\|\mathbf{Z}^\top\mathbf{L}^*\mathbf{Z} - \mathbf{Z}^\top\widehat{\mathbf{L}}\mathbf{Z}\|_{\mathrm{F}}^2}_{\mathcal{T}_2}. \end{aligned} \tag{A.9}$$

The first term is the difference between the expected Laplacian matrix and the real matrix, which has been derived in [15]. Specifically, for any $r_2 > 0$ and some $r_1 > 0$ satisfying $a \geq r_1 \ln D/D$, with probability at least $1 - D^{-r_2}$, we have that there exist a constant $C_2(r_1, r_2)$ such that

$$\mathcal{T}_1 \leq C_2(r_1, r_2)aD\ln D. \tag{A.10}$$

We then focus on $\|\mathbf{Z}^\top\mathbf{L}^*\mathbf{Z} - \mathbf{Z}^\top\widehat{\mathbf{L}}\mathbf{Z}\|_{\mathrm{F}}^2$ and have

$$\begin{aligned} &\|\mathbf{Z}^\top\mathbf{L}^*\mathbf{Z} - \mathbf{Z}^\top\widehat{\mathbf{L}}\mathbf{Z}\|_{\mathrm{F}} \leq \sqrt{D}\|\mathbf{Z}^\top\mathbf{L}^*\mathbf{Z} - \mathbf{Z}^\top\widehat{\mathbf{L}}\mathbf{Z}\|_2 \\ \leq&\sqrt{D}\|\mathbf{Z}^\top\|_2\|\mathbf{L}^* - \widehat{\mathbf{L}}\|_2\|\mathbf{Z}\|_2 = \sqrt{D}\|\mathbf{L}^* - \widehat{\mathbf{L}}\|_2 \leq \sqrt{D}\|\mathbf{L}^* - \widehat{\mathbf{L}}\|_{\mathrm{F}}. \end{aligned} \tag{A.11}$$

The last inequality holds due to $\|\mathbf{Z}^\top\|_2 = \|\mathbf{Z}\|_2 = 1$. Based on (A.11), we have

$$\mathcal{T}_2 \leq D\|\mathbf{L}^* - \widehat{\mathbf{L}}\|_{\mathrm{F}}^2 = D\epsilon_L^2 \tag{A.12}$$

Finally, we complete the proof.

References

[1] T. Lei, X. Jia, Y. Zhang, S. Liu, H. Meng, A.K. Nandi, Superpixel-based fast fuzzy C-means clustering for color image segmentation, IEEE Trans. Fuzzy Syst. 27 (9) (2018) 1753–1766.

[2] H. Xie, A. Zhao, S. Huang, J. Han, S. Liu, X. Xu, X. Luo, H. Pan, Q. Du, X. Tong, Unsupervised hyperspectral remote sensing image clustering based on adaptive density, IEEE Geosci. Remote S. 15 (4) (2018) 632–636.

[3] V.Y. Kiselev, T.S. Andrews, M. Hemberg, Challenges in unsupervised clustering of single-cell RNA-seq data, Nat. Rev. Genet. 20 (5) (2019) 273–282.

[4] A. Likas, N. Vlassis, J.J. Verbeek, The global k-means clustering algorithm, Pattern Recognit. 36 (2) (2003) 451–461.

[5] U. Von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (2007) 395–416.

[6] W.-B. Xie, Y.-L. Lee, C. Wang, D.-B. Chen, T. Zhou, Hierarchical clustering supported by reciprocal nearest neighbors, Inform. Sci. 527 (2020) 279–292.

[7] Z. Hu, F. Nie, W. Chang, S. Hao, R. Wang, X. Li, Multi-view spectral clustering via sparse graph learning, Neurocomputing 384 (2020) 1–10.

[8] A. Chouldechova, A. Roth, The frontiers of fairness in machine learning, 2018, arXiv:1810.08810.

[9] F. Chierichetti, R. Kumar, S. Lattanzi, S. Vassilvitskii, Fair clustering through fairlets, Proc. Adv. Neural Inf. Process. Syst. 30 (2017).

[10] S. Bera, D. Chakrabarty, N. Flores, M. Negahbani, Fair algorithms for clustering, Proc. Adv. Neural Inf. Process. Syst. 32 (2019).

[11] A. Backurs, P. Indyk, K. Onak, B. Schieber, A. Vakilian, T. Wagner, Scalable fair clustering, in: Proc. Int. Conf. Mach. Learn., PMLR, 2019, pp. 405–413.

[12] I.M. Ziko, J. Yuan, E. Granger, I.B. Ayed, Variational fair clustering, in: Proc. Natl. Conf. Artif. Intell., vol. 35, 2021, pp. 11202–11209.

[13] P. Zeng, Y. Li, P. Hu, D. Peng, J. Lv, X. Peng, Deep fair clustering via maximizing and minimizing mutual information: Theory, algorithm and metric, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2023, pp. 23986–23995.

[14] P. Li, H. Zhao, H. Liu, Deep fair clustering for visual learning, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 9070–9079.

