Decentralized Graph Based Filter Design Using Normalized Adjacency Matrix

Yufan Fan and Marius Pesavento

Communications Systems Group
Technische Universität Darmstadt

Graph Signal Processing Preliminaries

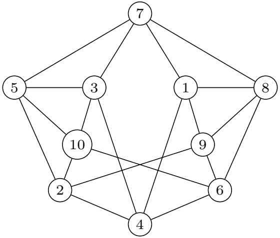


Figure: An undirected network $\mathcal{G}$ with $N = 10$ nodes.

- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, N\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$

  Adjacency matrix $\boldsymbol{A}$, where

  $$a_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

  Degree matrix $\boldsymbol{D}$, where

  $$d_{ij} = \begin{cases} \sum_{j=1}^{N} a_{ij} & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Graph Signal Processing Preliminaries

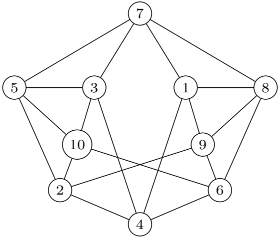


Figure: An undirected network $\mathcal{G}$ with $N = 10$ nodes.

Incidence matrix $\boldsymbol{B}$, where

$$b_{k\ell} = \begin{cases} 1 & \text{if } k = i, \ell = j \\ -1 & \text{if } k = j, \ell = i, \\ 0 & \text{otherwise} \end{cases}$$

Combinatorial Graph Laplacian matrix $\boldsymbol{L}$, where

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} = \boldsymbol{B}\boldsymbol{B}^{\mathsf{T}}$$

Graph Filter

- A linear shift-invariant finite impulse response (FIR) graph filter (GF) $\hat{\boldsymbol{H}}$ can be expressed as

$$\hat{\boldsymbol{H}} = h_0 \boldsymbol{S}^0 + h_1 \boldsymbol{S}^1 + \cdots + h_M \boldsymbol{S}^M = \sum_{m=0}^{M} h_m \boldsymbol{S}^M, \tag{1}$$

  where $h_m \in \mathbb{R}, m = 0, 1, \ldots, M$, denotes polynomial coefficients of the GF with filter order M, and $\boldsymbol{S}$ is the graph shift operator (GSO).
- The output $\boldsymbol{y}$ of the GF with respect to the graph signal $\boldsymbol{x}$ is $\boldsymbol{y} = \hat{\boldsymbol{H}}\boldsymbol{x}$.

Graph Filter

TECHNISCHE UNIVERSITÄT DARMSTADT

- The eigendecomposition of the GSO: $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathsf{T}}$.

  Graph frequencies are *known*
  $\rightarrow$ *graph dependent GF*
  - customized design
  - high computation cost

  Graph frequencies are *unknown*
  $\rightarrow$ *graph independent GF*
  - universal design
  - low computation cost

- Our goal: Design a graph dependent GF by finding graph frequencies in a fully decentralized manner.

Graph Shift Operator

TECHNISCHE UNIVERSITÄT DARMSTADT

- Different choices of the GSO $\boldsymbol{S}$ for undirected graphs[1]

GSO $\boldsymbol{S}$	Expression	Properties
Adjacency Matrix	$\boldsymbol{A}$	Symmetric, $\lambda_i \leq d_{\max}$
Normalized Adjacency	$\boldsymbol{A}_{\text{n}} = \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{A} \boldsymbol{D}^{-\frac{1}{2}}$	Symmetric, $\lambda_i \in [-1, 1]$
Combinatorial Laplacian	$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$	Symmetric, $\lambda_i \geq 0$
Normalized Laplacian	$\boldsymbol{L}_{\text{n}} = \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{L} \boldsymbol{D}^{-\frac{1}{2}}$	Symmetric, $\lambda_i \in [0, 2]$
Random Walk	$\boldsymbol{L}_{\text{rw}} = \boldsymbol{I} - \boldsymbol{D}^{-1} \boldsymbol{A}$	Asymmetric, $\lambda_i \geq 0$

λ_i denotes the ith largest eigenvalue.

