Journal of Computational Mathematics http://www.global-sci.org/jcm
Vol.xx, No.x, 2025, 1–19. doi:10.4208/jcm.2506-m2024-0128

A MATRIX OPTIMIZATION METHOD FOR BLIND EXTRACTION OF EXTERNAL EQUITABLE PARTITIONS FROM LOW PASS GRAPH SIGNALS[*]

Wenshun Teng
School of Mathematics and Statistics, Beijing Institute of Technology, Beijing 100081, China
Email: teng_wenshun@163.com
Qingna Li[1)]
School of Mathematics and Statistics, Beijing Key Laboratory on MCAACI, Beijing Institute of Technology, Beijing 100081, China
Email: qnl@bit.edu.cn

Abstract

Seeking the external equitable partitions (EEPs) of networks under unknown structures is an emerging problem in network analysis. The special structure of EEPs has found widespread applications in many fields such as cluster synchronization and consensus dynamics. While most literature focuses on utilizing the special structural properties of EEPs for network studies, there has been little work on the extraction of EEPs or their connection with graph signals. In this paper, we address the interesting connection between low pass graph signals and EEPs, which, as far as we know, is the first time. We provide a method BE-EEPs for extracting EEPs from low pass graph signals and propose an optimization model, which is essentially a problem involving nonnegative orthogonality matrix decomposition. We derive theoretical error bounds for the performance of our proposed method under certain assumptions and apply three algorithms to solve the resulting model, including the K-means algorithm, the practical exact penalty method and the iterative Lagrangian approach. Numerical experiments verify the effectiveness of the proposed method. Under strong low pass graph signals, the iterative Lagrangian and K-means perform equally well, outperforming the exact penalty method. However, under complex weak low pass signals, all three perform equally well.

Mathematics subject classification: 68R10, 05C90, 90C35, 90C90.
Key words: Matrix optimization model, External equitable partitions, Low pass graph signals, Topology inference, Nonnegative orthogonality constraints, Error bound.

1. Introduction

The extraction of external equitable partitions is an emerging problem in network analysis [5]. EEPs are a graph topological structure, which is a generalization of the well-known equitable partitions (EPs) [17], also known as almost equitable partitions (AEPs) [7] in graph theory. Cardoso *et al.* [7] introduced AEPs and provided their existence. Previous research has focused on various areas such as graph coloring [33] and automorphism [23] in graph theory. Recent studies have mostly focused on network dynamic processes. EEPs have an intriguing geometric interpretation, revealing spectral properties of the Laplacian matrix [1,25]. This property opens up more possibilities for network research, such as cluster synchronization [38, 41],

[*] Received October 30, 2024 / Revised version received January 29, 2025 / Accepted June 23, 2025 / Published online September 25, 2025 /

[1)] Corresponding author

consensus dynamics [10, 28, 54], network control problems [14, 15, 22], and epidemic spreading [3, 11]. EEPs have even been utilized to aid in studying the microstructure of molecules and crystalline solids [24]. Additionally, the research by Schaub *et al.* [40] involves the task of finding approximate EEPs. To address this, they propose minimizing a projection error problem and utilize K-means clustering to solve it.

On the other hand, the low pass graph signal model is a hot topic in the field of graph signal processing (GSP) [29, 45], with widespread applications in various domains such as economics, social networks, and power systems [35]. Many physical and social processes are naturally characterized by low pass graph filters, such as diffusion model [46], opinion dynamics [36], and soon. Recent research mainly focuses on two aspects:

(i) Due to the significant role played by low pass graph filters in tasks related to sampling [2], denoising [31], graph topology inference [13], and graph neural networks [27], most studies typically develop relevant graph signal processing algorithms with low pass attributes as a default assumption.

(ii) Considering scenarios where the properties of observed graph signals are unknown, such as systems under attack [39] or observed data being corrupted [16], many researchers studied the blind detection problem of low pass graph signals. This involves detecting whether a set of observed graph signals is generated by a low pass graph filter, as studied in [19, 26, 56].

Our work is related to graph topology inference, with recent studies focusing on blind inference of communities [47, 48] and centrality [18, 20, 37].

Having summarized the existing work on EEPs and low pass graph signal models, it seems that little work has been done on the connection of EEPs and low pass graph signal models. A natural question is whether the two topics can be related to each other and what is the exact connection between the two? Scholkemper and Schaub [42] established a link between EPs and a specific graph signal model, which inspired our research. In this paper, we aim to answer this question by establishing the mathematical model of EEPs via low pass graph signals.

The contribution of this paper is as follows. Firstly, we reveal the interesting connection between EEPs and low pass graph signals. Secondly, we propose a method for learning the EEP via low pass graph signals, which involves constructing an optimization model. Finally, we provide theoretical analysis for the proposed method and apply three algorithms to solve it. This paper opens a door to investigate EEPs from a new point of view, and provide a new approach to extract the EEP of a graph.

The organization of this paper is as follows. In Section 2, we introduce some preliminaries about EEPs of a graph and graph signal models. In Section 3, we establish the optimization model for extraction of EEPs via low pass graph filters and provide a method, namely method for blind extraction of external equitable partitions (BE-EEPs), for extracting EEPs from low pass graph signals. In Section 4, we derive theoretical error bounds for the performance of BE-EEPs under certain assumptions. In Section 5, we discuss how to solve the proposed optimization model. In Section 6, we present some numerical results. Final conclusions are made in Section 7.

2. Preliminaries

In this section, we introduce some preliminaries about EEPs of a graph and graph signal models.

2.1. External equitable partitions of a graph

For an undirected simple graph $G = (V(G), E(G))$, $V(G) = \{1, 2, \dots, n\}$ is a nonempty set of vertices, $E(G) = \{(i,j) \mid i,j = 1, \dots, n, i \text{ and } j \text{ are connected}\}$ is a set of edges. A partition π of $V(G)$ with cells $\mathbb{C} = \{C_1, C_2, \cdots, C_r\}$ means that $V(G)$ is divided into disjoint sets C_i, $i = 1, \dots, r$ such that $C_i \cap C_j = \emptyset, i \neq j, j = 1, \dots, r$, and $V(G) = \bigcup_i C_i$. We say that a partition π of $V(G)$ with cells $C_1, \cdots, C_r$ is an equitable partition if the number of neighbours in C_j of a vertex u in C_i is a constant b_{ij}, independent of u, where $i,j = 1, \dots, r$. Similarly, a partition π of $V(G)$ with cells $C_1, \cdots, C_r$ is an external equitable partition if the number of neighbours in C_j of a vertex u in C_i is a constant b_{ij}, independent of u, where $i \neq j$ and $i,j = 1, \dots, r$. Note that compared with EP, in EEP, only for $i \neq j$, it is required that all the vertices in the class C_i have the same number of neighbors in $C_j, i,j = 1, 2, \dots, r$. Let G/π denote the quotient graph in which each vertex represents a cell of the EEP π and the arcs between the new vertices indicate the links between cells in the original graph G (see Fig. 2.1). Note that the quotient graph G/π is a multi-digraph.

To characterize a partition, a binary indicator matrix [53] $H \in \mathbb{R}^{n \times r}$ is often used, where

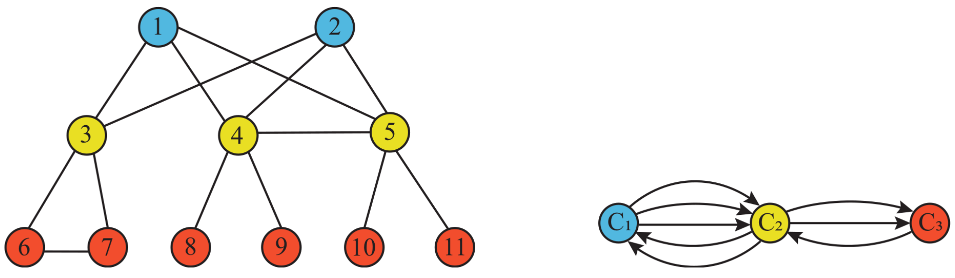
$$H_{ij} = \begin{cases} 1, & \text{if} \quad i \in C_j, \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

Next, for an EEP π of G, let $A^G \in \{0,1\}^{n \times n}$ be the adjacency matrix of the graph G in which $(A^G)_{ij} = 1$, if $(i,j) \in E(G)$, and $(A^G)_{ij} = 0$ otherwise. Let $A^{G/\pi} \in \mathbb{R}^{r \times r}$ be the adjacency matrix of the quotient graph G/π in which $(A^{G/\pi})_{ij} = b_{ij}$, if $i \neq j$, and $(A^{G/\pi})_{ij} = 0$ otherwise, where b_{ij} is the constant defined in EEP. Let $L^G \in \mathbb{R}^{n \times n}$ and $L^{G/\pi} \in \mathbb{R}^{r \times r}$ be the Laplacian matrix of the graph G and the quotient graph G/π, that is,

$$L^G = D^G - A^G, \quad L^{\frac{G}{\pi}} = D^{\frac{G}{\pi}} - A^{\frac{G}{\pi}},$$

where $D^G = \text{diag}(A^G \mathbf{1})$ and $D^{G/\pi} = \text{diag}(A^{G/\pi} \mathbf{1})$ are diagonal matrices with the diagonal elements being $A^G \mathbf{1}$ and $A^{G/\pi} \mathbf{1}$ respectively, and $\mathbf{1}$ is an all-one vector of proper size. Although L^G is symmetric, $L^{G/\pi}$ will be asymmetric in general. The Laplacian matrix of the quotient graph G/π ignores the internal connectivity. Note that, from the definition of the Laplacian, there is always a trivial EEP in which the whole graph is grouped into one class, i.e. $H = \mathbf{1}$ and $L^{G/\pi} = 0$. Moreover, every EP is necessarily an EEP, while the converse is not true.