[15] M. Kleindessner, S. Samadi, P. Awasthi, J. Morgenstern, Guarantees for spectral clustering with fairness constraints, in: Proc. Int. Conf. Mach. Learn., PMLR, 2019, pp. 3458–3467.

[16] J. Wang, D. Lu, I. Davidson, Z. Bai, Scalable spectral clustering with group fairness constraints, in: Proc. Int. Conf. Artif. Intell. Stat., AISTATS, PMLR, 2023, pp. 6613–6629.

[17] J. Li, Y. Wang, A. Merchant, Spectral normalized-cut graph partitioning with fairness constraints, 2023, arXiv:2307.12065.

[18] S. Gupta, A. Dukkipati, Protecting individual interests across clusters: Spectral clustering with guarantees, 2021, arXiv:2105.03714.

[19] Y. Wang, J. Kang, Y. Xia, J. Luo, H. Tong, iFiG: Individually fair multi-view graph clustering, in: 2022 IEEE International Conference on Big Data (Big Data), IEEE, 2022, pp. 329–338.

[20] J. Huang, F. Nie, H. Huang, Spectral rotation versus k-means in spectral clustering, in: Proc. Natl. Conf. Artif. Intell., vol. 27, 2013, pp. 431–437.

[21] Z. Kang, C. Peng, Q. Cheng, Z. Xu, Unified spectral clustering with optimal graph, in: Proc. Natl. Conf. Artif. Intell., vol. 32, 2018, pp. 3366–3373.

[22] Z. Kang, C. Peng, Q. Cheng, Twin learning for similarity and clustering: A unified kernel approach, in: Proc. Natl. Conf. Artif. Intell., vol. 31, 2017, pp. 2080–2086.

[23] J. Huang, F. Nie, H. Huang, A new simplex sparse learning model to measure data similarity for clustering, in: Int. Joint Conf. Artif. Intell., 2015, pp. 3569–3575.

[24] Y. Peng, W. Huang, W. Kong, F. Nie, B.-L. Lu, JGSED: An end-to-end spectral clustering model for joint graph construction, spectral embedding and discretization, IEEE Trans. Emerg. Topics Comput. Intell. (2023).

[25] E. Elhamifar, R. Vidal, Sparse subspace clustering: Algorithm, theory, and applications, IEEE Trans. Pattern Anal. Mach. Intell 35 (11) (2013) 2765–2781.

[26] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, IEEE Trans. Pattern Anal. Mach. Intell 35 (1) (2012) 171–184.

[27] M. Chen, M. Gong, X. Li, Robust doubly stochastic graph clustering, Neurocomputing 475 (2022) 15–25.

[28] F. Nie, X. Wang, H. Huang, Clustering and projected clustering with adaptive neighbors, in: Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2014, pp. 977–986.

[29] S.-J. Xiang, H.-C. Li, J.-H. Yang, X.-R. Feng, Dual auto-weighted multi-view clustering via autoencoder-like nonnegative matrix factorization, Inform. Sci. 667 (2024) 120458.

[30] Z. Kang, H. Pan, S.C. Hoi, Z. Xu, Robust graph learning from noisy data, IEEE Trans. Cybern. 50 (5) (2019) 1833–1843.

[31] X. Li, M. Chen, Q. Wang, Adaptive consistency propagation method for graph clustering, IEEE Trans. Neural Netw. Learn. Syst. 32 (4) (2019) 797–802.

[32] C. Gao, Y. Wang, J. Zhou, W. Ding, L. Shen, Z. Lai, Possibilistic neighborhood graph: A new concept of similarity graph learning, IEEE Trans. Emerg. Topics Comput. Intell. (2022).

[33] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, P. Vandergheynst, The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains, IEEE Signal Process. Mag. 30 (3) (2013) 83–98.

[34] X. Dong, D. Thanou, P. Frossard, P. Vandergheynst, Learning Laplacian matrix in smooth graph signal representations, IEEE Trans. Signal Process. 64 (23) (2016) 6160–6173.

[35] V. Kalofolias, How to learn a graph from smooth signals, in: Proc. Int. Conf. Artif. Intell. Stat., AISTATS, PMLR, 2016, pp. 920–929.

[36] X. Dong, D. Thanou, M. Rabbat, P. Frossard, Learning graphs from data: A signal representation perspective, IEEE Signal Process. Mag. 36 (3) (2019) 44–63.

[37] X. Zhang, Q. Wang, A graph-assisted framework for multiple graph learning, IEEE Trans. Signal. Inf. Process. Netw. (2024).

[38] F. Nie, D. Wu, R. Wang, X. Li, Self-weighted clustering with adaptive neighbors, IEEE Trans. Neural Netw. Learn. Syst. 31 (9) (2020) 3428–3441.

[39] Y. Pang, J. Xie, F. Nie, X. Li, Spectral clustering by joint spectral embedding and spectral rotation, IEEE Trans. Cybern. 50 (1) (2018) 247–258.

[40] Y. Yang, F. Shen, Z. Huang, H.T. Shen, A unified framework for discrete spectral clustering, in: IJCAI, 2016, pp. 2273–2279.