[1] For directed graphs, cf. A. Anis, A. Gadde, and A. Ortega, “Efficient Sampling Set Selection for Bandlimited Graph Signals Using Graph Spectral Proxies”.

Normalized Adjacency Matrix

- The mth power of the GSO can be expressed as

$$\boldsymbol{S}^m = \boldsymbol{U}\boldsymbol{\Lambda}^m\boldsymbol{U}^\mathsf{T}. \tag{2}$$

- If $|\lambda_i| > 1 \rightarrow$
  - amplification of intermediate graph signals
  - potential round-off errors
- We propose to use the normalized adjacency matrix $\boldsymbol{A}_\mathsf{n}$ as the GSO, whose eigenvalue can be found by

$$\lambda_i(\boldsymbol{A}_\mathsf{n}) = 1 - \lambda_{N+1-i}(\boldsymbol{L}_\mathsf{n}) \tag{3}$$

- Our goal: Estimate the eigenvalues of $\boldsymbol{L}_\mathsf{n}$ in fully decentralized manner.

Decentralized Graph Frequency Estimation

- Notice that the degree matrix $\boldsymbol{D}$ is diagonal, and $\boldsymbol{L} = \boldsymbol{B}\boldsymbol{B}^{\mathsf{T}}$, we have

$$\boldsymbol{L}_{\mathsf{n}} = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{B}(\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{B})^{\mathsf{T}}. \tag{4}$$

- Denote the columns of the matrix $\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{B}$ as $\boldsymbol{x}_i, i = 1, \ldots, N_e$, i.e.,

$$\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{B} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_e}], \tag{5}$$

  where $N_e = |\mathcal{V}|$.
- Entries of $\boldsymbol{x}_i$ are available at corresponding nodes $\rightarrow$ $\boldsymbol{x}_i$ is the ith graph signal.
- The matrix $\boldsymbol{L}_{\mathsf{n}}$ can be rewritten as a summation of rank-one modifications

$$\boldsymbol{L}_{\mathsf{n}} = \sum_{i=1}^{N_e} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathsf{T}}. \tag{6}$$

Online Decentralized Eigendecomposition

- The eigenvalues of $\boldsymbol{L}_{\mathrm{n}}$ can be iteratively estimated by examining all column vectors $\boldsymbol{x}_i$.
- $\boldsymbol{L}_n$ can be expressed as

$$\boldsymbol{L}_n(i) = \boldsymbol{L}_n(i-1) + \boldsymbol{x}_i \boldsymbol{x}_i^{\mathsf{T}}, \quad i = 1, 2, \ldots, N_e. \tag{7}$$

  where $\boldsymbol{L}_n(0) = \mathbf{0}$, and $\boldsymbol{L}_n(N_e) = \boldsymbol{L}_n$.
- Online Decentralized Eigendecomposition Algorithm.

1. Y. Fan, M. Trinh-Hoang, and M. Pesavento, “Decentralized Eigendecomposition for Online Learning over Graphs,” in 2021 29th European Signal Processing Conference (EUSIPCO), Aug. 2021, pp. 1825–1829.
2. Y. Fan, M. Trinh-Hoang, C. E. Ardic, and M. Pesavento, “Decentralized Eigendecomposition for Online Learning over Graphs with Applications,” no. arXiv:2209.01257, Sep. 2022.