Based on [7,17,40], we have the following properties which will be used later. More properties can be found in [7,17,40,41].



(a) The original graph G

(b) Quotient graph G/π

Fig. 2.1. The original graph G with $n = 11$ vertices with an EEP π into three cells (indicated with colors) and its associated quotient graph G/π.

Proposition 2.1. *Let G be a graph, and π be a partition of $V(G)$ encoded by the indicator matrix H. The following results hold:*

(i) The partition π is an EEP if and only if $L^G H = HL^{G/\pi}$.

(ii) If π is an EEP, then there exist r eigenvectors contained in $V^G = [v_1, \cdots, v_r] \in \mathbb{R}^{n\times r}$ for the Laplacian matrix L^G such that $V^G = HV^{G/\pi}$, where $V^{G/\pi} = [v_1^{G/\pi}, \cdots, v_r^{G/\pi}]$ contains r eigenvectors of $L^{G/\pi}$.

Definition 2.1. *For an EEP denoted as π, if there exists an eigenvector $v \in \mathbb{R}^n$ of L^G and an eigenvector $v^{G/\pi} \in \mathbb{R}^r$ of $L^{G/\pi}$ such that $v = Hv^{G/\pi}$, then v is referred to as a structural eigenvector of L^G.*

2.2. Graph signal model

Recall the undirected graph $G = (V(G), E(G))$ defined in Section 2.1. Since L^G is real symmetric and positive semidefinite [48], it can be written as

$$L^G = V\Lambda V^\top, \tag{2.2}$$

where $V = [v_1, v_2, \cdots, v_n]$ is the eigenvector matrix that contains the eigenvectors as columns, and $\Lambda = \text{diag}[\lambda_1, \lambda_2, \cdots, \lambda_n]$ is the diagonal eigenvalue matrix. The eigenvalues are ordered as $0 = \lambda_1 \leq \cdots \leq \lambda_n$. Note that V is an orthogonal matrix.

A signal on the graph G is defined as a function on the vertices of G, $f : V(G) \to \mathbb{R}$. For the sake of description, a signal on the graph G can be represented as a vector

$$\mathbf{x} := [x_1, x_2, \cdots, x_n] \in \mathbb{R}^n,$$

that attaches a value x_i to the vertex i in $V(G)$, $i = 1, 2, \ldots, n$. The graph signal in Fig. 2.2 is one such example. The output graph signal $\mathbf{y} = [y_1, y_2, \cdots, y_n] \in \mathbb{R}^n$ of a graph filter $\mathcal{H}$ applied to the input graph signal $\mathbf{x}$ is

$$\mathbf{y} = \mathcal{H}\mathbf{x}, \tag{2.3}$$

where $\mathcal{H}$ is represented by matrices called a graph shift operator[1)] [47], which can be used to define linear graph filters. Here, we use the graph Laplacian matrix L^G as a graph shift operator. A graph filter $\mathcal{H}$ of order T ($T \leq +\infty$) can be expressed as a polynomial of matrix in the graph shift operator L^G of degree T (recall (2.2))

$$\mathcal{H}(L^G) := \sum_{t=0}^{T} a_t (L^G)^t = V\text{diag}\big(h(\lambda_1), \cdots, h(\lambda_n)\big)V^\top, \tag{2.4}$$

where $h(\cdot) : \mathbb{R} \to \mathbb{R}$ is the scalar generating polynomial defined by

$$h(\mu) = \sum_{t=0}^{T} a_t \mu^t, \quad \mu \in \mathbb{R}, \tag{2.5}$$

and $a_t \in \mathbb{R}$, $t = 0, \ldots, T$ is the filter weight. Let the filtered graph signals be given by $\mathbf{y} = \mathcal{H}(L^G)\mathbf{x}$. Now we describe the definition of a low pass graph filter.

[1)] Graph shift operator is commonly used as the graph Laplacian matrix L^G and the adjacency matrix A^G. See [45] for an overview on the subject.

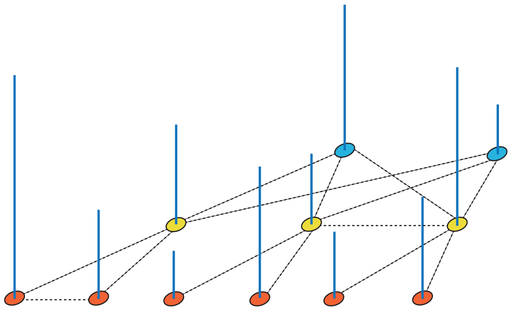


Fig. 2.2. A graph signal on the vertices of graph G in Fig. 2.1. The height of each blue bar represents the magnitude of the signal values.

Definition 2.2 ([35]). *We say that a graph filter $\mathcal{H}(L^G)$ is r-low-pass if for any $1 \leq r \leq n-1$,*

$$\eta_r := \frac{\max\{|h(\lambda_{r+1})|, \cdots, |h(\lambda_n)|\}}{\min\{|h(\lambda_1)|, \cdots, |h(\lambda_r)|\}} \in [0,1). \tag{2.6}$$

The ratio η_r quantifies the strength of the low pass graph filter. Generally, if $\eta_r \approx 0$, then we can say the low pass graph filter is strong; if $\eta_r \approx 1$, then the low pass graph filter is weak.

Definition 2.2 implies that the eigenvectors of a r-low-pass graph filter $\mathcal{H}(L^G)$ corresponding to the largest r eigenvalues are the left-most r vectors in V. For the sake of convenience, in this paper, we simply refer to such filter as a low pass graph filter. A low pass graph signal is defined as a graph signal generated by the excitation of a low pass filter, including but not limited to white noise. Using Definition 2.2, the ideal low pass graph filter has $\eta_r = 0$. Low pass graph filters are commonly employed in modeling practical network data, as detailed in [35]. Below we present two examples.

Example 2.1 ([20, 46]). The graph filter $\mathcal{H}(L^G)$ defined by

$$\mathcal{H}(L^G) = \sum_{t=0}^{+\infty} \frac{\sigma^t}{t!}(L^G)^t$$

is a strong low pass graph filter for large σ. Corresponding to the above (2.4), we know that $a_t = \sigma^t/t!$ and $T = +\infty$. According to the definition of the series, we can also write it as $\mathcal{H}(L^G) = e^{-\sigma L^G}$. Note that

$$\eta_r = \frac{e^{-\sigma\lambda_{r+1}}}{e^{-\sigma\lambda_r}} = e^{-\sigma(\lambda_{r+1}-\lambda_r)}.$$

Since $\lambda_{r+1} > \lambda_r$ and $\sigma > 0$, the low pass ratio decreases to zero exponentially with σ.

Example 2.2 ([20]). The graph filter $\mathcal{H}(L^G)$ defined by

$$\mathcal{H}(L^G) = \sum_{t=0}^{+\infty} (-\alpha)^t (L^G)^t$$

with $\alpha > 0$, is a weak low pass graph filter for small α. Corresponding to the above (2.4), we know that $a_t = (-\alpha)^t$ and $T = +\infty$. We can also express the above graph filter as $\mathcal{H}(L^G) = (I + \alpha L^G)^{-1}$. Note that

$$\eta_r = \frac{1+\alpha\lambda_r}{1+\alpha\lambda_{r+1}} = 1 - \alpha\frac{\lambda_{r+1}-\lambda_r}{1+\alpha\lambda_r}.$$

Since $\lambda_{r+1} > \lambda_r$ and $\alpha > 0$, the low pass ratio $\eta_r \approx 1$ when α is sufficiently small.

3. Blind Extraction with Low Pass Graph Filters

In this section, we establish the interesting connection between EEP and low pass graph signals. Moreover, we propose Algorithm 3.2, namely BE-EEPs, to address this blind extraction problem by establishing the matrix optimization model for extraction of EEPs via low pass graph filters.