[41] W. Huang, Y. Peng, Y. Ge, W. Kong, A new kmeans clustering model and its generalization achieved by joint spectral embedding and rotation, PeerJ Comput. Sci. 7 (2021) e450.

[42] Y. Han, L. Zhu, Z. Cheng, J. Li, X. Liu, Discrete optimal graph clustering, IEEE Trans. Cybern. 50 (4) (2018) 1697–1710.

[43] C. Tang, Z. Li, J. Wang, X. Liu, W. Zhang, E. Zhu, Unified one-step multi-view spectral clustering, IEEE Trans. Knowl. Data Eng. 35 (6) (2022) 6449–6460.

[44] F. Zhang, J. Zhao, X. Ye, H. Chen, One-step adaptive spectral clustering networks, IEEE Signal Process. Lett. 29 (2022) 2263–2267.

[45] P.W. Holland, K.B. Laskey, S. Leinhardt, Stochastic blockmodels: First steps, Soc. Netw. 5 (2) (1983) 109–137.

[46] J. Lei, A. Rinaldo, Consistency of spectral clustering in stochastic block models, Ann. Statist. 43 (1) (2015).

[47] Q. Li, X.-M. Wu, H. Liu, X. Zhang, Z. Guan, Label efficient semi-supervised learning via graph filtering, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 9582–9591.

[48] Y.Y. Pilavcı, P.-O. Amblard, S. Barthelmé, N. Tremblay, Graph tikhonov regularization and interpolation via random spanning forests, IEEE Trans. Signal. Inf. Process. Netw. 7 (2021) 359–374.

[49] E. Pan, Z. Kang, Multi-view contrastive graph clustering, Proc. Adv. Neural Inf. Process. Syst. 34 (2021) 2148–2159.

[50] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008).

[51] J.-H. Yang, C. Chen, H.-N. Dai, M. Ding, Z.-B. Wu, Z. Zheng, Robust corrupted data recovery and clustering via generalized transformed tensor low-rank representation, IEEE Trans. Neural Netw. Learn. Syst. (2022).

[52] K. Fan, On a theorem of weyl concerning eigenvalues of linear transformations I, Proc. of the Nat. Academy. of Sci. 35 (11) (1949) 652–655.

[53] S. Kumar, J. Ying, J.V. de Miranda Cardoso, D.P. Palomar, A unified framework for structured graph learning via spectral constraints, J. Mach. Learn. Res. 21 (22) (2020) 1–60.

[54] D. Wu, F. Nie, J. Lu, R. Wang, X. Li, Effective clustering via structured graph learning, IEEE Trans. Knowl. Data Eng. (2022).

[55] S.S. Saboksayr, G. Mateos, Accelerated graph learning from smooth signals, IEEE Signal Process. Lett. 28 (2021) 2192–2196.

[56] Z. Wen, W. Yin, A feasible method for optimization with orthogonality constraints, Math. Program. 142 (2013) 397–434.

[57] P.H. Schönemann, A generalized solution of the orthogonal procrustes problem, Psychometrika 31 (1) (1966) 1–10.

[58] O. Axelsson, G. Lindskog, On the rate of convergence of the preconditioned conjugate gradient method, Numer. Math. 48 (1986) 499–523.

[59] V. Kalofolias, N. Perraudin, Large scale graph learning from smooth signals, in: Int. Conf. Learn. Representations, 2019, pp. 1–12.

[60] D.A. Tarzanagh, L. Balzano, A.O. Hero, Fair structure learning in heterogeneous graphical models, 2021, arXiv:2112.05128.

[61] H. Wang, N. Wang, D.-Y. Yeung, Collaborative deep learning for recommender systems, in: Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2015, pp. 1235–1244.

Xiang Zhang received the B.Sc. degree and M.Sc. degree in signal processing from the Nanjing University of Aeronautics and Astronautics, Jiangsu, China, in 2016 and 2019. He is currently working toward the Ph.D. degree with the School of Information Science and Engineering, Southeast University, Nanjing, China. His research interests include graph signal processing and graph machine learning.

Qiao Wang was born in Anqing, Anhui, China, in 1966. He received the B.S., M.S., and Ph.D. degrees in mathematics from Wuhan University, Wuhan, China, in 1988, 1994, and 1997, respectively. In 1997, he joined the School of Information Science and Engineering, Southeast University, Nanjing, China, and was appointed as an Associate Professor, in 1999, then a Full Professor, in 2001. From 2003 to 2004, he was a Visiting Scientist with Harvard University, Cambridge, MA, USA. His research interests include urban science and urban design, data science, harmonic analysis, applied statistics, and information theory. He was the recipient of the Excellence Design by the International Society of the Built Environment, in 2019, first prize of China Construction Science and Technology Award in 2020 for research in urban design, 2021 National Excellent Urban Design First Prize, and second prize of the Science and Technology Progress Award of the Ministry of Education of China in 2021 for his research on data analysis of the diagnosis and treatment of depression. He is currently the Executive Editor of the ICT Express of Elsevier.