Online Decentralized Eigendecomposition

TECHNISCHE UNIVERSITÄT DARMSTADT

- Assumption: the eigenvalue $\boldsymbol{\Lambda}(i-1)$ and corresponding eigenvector matrix $\boldsymbol{U}(i-1)$ of $\boldsymbol{L}_n(i-1)$ are known.
- Known at the jth node: $\boldsymbol{\Lambda}(i-1)$, jth row of $\boldsymbol{U}(i-1)$, and the jth entry of $\boldsymbol{x}_i$.
- We have

$$\boldsymbol{U}(i-1)^{\mathsf{T}}\boldsymbol{L}_n(i)\boldsymbol{U}(i-1) = \boldsymbol{\Lambda}(i-1) + \boldsymbol{z}(i)\boldsymbol{z}(i)^{\mathsf{T}}, \tag{8}$$

  where

$$\boldsymbol{z}(i) = [z_1(i), \ldots, z_N(i)]^{\mathsf{T}} = \boldsymbol{U}(i-1)^{\mathsf{T}}\boldsymbol{x}(i). \tag{9}$$

- Equation (9) can be realized by *network gossiping protocols*, e.g., the average consensus algorithm, the Push-Sum algorithm, the finite-time average consensus algorithm.
- Equation (8) can be realized by efficient implementation of the *rational function approximation* in each node locally.

Simulation Result

TECHNISCHE UNIVERSITÄT DARMSTADT

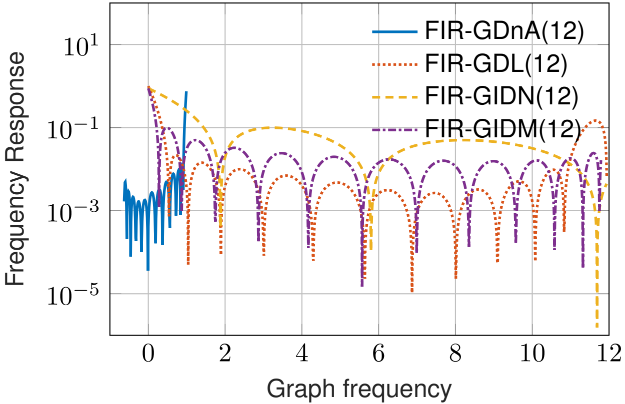
- A random Erdős-Rényi graph with $N = 80$ nodes.
- Push-Sum iterations 80.
- We design an ideal graph based low pass FIR GF using $\boldsymbol{A}_{\mathrm{n}}$, denoted as FIR-GDnA, to compute the average value of a random graph signal, i.e., $\bar{\boldsymbol{x}} = \frac{\mathbf{1}^{\mathsf{T}}\boldsymbol{x}}{N}$.
- Filter order $M = 12$.
- Relative error

$$\eta = \frac{\|\boldsymbol{y} - \bar{\boldsymbol{x}}\|_2^2}{\|\bar{\boldsymbol{x}}\|_2^2}. \tag{10}$$

- Comparisons:
  - FIR-GDL: graph based FIR using the combinatorial graph Laplacian matrix $\boldsymbol{L}$
  - FIR-GIDN: graph independent FIR based on the network size
  - FIR-GIDM: graph independent FIR based on the maximum eigenvalue of $\boldsymbol{L}$.

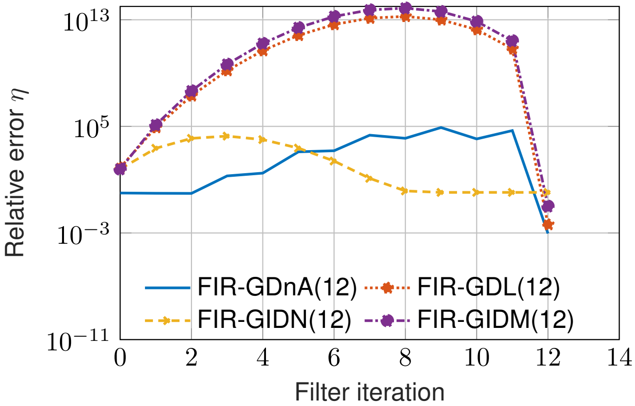
Frequency Responses

- Frequency responses of graph dependent and graph independent FIR graph filters with order $M = 12$.



Relative Error Performance

- Relative error performance of graph dependent and graph independent FIR graph filters with order $M = 12$.



TECHNISCHE UNIVERSITÄT DARMSTADT

Thank you very much!