3.1. Problem statement

Given the observations $\mathbf{y}^1, \cdots, \mathbf{y}^m \in \mathbb{R}^n$, which are the graph filter's outputs subject to the unknown excitation $\mathbf{x}^1, \cdots, \mathbf{x}^m \in \mathbb{R}^n$

$$\mathbf{y}^l = \mathcal{H}(L^G)\mathbf{x}^l + \omega^l, \quad l = 1, \dots, m, \tag{3.1}$$

where $\mathcal{H}(L^G)$ is defined by (2.4), $\omega^l \in \mathbb{R}^n$ represents the noise satisfying normal distribution $\omega^l \sim \mathcal{N}(0, \sigma^2 I)$, our aim is to find an EEP π of G from the graph filters. The following assumptions are standard assumptions based on which will be convenient for our analysis.

Assumption 3.1. The graph filter $\mathcal{H}(L^G)$ of order T is a low pass graph filter.

Assumption 3.2. The excitation graph signals $\mathbf{x}^l$ is zero-mean white noise with

$$\mathbb{E}\big[\mathbf{x}^l(\mathbf{x}^l)^\top\big] = I, \quad l = 1, \dots, m.^{1)}$$

With Assumptions 3.1 and 3.2, the covariance matrix of the observed signals $\mathbf{y}^l$, is given by

$$\Sigma := \mathbb{E}\big[\mathbf{y}^l(\mathbf{y}^l)^\top\big] = \mathcal{H}(L^G)\mathcal{H}(L^G)^\top + \sigma^2 I. \tag{3.2}$$

To derive EEP of G from low pass graph filters, the indicator matrix $H \in \mathbb{R}^{n\times r}$ can be used to derive the EEP π. Take the graph in Fig. 2.1(a) as an example. We clearly know that

$$L^G = \left[\begin{array}{cc:ccc:cccccc} 3 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline -1 & -1 & 4 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 5 & -1 & 0 & 0 & -1 & -1 & 0 & 0 \\ -1 & -1 & 0 & -1 & 5 & 0 & 0 & 0 & 0 & -1 & -1 \\ \hdashline 0 & 0 & -1 & 0 & 0 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right], \quad H = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 1 & 0 & 0 \\ \hdashline 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ \hdashline 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{array}\right],$$

and

$$L^{\frac{G}{\pi}} = \begin{bmatrix} 3 & -3 & 0 \\ -2 & 4 & -2 \\ 0 & -1 & 1 \end{bmatrix}.$$

One can verify that $L^G H = H L^{G/\pi}$. Then we can calculate that $\mathbb{C} = \{C_1, C_2, C_3\}$ and

$$C_1 = \{1, 2\}, \quad C_2 = \{3, 4, 5\}, \quad C_3 = \{6, 7, 8, 9, 10, 11\}.$$

The details are summarized in Algorithm 3.1.

[1)] Most prior work on graph learning made this assumption about the excitation graph signals $\mathbf{x}^l, l = 1, \dots, m$, such as in [37, 42, 43, 48].This assumption implies that the excitations at each node are independent.

Algorithm 3.1: The Procedure to Finding EEP π through an Indicator Matrix H.

```
Input  : An indicator matrix $H \in \mathbb{R}^{n \times r}$, an $n$-dimensional zero vector
         $\mathbf{c} = \{c_1, \cdots, c_n\}$.
For $i = 1, 2, \ldots, n$, set $c_i = k$ if $H_{ik} \neq 0$, $k = 1, 2, \ldots, r$.
$C_k = \{i : c_i = k\}$, where $k = 1, 2, \ldots, r$ and $i = 1, 2, \ldots, n$.
Output: An EEP $\pi$ with $\mathbb{C} = \{C_1, C_2, \cdots, C_r\}$.
```

3.2. Connections between EEP and low pass graph filters

Note that as mentioned in Section 3.1, EEP of a graph can be derived by the indicator matrix H. In this part, we will derive the relation between the indicator matrix H and the low pass graph filters. The rough idea is as follows. We first establish the relationship between $\mathcal{H}(L^G)$ and H in Proposition 3.1. Combining Proposition 2.1, we derive that $\mathcal{H}(L^G)$ also possesses structural eigenvectors as described in Proposition 2.1(ii). Furthermore, through Proposition 3.2, we infer that starting from the covariance matrix, we can obtain the indicator matrix H and consequently derive the EEP π with $\mathbb{C} = \{C_1, C_2, \cdots, C_r\}$.

Proposition 3.1. *For an EEP π as indicated by H, it holds that $\mathcal{H}(L^G)H = H\mathcal{H}(L^{G/\pi})$.*

Proof. Note that $\mathcal{H}(L^G)$ is a polynomial of the matrix L^G. According to $L^G H = HL^{G/\pi}$ in Proposition 2.1(i), it holds that $(L^G)^r H = H(L^{G/\pi})^r$ for any nonnegative integer r. Therefore, by (2.4), it holds that

$$\mathcal{H}(L^G)H = \sum_{t=0}^{T} a_t (L^G)^t H = \sum_{t=0}^{T} a_t H \big(L^{\frac{G}{\pi}}\big)^t = H\mathcal{H}\big(L^{\frac{G}{\pi}}\big).$$

The proof is complete. □

Since the eigenvectors of L^G are also eigenvectors of the polynomial of the matrix L^G, that is, the structural eigenvectors of L^G are also the eigenvectors of $\mathcal{H}(L^G)$.

Definition 3.1. *The structural eigenvectors of L^G are defined as the structural eigenvectors of $\mathcal{H}(L^G)$.*

The following result reveals the equality relationship between the structural eigenvectors of $\mathcal{H}(L^G)$ and some of the eigenvectors of the covariance matrix Σ.

Proposition 3.2. *For an EEP π as indicated by H, the eigenvectors of Σ in the form of $H\tilde{v}$ are the structural eigenvectors of $\mathcal{H}(L^G)$, where $\tilde{v}$ is an eigenvector of $\mathcal{H}(L^{G/\pi})$.*

Proof. Let $\mathcal{H}(L^{G/\pi})\tilde{v} = \tilde{\lambda}\tilde{v}$, where $\tilde{v}$ is an eigenvector of $\mathcal{H}(L^{G/\pi})$ with eigenvalue $\tilde{\lambda}$, we obtain that

$$\begin{aligned}
\Sigma H\tilde{v} &= \mathcal{H}(L^G)\big(\mathcal{H}(L^G)\big)^\top H\tilde{v} + \sigma^2 I H\tilde{v} \\
&= \mathcal{H}(L^G)\mathcal{H}(L^G)H\tilde{v} + \sigma^2 I H\tilde{v} \\
&= \mathcal{H}(L^G)H\mathcal{H}\big(L^{\frac{G}{\pi}}\big)\tilde{v} + \sigma^2 I H\tilde{v} \\
&= H\big(\mathcal{H}\big(L^{\frac{G}{\pi}}\big)\big)^2\tilde{v} + \sigma^2 I H\tilde{v} \\
&= (\tilde{\lambda}^2 + \sigma^2)H\tilde{v}, \qquad (3.3)
\end{aligned}$$

where the second equality stems from the fact that $\mathcal{H}(L^G)$ is a symmetric matrix; the third equality and the forth equality are from Proposition 3.1; the fifth equality comes from $\mathcal{H}(L^{G/\pi})\tilde{v} = \tilde{\lambda}\tilde{v}$. Hence, the eigenvectors of Σ in the form of $H\tilde{v}$ are the structural eigenvectors of $\mathcal{H}(L^G)$. The proof is complete. □

Definition 3.2. *Let v be an eigenvector of Σ. For an EEP π as indicated by H, if there exists an eigenvector of $\mathcal{H}(L^{G/\pi})$, denoted as $\tilde{v}$, such that $v = H\tilde{v}$, then v is referred to as a structural eigenvector of Σ.*

Through the two propositions above, for low pass graph signal models that satisfy Assumptions 3.1 and 3.2, the covariance matrix contains structural eigenvectors that reflect the indicator matrix. Therefore, by computing the eigenvectors of the covariance matrix and identifying the structural eigenvectors among them, we can determine the indicator matrix H and subsequently find the EEP. The process of identifying these structural eigenvectors is described in Section 3.4.

3.3. A function related to structural eigenvectors and an indicator matrix

For convenience, we perform the spectral decomposition of $\mathcal{H}(L^G)$ and Σ respectively. Let $\overline{V} \in \mathbb{R}^{n\times r}$ be a matrix containing r structural eigenvectors of $\mathcal{H}(L^G)$. Let $P \in \mathbb{R}^{n\times r}$ be a matrix containing r structural eigenvectors of Σ, and is associated with r eigenvalues $\xi_1 \geq \cdots \geq \xi_r$. Then we can get that $\overline{V} = P$ due to Propositions 3.1 and 3.2. Define the cost function [42] as follows:

$$F(\mathbb{C}, V) := \sum_{k=1}^{r}\sum_{i\in C_k}\left\| V_{i,\cdot} - \frac{1}{|C_k|}\sum_{j\in C_k} V_{j,\cdot}\right\|_2^2, \tag{3.4}$$

where $V_{i,\cdot}$ denotes the i-th row of V. For an EEP, the components of any structural eigenvector corresponding to the same class do not change. Therefore, for an EEP $\overline{\pi}$ of G with

$$\overline{\mathbb{C}} = \{\overline{C}_1, \overline{C}_2, \cdots, \overline{C}_r\},$$

we obtain that $F(\overline{\mathbb{C}}, \overline{V}) = F(\overline{\mathbb{C}}, P) = 0$.

To reformulate (3.4) in the form of a binary indicator matrix $H \in \mathbb{R}^{n\times r}$ of the partition π with $\mathbb{C}$, we define a normalized indicator matrix (see [4])

$$\widehat{H} = H\mathrm{diag}\left(\frac{1}{\sqrt{|C_1|}}, \frac{1}{\sqrt{|C_2|}}, \cdots, \frac{1}{\sqrt{|C_r|}}\right), \tag{3.5}$$

where $|C_i|$ denote the number of vertices in the set C_i. It holds that

$$\left(\widehat{H}\widehat{H}^\top\right)_{ij} = \begin{cases} \frac{1}{|C_k|}, & \text{if} \quad i \in C_k, \quad j \in C_k, \quad k = 1, 2, \ldots, r, \\ 0, & \text{otherwise}. \end{cases} \tag{3.6}$$

It gives the following:

$$\sum_{k=1}^{r}\sum_{i\in C_k}\left\| V_{i,\cdot} - \frac{1}{|C_k|}\sum_{j\in C_k} V_{j,\cdot}\right\|_2^2 = \left\|V - \widehat{H}\widehat{H}^\top V\right\|_F^2. \tag{3.7}$$

Let $\overline{H}$ be the normalized indicator matrix corresponding to the EEP $\overline{\pi}$. Then we know that $\|\overline{V} - \overline{H}(\overline{H})^\top\overline{V}\|_F^2 = 0$. Below, we will use (3.7) to relate the structural eigenvectors and an indicator matrix to construct an optimization model.

3.4. The optimization model

As mentioned above, our goal is to derive the EEP of the graph G from the outputs of an unknown network process represented by a low pass graph filter. Given the observed signals $\mathbf{y}^l, l = 1, 2, \ldots, m$, instead of the covariance matrix Σ, the following sample covariance matrix (denoted by $\widehat{\Sigma}$):

$$\widehat{\Sigma} = \frac{1}{m}\sum_{l=1}^{m} \mathbf{y}^l(\mathbf{y}^l)^\top \tag{3.8}$$

is available in practice.

According to [42, Proposition 1] and Assumption 3.1, most of top r eigenvectors of $\widehat{\Sigma}$ are structural eigenvectors. Hence we can simply choose the top r eigenvectors of $\widehat{\Sigma}$ as the structural eigenvectors. Subsequently, we will utilize these top r eigenvectors of $\widehat{\Sigma}$ to approximately extract the EEP.

Let $\widehat{P} \in \mathbb{R}^{n\times r}$ be a matrix containing r estimated structural eigenvectors of $\widehat{\Sigma}$ with r eigenvalues $\hat{\xi}_1 \geq \cdots \geq \hat{\xi}_r$. Recall the cost function in (3.4). We minimize the cost function with respect to the estimated structural eigenvectors $\widehat{P}$, which gives the following model:

$$\min_{\mathbb{C}} f(\mathbb{C}, \widehat{P}) := \min_{\mathbb{C}} \sum_{k=1}^{r}\sum_{i\in C_k} \left\| \widehat{P}_{i,\cdot} - \frac{1}{|C_k|}\sum_{j\in C_k} \widehat{P}_{j,\cdot} \right\|_2^2. \tag{3.9}$$

Keeping (3.7) in mind, for the graph signal model with low pass graph filters, we can search for an approximate EEP by solving the following optimization problem (recall that $\widehat{P}$ is given):

$$\min_{\widehat{H}\in\mathbb{H}_r} \|\widehat{P} - \widehat{H}\widehat{H}^\top\widehat{P}\|_F^2, \tag{3.10}$$

where

$$\mathbb{H}_r = \left\{\widehat{H} \in \mathbb{R}^{n\times r} \,|\, \widehat{H}^\top\widehat{H} = I_r,\ \widehat{H}(\widehat{H})^\top\mathbf{1}_n = \mathbf{1}_n,\ \|\widehat{H}_{i,:}\|_0 = 1, \text{ and } \widehat{H} \geq 0\right\}.$$

The constraint $\|\widehat{H}_{i,:}\|_0 = 1$ shows that $\widehat{H}$ have exactly one non-zero element per row, which denotes the clustering membership. The constraints $\widehat{H} \geq 0, \widehat{H}^\top\widehat{H} = I_r$ and $\widehat{H}(\widehat{H})^\top\mathbf{1}_n = \mathbf{1}_n$ indicate that $\widehat{H}$ is a normalized indicator matrix. $\widehat{H}^\top\widehat{H} = I_r$ ensures that each column of $\widehat{H}$ contains non-zero elements. Furthermore, $\widehat{H}(\widehat{H})^\top\mathbf{1}_n = \mathbf{1}_n$ ensures that the non-zero elements in the columns of $\widehat{H}$ are consistent. The following algorithm, named as BE-EEPs, summarizes the above procedure in a clear framework.

Algorithm 3.2: BE-EEPs: Blind EEPs Extraction by Low Pass Graph Signal.

```
Input  : Graph signals $\{\mathbf{y}^l\}_{l=1}^m$; desired number of classes $r$.
Compute the sample covariance $\widehat{\Sigma}$ as (3.8).
Find the top $r$ eigenvectors of $\widehat{\Sigma}$, denoted as $\widehat{P} \in \mathbb{R}^{n\times r}$.
Solve (3.10) to get $\widehat{H}$.
Use Algorithm 3.1 to get $\widehat{\mathbb{C}}$.
Output: $\widehat{H}$ and $\widehat{\mathbb{C}} = \{\widehat{C}_1, \widehat{C}_2, \cdots, \widehat{C}_r\}$.
```

4. Error Bound Analysis

In this part, we derive theoretical error bounds for the performance of Algorithm 3.2 under certain assumptions. We need the following result, which comes from [42, Theorem 2].

Theorem 4.1 ([42, Theorem 2]). *Let* $\mathbf{y}^1, \cdots, \mathbf{y}^m$ *be independent samples of the graph filter as in* (3.1). *Assume that for some* $K \geq 1$,

$$\|\mathbf{y}^l\|_2 \leq K\big(\mathbb{E}(\|\mathbf{y}^l\|_2^2)\big)^{\frac{1}{2}}, \quad l = 1, \ldots, m$$

almost surely. Let the effective rank of the covariance be $b = \mathrm{tr}(\Sigma)/\|\Sigma\|_2$, *then for every* $c' > 0$ *it holds that*

$$\|\widehat{\Sigma} - \Sigma\|_2 \leq \epsilon \left(\sqrt{\frac{K^2 b \log(n/c')}{m}} + \frac{K^2 b \log(n/c')}{m} \right) \|\Sigma\|_2 \tag{4.1}$$

with probability at least $1 - c'$, *where* ϵ *is an constant.*

Theorem 4.1 bounds the difference between sample covariance and covariance. Following this, and based on Theorem 4.1, we provide the upper bound for the difference between the cost function $F(\overline{\mathbb{C}}, \overline{V})$ under ideal conditions and the cost achieved by the partition

$$\widehat{\mathbb{C}} = \{\widehat{C}_1, \widehat{C}_2, \cdots, \widehat{C}_r\},$$

identified by Algorithm 3.2, as demonstrated in Theorem 4.2.

Theorem 4.2. *Let* $\mathbf{y}^1, \cdots, \mathbf{y}^m$ *be independent samples of the graph filter as in* (3.1). *Let* $b = \mathrm{tr}(\Sigma)/\|\Sigma\|_2$ *and the following conditions hold:*

(i) Algorithm 3.2 *finds a solution* $\widehat{\mathbb{C}} = \{\widehat{C}_1, \widehat{C}_2, \cdots, \widehat{C}_r\}$ *that exactly minimizes the problem* (3.10).

(ii) There exists $\delta > 0$ *such that* $\|\widehat{\Sigma} - \Sigma\|_2 + \delta \leq \xi_r - \hat{\xi}_{r+1}$.

(iii) For some $K \geq 1$,

$$\|\mathbf{y}^l\|_2 \leq K\big(\mathbb{E}(\|\mathbf{y}^l\|_2^2)\big)^{\frac{1}{2}}, \quad l = 1, \ldots, m$$

almost surely.

Then for every $c' > 0$, *it holds that*

$$\sqrt{F(\widehat{\mathbb{C}}, \overline{V})} - \sqrt{F(\overline{\mathbb{C}}, \overline{V})} \leq 2\sqrt{2r}\frac{\epsilon}{\delta} \left(\sqrt{\frac{K^2 b \log(n/c')}{m}} + \frac{K^2 b \log(n/c')}{m} \right) \|\Sigma\|_2 \tag{4.2}$$

with probability at least $1 - c'$, *where* ϵ *is an constant.*

Proof. It follows from (3.7) and $F(\overline{\mathbb{C}}, \overline{V}) = 0$ that

$$\sqrt{F(\widehat{\mathbb{C}}, \overline{V})} - \sqrt{F(\overline{\mathbb{C}}, \overline{V})} = \big\|\overline{V} - \widehat{H}\widehat{H}^\top \overline{V}\big\|_F.$$

The rest of the proof is similar to that in the proof of [42, Theorem 1]. □

Eq. (4.2) implies that the projection error associated with the estimated structural eigenvectors $\widehat{P}$ will be small if the sample size is large enough.

5. Three Methods for Solving (3.10)

In this part, we will discuss how to solve (3.10) in Algorithm 3.2 using three methods.

To be precise, (3.10) can also be written in the following matrix optimization problem (recall that $\widehat{P}$ is given)

$$\begin{aligned} \min_{\widehat{H}\in\mathbb{R}^{n\times r}} \quad & \left\|\widehat{P} - \widehat{H}\widehat{H}^\top \widehat{P}\right\|_F^2 \\ \text{s.t.} \quad & \widehat{H}^\top \widehat{H} = I_r, \\ & \widehat{H}(\widehat{H})^\top \mathbf{1}_n = \mathbf{1}_n, \\ & \|\widehat{H}_{i,:}\|_0 = 1, \\ & \widehat{H} \geq 0. \end{aligned} \tag{5.1}$$

The objective function in (5.1) is continuously differentiable. We know that the feasible set of (5.1) is non-convex with some combinatorial structures. Due to the combinatorial structures, solving (5.1) is generally nondeterministic polynomial-time hard (NP-hard). This model also arises in various fields such as classification [12,44], gene expression studies [6], and text mining [51]. We summarize two key points of (5.1) that need special attention: (i) $\widehat{P}$ may have negative elements; (ii) each row of $\widehat{H}$ must contain exactly one positive element. While optimization with nonnegative orthogonal constraints has been extensively explored [21, 49, 50, 52, 57], few studies consider models that simultaneously satisfy these two points. Taking these into account, we choose to address the problem using the following three methods.

Method One. While the classical K-means model is typically represented using k centroids, it can also be expressed using an indicator matrix. As is well known, the classical K-means model can also be rewritten as (5.1) [4,8]. It is evident that to solve problem (5.1), we only need to solve the K-means problem. Additionally, [40] demonstrated a significant duality between the extraction problem of the EEP and the K-means problem on the rows of Laplacian eigenvectors. Therefore, we can consider using the K-means algorithm to solve problem (5.1). There are many software packages available to solve the K-means model, such as the built-in K-means clustering function in MATLAB[1] and `sklearn.cluster.KMeans` in Python[2]. Here, we choose to use the `sklearn.cluster.KMeans` class from the popular Python machine learning library scikit-learn to solve the K-means model.

Method Two. On the other hand, the orthonormal projective nonnegative matrix factorization (OPNMF) model in [52] takes the following form:

$$\begin{aligned} \min_{\widehat{H}\in\mathbb{R}^{n\times r}} \quad & \left\|\bar{P} - \widehat{H}\widehat{H}^\top \bar{P}\right\|_F^2 \\ \text{s.t.} \quad & \widehat{H}^\top \widehat{H} = I_r, \\ & \widehat{H} \geq 0, \end{aligned} \tag{5.2}$$

where $\bar{P} \in \mathbb{R}_+^{n\times r}$ is given. Eq. (5.2) is equivalent to the classical orthonormal nonnegative matrix factorization (ONMF) model [12] and has been widely studied, such as in [9, 12, 34, 52, 55]. Many approaches have been proposed to solve ONMF or OPNMF [30, 32, 52]. Comparing (5.2) with (5.1), the difference lies in two aspects. One is that the constraints $\|\widehat{H}_{i,:}\|_0 = 1$ and $\widehat{H}(\widehat{H})^\top \mathbf{1}_n = \mathbf{1}_n$ are dropped in (5.2). The other is $\bar{P}$ in (5.2) is nonnegative whereas $\widehat{P}$

[1] Details can be found at `https://ww2.mathworks.cn/help/stats/kmeans.html`

[2] Details can be found at `https://scikit-learn.org/stable/modules/clustering.html#k-means`

in (5.1) may have negative elements. By letting $\bar{P} = \widehat{P} - a\mathbf{1}_{n\times r}$, where a is the minimum value in the matrix $\widehat{P}$, one can guarantee that $\bar{P}$ is nonnegative. Note that, due to the existence of (3.3) and Theorem 4.1, and considering that the eigenvectors corresponding to different eigenvalues of the Laplacian matrix are orthogonal, a must necessarily be a negative value. Furthermore, due to the extraction concept of EEP in Section 3.3, this transformation will not affect the original objective function value. Next, we choose the practical exact penalty method EP4Orth+ proposed in [21] since it performs well and can always return high quality orthogonal nonnegative matrices, as demonstrated in [21]. Additionally, the postprocessing step of EP4Orth+ ensures that the resulting matrix satisfies the constraint $\|\widehat{H}_{i,:}\|_0 = 1$. We refer to [21] for more details of the method.

Method Three. Noting that $\widehat{P}$ in (5.1) may have negative elements, inspired by [52], we choose to rewrite the version without the zero norm constraint of (5.1) as a projective semi-nonnegative matrix factorization (PSNMF) problem. We then solve it using an iterative Lagrangian approach [52], detailed as follows. We first remove the zero norm constraint and $\widehat{H}\widehat{H}^\top \mathbf{1}_n = \mathbf{1}_n$ from (5.1), resulting in the following form:

$$\begin{aligned} \min_{\widehat{H}\in\mathbb{R}^{n\times r}} \quad & \left\|\widehat{P} - \widehat{H}\widehat{H}^\top \widehat{P}\right\|_F^2 \\ \text{s.t.} \quad & \widehat{H}^\top \widehat{H} = I_r, \\ & \widehat{H} \geq 0. \end{aligned} \tag{5.3}$$

Let $\widehat{p}_i := \widehat{P}_{i,\cdot} \in \mathbb{R}^r, i = 1,\dots,n$. As shown in [52, Sections II-E, II-F], let

$$\Phi = \left[\phi^\top(\widehat{p}_1), \phi^\top(\widehat{p}_2), \cdots, \phi^\top(\widehat{p}_n)\right]^\top,$$

where ϕ is a vector function. Denote the kernel matrix [52] $K = \Phi\Phi^\top$ with $K_{ij} = \phi(\widehat{p}_i)\phi^\top(\widehat{p}_j)$, $i,j = 1,\dots,n$. Here we choose the linear kernel function $\phi(\widehat{p}_i) = \widehat{p}_i$ for $i = 1,\dots,n$, i.e. $\Phi = \widehat{P}$. Hence, we can know objective function $\|\widehat{P} - \widehat{H}\widehat{H}^\top \widehat{P}\|_F^2$ in (5.3) is equal to $\|\Phi - \widehat{H}\widehat{H}^\top \Phi\|_F^2$. Because $\widehat{P}$ may not be nonnegative, K is not necessarily nonnegative either. As shown in [52], for a kernel matrix K that has both positive and negative elements, we can separate the positive and negative parts of the matrix by calculating

$$K_{ij}^+ = \frac{|K_{ij}| + K_{ij}}{2}, \quad K_{ij}^- = \frac{|K_{ij}| - K_{ij}}{2}, \quad i,j = 1,\dots,n,$$

where $K = K^+ - K^-$ with $K^+ \geq 0$ and $K^- \geq 0$. Next, we rewrite (5.3) as a projective semi-NMF problem

$$\begin{aligned} \min_{\widehat{H},\widehat{U}\in\mathbb{R}^{n\times r}} \quad & \left\|\Phi - \widehat{H}\widehat{U}^\top \Phi\right\|_F^2 \\ \text{s.t.} \quad & \widehat{U} = \widehat{H}, \\ & \widehat{H}^\top \widehat{H} = I_r, \\ & \widehat{H} \geq 0. \end{aligned} \tag{5.4}$$

Then, following the derivation in [52, Appendix VIII], we obtain its iterative Lagrangian solution

$$\widehat{H}_{ij}^{(k+1)} = \widehat{H}_{ij}^{(k)} \frac{\left(K^+\widehat{H}^{(k)} + \widehat{H}^{(k)}\left(\widehat{H}^{(k)}\right)^\top K^-\widehat{H}^{(k)}\right)_{ij}}{\left(K^-\widehat{H}^{(k)} + \widehat{H}^{(k)}\left(\widehat{H}^{(k)}\right)^\top K^+\widehat{H}^{(k)}\right)_{ij}}. \tag{5.5}$$

It is proved in [52] that the iterative scheme in (5.5) is an iterative Lagrangian solution of (5.4).

However, unfortunately, while this method is suitable for cases where $\widehat{P}$ contains negative elements, it has certain limitations. Specifically, the $\widehat{H}$ obtained through the above method does not satisfy the zero norm constraint $\|\widehat{H}_{i,:}\|_0 = 1$. Generally, the following two situations may arise: (i) there may be more than one nonzero element in a row of $\widehat{H}$; (ii) iteration (5.5) cannot restore zero entries to positive values, which may result in rows in $\widehat{H}$ being all zeros. This also leads to the fact that the iterative (5.5) does not guarantee orthogonality. To address these issues, we implement some corrective measures.

For case (i), inspired by [21], we extract the index with the maximum value in a row. As for case (ii), we need to stop the iteration immediately upon discovering a row in $\widehat{H}$ with all zero elements. Then, we apply the operation for (i) to ensure that $\widehat{H}$ satisfies the zero norm constraint in (5.1). Notably, for the blind extraction problem of EEP, our focus lies in identifying the correct EEP rather than the specific values of $\widehat{H}$. This motivation also drives us to perform the aforementioned operations.

6. Numerical Results

In this section, we show the numerical performance of Algorithm 3.2 to verify the efficiency of our proposed approach. As mentioned in Section 5, we compare the K-means algorithm, the practical exact penalty method and the iterative Lagrangian approach, denoted as K-means, EP4Orth+ and PSNMF, to solve (5.1). The experiments are performed in Windows 10 on an Intel(R) Core(TM) i7-1065G7 CPU at 1.30 GHZ with 16.0 GB of RAM. All codes are written in MATLAB R2022a.

We evaluate the performances of algorithms using synthetic graph signals. The graph sampled from a symmetric hierarchical network model [40] are considered, which guarantee that the sampled graph has an EEP. For more details of this model we refer the reader to [40]. Here, we set the graph to have 378 nodes and an EEP with three same sized classes. Furthermore, the observed graph signal is generated as (3.1). Let $D_{\max}$ be the highest degree of graph G. We consider the two types of low pass graph filters mentioned in Examples 2.1 and 2.2:

(a) $\mathcal{H}_{strong}(L^G) = e^{-\sigma L^G}$ where $\sigma = 10/D_{\max}$.

(b) $\mathcal{H}_{weak}(L^G) = (I + \alpha L^G)^{-1}$ where $\alpha = 0.5/D_{\max}$.

Notice that $\mathcal{H}_{strong}(L^G)$ is strong low pass with $\eta_r \approx 0$, and $\mathcal{H}_{weak}(L^G)$ is weak low pass with $\eta_r \approx 1$.

We first use the cost function F_c and the group accuracy rate γ to measure the performance of BE-EEPs, which is Algorithm 3.2. More specifically, the cost function F_c is given by $F_c = F(\widehat{\mathbb{C}}, \overline{V})$, where $\widehat{\mathbb{C}}$ represents the EEP we found using either K-means, PSNMF or EP4Orth+, and $\overline{V} \in \mathbb{R}^{n\times r}$ is a matrix containing the r structural eigenvectors of the Laplacian matrix of the graph under consideration. The lower the cost function value, the higher the quality of the partition extracted by the algorithm. Note that each class in the planting partition contains 126 vertices, denoted as $|C_i| = 126$, $i = 1, \ldots, r$. Let C^w represent the sum of the absolute difference between the number of vertices in each class of the output partition and $|C_i|$. The group accuracy rate γ is given by

$$\gamma = \frac{|V(G)| - C^w}{|V(G)|}.$$

Hence, the group accuracy rate of outputs over multiple trials is a value greater than or equal to 0 and less than or equal to 1. This value indicates the degree of similarity between the number of vertices in the output partitions and the number of vertices in the corresponding classes of the planting partition. The higher the group accuracy, the closer the number of vertices in the output partitions is to the number of vertices in the original partitions. The two values are shown in Figs. 6.1 and 6.2, with the results being the averages over 200 repetitions with different sample sizes.

From Figs. 6.1 and 6.2, we observe that for all algorithms, the cost function F_c decreases and the group accuracy rate γ increases as the number of samples increases. This indicates that, our method BE-EEPs is effective for the blind extraction of EEPs dealing with in both strong and weak low pass graph signals. Specifically, when extracting from strong low pass graph signals, as shown in the left plot of Fig. 6.1, a decrease that all three algorithms have in the cost function can be seen for increasing sample size. Obviously, EP4Orth+ yields the highest cost function value relative to PSNMF and K-means, while PSNMF and K-means provide nearly identical cost function values. In the right plot of Fig. 6.1, we can clearly observe that although PSNMF achieves the highest group accuracy value compared to K-means and EP4Orth+. When extracting from weak low pass graph signals, as shown in Fig. 6.2, we observe that there is no significant difference in the cost function F_c when comparing the use of K-means, PSNMF and EP4Orth+ for generation. However, it is worth noting that in terms of the group accuracy rate of the generated EEPs, PSNMF slightly outperforms both K-means and EP4Orth+.

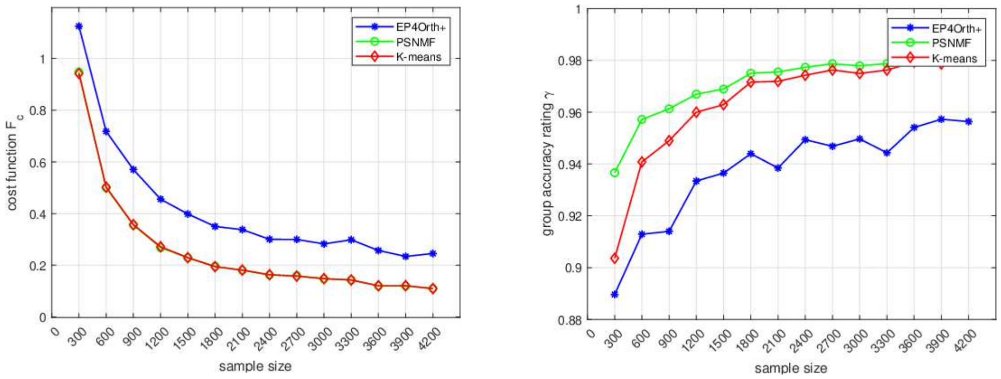


Fig. 6.1. The cost function F_c and the group accuracy rate γ of K-means, PSNMF and EP4Orth+ for strong low pass graph signals.

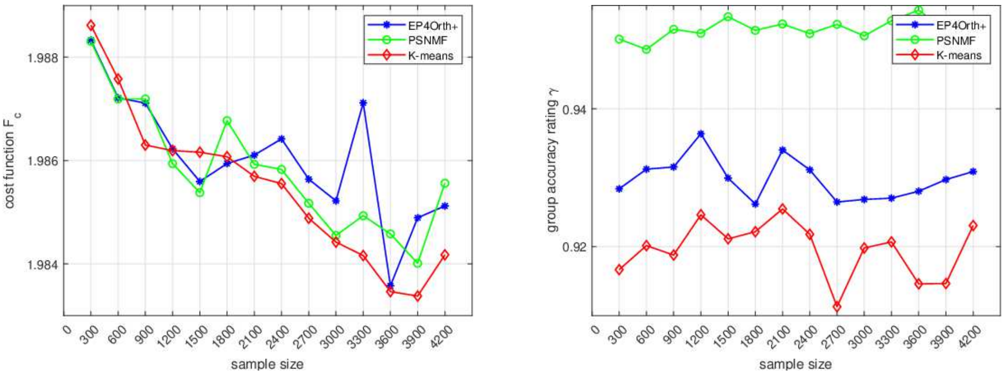
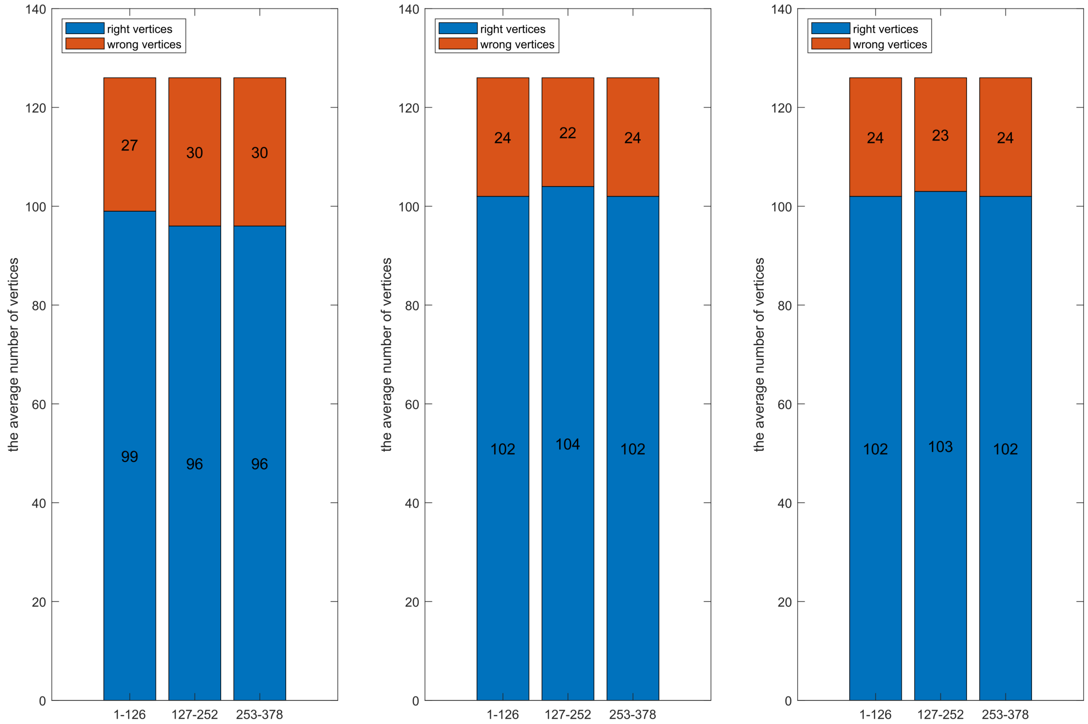


Fig. 6.2. The cost function F_c and the group accuracy rate γ of K-means, PSNMF and EP4Orth+ for weak low pass graph signals.

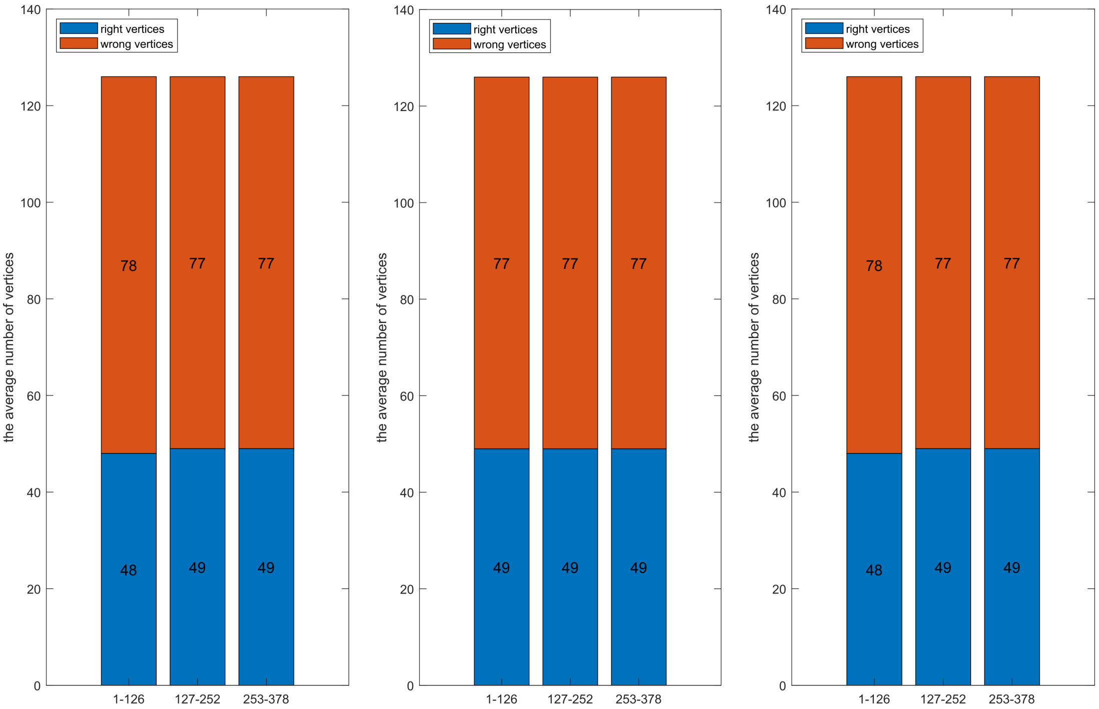
To more accurately evaluate the correctness of the partitions extracted by the algorithm, we present Figs. 6.3 and 6.4, respectively. Note that we know the true partitions, where vertices 1-126 form one partition, vertices 127-252 form another, and vertices 253-378 form a third partition. In Figs. 6.3 and 6.4, each bar represents the number of vertices in a partition. The orange color indicates the number of incorrectly classified vertices, while the blue color represents the number of correctly classified vertices. Figs. 6.3(a), 6.3(b) and 6.3(c) represent the number of correctly and incorrectly classified vertices under strong low pass graph signals by EP4Orth+, PSNMF, and K-means, respectively. Figs. 6.4(a), 6.4(b) and 6.4(c) represent the number of correctly and incorrectly classified vertices under weak low pass graph signals by EP4Orth+, PSNMF, and K-means, respectively. The number of vertices in Figs. 6.3 and 6.4 is the average number of vertices from 181 experiments, using 300 samples as an example.

Fig. 6.3 shows that the number of correctly classified vertices provided by PSNMF and K-means is similar and slightly higher than that of EP4Orth+. Fig. 6.4 shows that when extracting from weak low pass graph signals, the number of correctly classified vertices generated by the three algorithms is very similar. Therefore, although PSNMF slightly outperforms K-means and EP4Orth+ in terms of group accuracy rate for generating EEPs, when extracting from strong low pass graph signals, K-means and PSNMF perform equally well in both the cost function F_c and the number of correctly classified vertices (i.e. low F_c and high number of correctly classified vertices), with both slightly better than EP4Orth+. However, when extracting from weak low pass graph signals, F_c and the number of correctly partitioned vertices generated by all three algorithms are very similar.



(a) The result for EP4Orth+ (b) The result for PSNMF (c) The result for K-means

Fig. 6.3. The facet stacked bar chart of EEP extraction from strong low pass graph signals using three methods.



(a) The result for EP4Orth+ (b) The result for PSNMF (c) The result for K-means

Fig. 6.4. The facet stacked bar chart of EEP extraction from weak low pass graph signals using three methods.

In summary, as mentioned above, our proposed approach BE-EEPs is effective for the blind extraction of EEPs. We find that when extracting EEPs from strong low pass graph signals, PSNMF and K-means perform slightly better than EP4Orth+. However, when extracting EEPs from weak low pass graph signals, all three algorithms are equally effective.

7. Conclusion

In this paper, we addressed the link between low pass graph signals and the extraction of EEPs, alongside addressing the challenge of blind extraction of EEPs under low pass graph filters. To tackle this problem, we proposed a method called BE-EEPs, which reformulated EEP as a matrix optimization model. We introduced the use of K-means, PSNMF and EP4Orth+ to solve this optimization model and offer theoretical support for the proposed method. The experimental results verified that our method for extracting EEPs from low pass graph signals is effective. In terms of three candidate solvers for solving the resulting matrix optimization model, PSNMF and K-means performed equally well under strong low pass graph signals, while EP4Orth+ was slightly less effective. In more complex weak low pass graph signals, all three algorithms performed equally well.

Acknowledgements. We would like to thank the team of Prof. Jiang Bo from Nanjing Normal University for providing the code for the EP4Orth+ algorithm in [21]. Without their code, our comparison using EP4Orth+ would not have been possible.

Qingna Li's research is supported by the NSFC (Grant Nos. 12071032, 12271526).

References

[1] C.O. Aguilar and B. Gharesifard, On almost equitable partitions and network controllability, in: *Proceedings of the 2016 American Control Conference*, IEEE, (2016), 179–184.

[2] A. Anis, A. Gadde, and A. Ortega, Efficient sampling set selection for bandlimited graph signals using graph spectral proxies, *IEEE Trans. Signal Process.*, **64**:14 (2016), 3775–3789.

[3] S. Bonaccorsi, S. Ottaviano, D. Mugnolo, and F.D. Pellegrini, Epidemic outbreaks in networks with equitable or almost-equitable partitions, *SIAM J. Appl. Math.*, **75**:6 (2015), 2421–2443.

[4] C. Boutsidis, P. Drineas, and M.W. Mahoney, Unsupervised feature selection for the k-means clustering problem, in: *Advances in Neural Information Processing Systems*, Curran Associates, Inc., **22** (2009), 153–161.

[5] U. Brandes and T. Erlebach, *Network Analysis: Methodological Foundations*, in: *Lecture Notes in Computer Science*, Vol. 3418, Springer, 2005.

[6] J.-P. Brunet, P. Tamayo, T.R. Golub, and J.P. Mesirov, Metagenes and molecular pattern discovery using matrix factorization, *Proc. Natl. Acad. Sci. USA*, **101**:12 (2004), 4164–4169.

[7] D.M. Cardoso, C. Delorme, and P. Rama, Laplacian eigenvectors and eigenvalues and almost equitable partitions, *Eur. J. Comb.*, **28**:3 (2007), 665–673.

[8] F. Chen, Y. Yang, L. Xu, T. Zhang, and Y. Zhang, Big-data clustering: K-means or K-indicators?, *arXiv:1906.00938*, 2019.

[9] S. Choi, Algorithms for orthogonal nonnegative matrix factorization, in: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, IEEE, (2008), 1828–1832.

[10] K. Devriendt and R. Lambiotte, Nonlinear network dynamics with consensus-dissensus bifurcation, *J. Nonlinear Sci.*, **31**:1 (2021), 18.

[11] K. Devriendt and P. Van Mieghem, Unified mean-field framework for susceptible-infected-susceptible epidemics on networks, based on graph partitioning and the isoperimetric inequality, *Phys. Rev. E*, **96**:5 (2017), 052314.

[12] C. Ding, T. Li, W. Peng, and H. Park, Orthogonal nonnegative matrix t-factorizations for clustering, in: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, (2006), 126–135.

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

[14] M. Egerstedt, Controllability of networked systems, in: *Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems*, (2010), 2161–2164.

[15] M. Egerstedt, S. Martini, M. Cao, K. Camlibel, and A. Bicchi, Interacting with networks: How does structure relate to controllability in single-leader, consensus networks?, *IEEE Control Syst. Mag.*, **32**:4 (2012), 66–73.

[16] P. Ferrer-Cid, J.M. Barcelo-Ordinas, and J. Garcia-Vidal, Volterra graph-based outlier detection for air pollution sensor networks, *IEEE Trans. Network Sci. Eng.*, **9**:4 (2022), 2759–2771.

[17] C. Godsil and G.F. Royle, *Algebraic Graph Theory*, in: *Graduate Texts in Mathematics*, Vol. 207, Springer, 2001.

[18] Y. He and H.T. Wai, Estimating centrality blindly from low-pass filtered graph signals, in: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, (2020), 5330–5334.

[19] Y. He and H.T. Wai, Identifying first-order lowpass graph signals using Perron Frobenius theorem, in: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, (2021), 5285–5289.

[20] Y. He and H.T. Wai, Detecting central nodes from low-rank excited graph signals via structured factor analysis, *IEEE Trans. Signal Process.*, **70** (2022), 2416–2430.

[21] B. Jiang, X. Meng, Z. Wen, and X. Chen, An exact penalty approach for optimization with

nonnegative orthogonality constraints, *Math. Program.*, **198**:1 (2023), 855–897.

[22] X. Liu, Z. Ji, and T. Hou, Graph partitions and the controllability of directed signed networks, *Sci. China-Inf. Sci.*, **62** (2019), 1–11.

[23] B.D. McKay and A. Piperno, Practical graph isomorphism II, *J. Symb. Comput.*, **60** (2014), 94–112.

[24] I. Michos and V. Raptis, Graph partitions in chemistry, *Entropy*, **25**:11 (2023), 1504.

[25] S. Monaco and L.R. Celsi, On multi-consensus and almost equitable graph partitions, *Automatica*, **103** (2019), 53–61.

[26] H.S. Nguyen and H.T. Wai, On detecting low-pass graph signals under partial observations, *arXiv:2405.10001*, 2024.

[27] H. NT and T. Maehara, Revisiting graph neural networks: All we have is low-pass filters, *arXiv:1905.09550*, 2019.

[28] N. O'Clery, Y. Yuan, G.B. Stan, and M. Barahona, Observability and coarse graining of consensus dynamics through the external equitable partition, *Phys. Rev. E*, **88**:4 (2013), 042805.

[29] A. Ortega, P. Frossard, J. Kovačević, J.M. Moura, and P. Vandergheynst, Graph signal processing: Overview, challenges, and applications, *Proc. IEEE*, **106**:5 (2018), 808–828.

[30] J. Pan and M.K. Ng, Orthogonal nonnegative matrix factorization by sparsity and nuclear norm optimization, *SIAM J. Matrix Anal. Appl.*, **39**:2 (2018), 856–875.

[31] J. Pang, G. Cheung, A. Ortega, and O.C. Au, Optimal graph Laplacian regularization for natural image denoising, in: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, (2015), 2294–2298.

[32] F. Pompili, N. Gillis, P.A. Absil, and F. Glineur, Two algorithms for orthogonal nonnegative matrix factorization with application to clustering, *Neurocomputing*, **141** (2014), 15–25.

[33] D.L. Powers and M.M. Sulaiman, The walk partition and colorations of a graph, *Linear Algebra Appl.*, **48** (1982), 145–159.

[34] A. Rahiche and M. Cheriet, Blind decomposition of multispectral document images using orthogonal nonnegative matrix factorization, *IEEE Trans. Image Process.*, **30** (2021), 5997–6012.

[35] R. Ramakrishna, H.T. Wai, and A. Scaglione, A user guide to low-pass graph signal processing and its applications: Tools and applications, *IEEE Signal Process. Mag.*, **37**:6 (2020), 74–85.

[36] C. Ravazzi, R. Tempo, and F. Dabbene, Learning influence structure in sparse social networks, *IEEE Trans. Control Network Syst.*, **5**:4 (2017), 1976–1986.

[37] T.M. Roddenberry and S. Segarra, Blind inference of centrality rankings from graph signals, in: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, (2020), 5335–5339.

[38] A. Salova and R.M. D'Souza, Decoupled synchronized states in networks of linearly coupled limit cycle oscillators, *Phys. Rev. Res.*, **2**:4 (2020), 043261.

[39] A. Sandryhaila and J.M. Moura, Discrete signal processing on graphs: Frequency analysis, *IEEE Trans. Signal Process.*, **62**:12 (2014), 3042–3054.

[40] M.T. Schaub, J. Li, and L. Peel, Hierarchical community structure in networks, *Phys. Rev. E*, **107**:5 (2023), 054305.

[41] M.T. Schaub, N. O'Clery, Y.N. Billeh, J.C. Delvenne, R. Lambiotte, and M. Barahona, Graph partitions and cluster synchronization in networks of oscillators, *Chaos*, **26**:9 (2016), 094821.

[42] M. Scholkemper and M.T. Schaub, Blind extraction of equitable partitions from graph signals, in: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, (2022), 5832–5836.

[43] S. Segarra, A.G. Marques, G. Mateos, and A. Ribeiro, Network topology inference from spectral templates, *IEEE Trans. Signal Inf. Process. Networks*, **3**:3 (2017), 467–483.

[44] F. Sha, L.K. Saul, and D.D. Lee, Multiplicative updates for nonnegative quadratic programming in support vector machines, in: *Advances in Neural Information Processing Systems*, Curran Associates, Inc., **15** (2003), 897–904.

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

[46] D. Thanou, X. Dong, D. Kressner, and P. Frossard, Learning heat diffusion graphs, *IEEE Trans. Signal Inf. Process. Networks*, **3**:3 (2017), 484–499.

[47] H.T. Wai, S. Segarra, A.E. Ozdaglar, A. Scaglione, and A. Jadbabaie, Community detection from low-rank excitations of a graph filter, in: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, (2018), 4044–4048.

[48] H.T. Wai, S. Segarra, A.E. Ozdaglar, A. Scaglione, and A. Jadbabaie, Blind community detection from low-rank excitations of a graph filter, *IEEE Trans. Signal Process.*, **68** (2019), 436–451.

[49] S. Wang, T.H. Chang, Y. Cui, and J.S. Pang, Clustering by orthogonal NMF model and non-convex penalty optimization, *IEEE Trans. Signal Process.*, **69** (2021), 5273–5288.

[50] Z. Wen and W. Yin, A feasible method for optimization with orthogonality constraints, *Math. Program.*, **142**:1 (2013), 397–434.

[51] W. Xu, X. Liu, and Y. Gong, Document clustering based on non-negative matrix factorization, in: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, ACM, (2003), 267–273.

[52] Z. Yang and E. Oja, Linear and nonlinear projective nonnegative matrix factorization, *IEEE Trans. Neural Networks*, **21**:5 (2010), 734–749.

[53] S. Yu and J. Shi, Multiclass spectral clustering, in: *Proceedings Ninth IEEE International Conference on Computer Vision*, IEEE, (2003), 313–319.

[54] Y. Yuan, G.B. Stan, L. Shi, M. Barahona, and J. Goncalves, Decentralised minimum-time consensus, *Automatica*, **49**:5 (2013), 1227–1235.

[55] Z. Yuan, Z. Yang, and E. Oja, Projective nonnegative matrix factorization: Sparseness, orthogonality, and clustering, 2009. `https://users.ics.aalto.fi/zyuan/dissertation/article-7.pdf`.

[56] C. Zhang, Y. He, and H.T. Wai, Detecting low pass graph signals via spectral pattern: Sampling complexity and applications, *arXiv:2306.01553*, 2023.

[57] K. Zhang, S. Kai, J. Liu, J. Wang, and J. Zhang, Greedy orthogonal pivoting algorithm for non-negative matrix factorization, in: *International Conference on Machine Learning*, PMLR, (2019), 7493–7501.