Distributed Edge-Variant Graph Filters

Mario Coutino, Elvin Isufi, Geert Leus

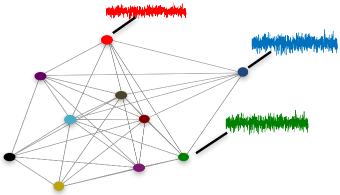
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology, Delft, The Netherlands

CAMSAP 2017
Curaçao, Dutch Antilles

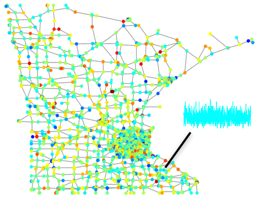
[m.a.coutinominguez@tudelft.nl]

TU Delft Delft University of Technology

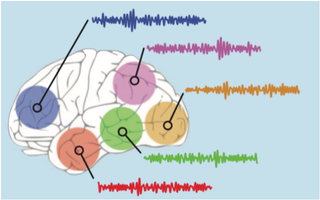
Signal Processing over Graphs



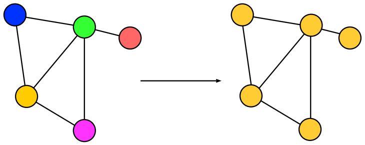
Sensor networks (e.g. pollution)



Transport networks (number of vehicles crossing the junction)



Brain networks (fMRI time series)

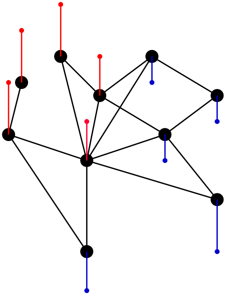


Graph signal consensus

Preliminaries

- Undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
- $|\mathcal{V}| = N$ nodes, $|E| = M$ edges
- Graph signal $x \in \mathbb{R}^N$
- Symmetric shift matrix
  $\boldsymbol{S} \in \mathbb{R}^{N \times N}$
  Adjacency ($\boldsymbol{A}$), Laplacian ($\boldsymbol{L}$)
- Graph Fourier transform
  Eigenvalue decomposition
  $\boldsymbol{S} = \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U}^T$; $\boldsymbol{U} = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_n]$; $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_1, \cdots, \lambda_n\}$
- Transform and inverse transform

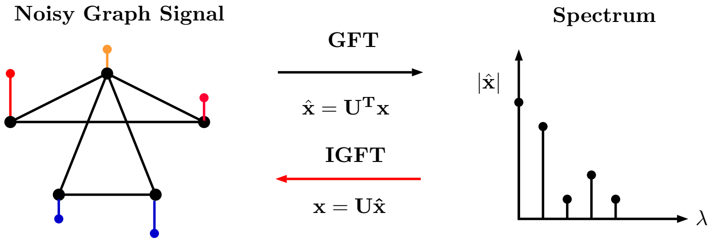
$$\hat{\boldsymbol{x}} = \boldsymbol{U}^T \boldsymbol{x}; \quad \boldsymbol{x} = \boldsymbol{U} \hat{\boldsymbol{x}}$$



[Sandryhaila 2014, SPM] [Shuman 2013, SPM]

TUDelft Delft University of Technology

- **Graph Fourier Transform** (Illustration)



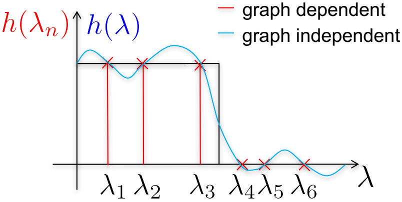
Distributed Graph Filters

- **Shift-Invariant Graph Filter**

$$\hat{y} = h(\lambda_n)\hat{x}_n$$

$$\hat{\boldsymbol{y}} = h(\boldsymbol{\Lambda})\hat{\boldsymbol{x}}$$

$$h(\boldsymbol{\Lambda}) = \mathrm{diag}\{h(\lambda_1), \cdots, h(\lambda_n)\}$$



- **In the vertex domain**

$$\boldsymbol{y} = \boldsymbol{U}h(\boldsymbol{\Lambda})\boldsymbol{U}^T\boldsymbol{x} = \boldsymbol{H}\boldsymbol{x}$$

- for $\boldsymbol{H}$ polynomial of $\boldsymbol{S}$

$$\boldsymbol{H} = \sum_{k=0}^{K} \phi_k \boldsymbol{S}^K; \qquad h(\lambda_n) = \sum_{k=0}^{K} \phi_k \lambda_n^k \text{ (Filter Freq. Response)}$$

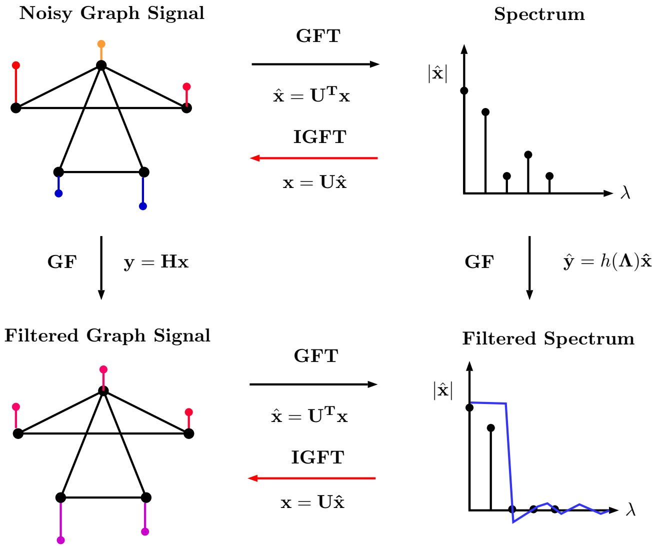
- **Distributed Implementation** ($\boldsymbol{S}$ is "local") [Shuman 2011, DCOSS, arxiv]

$$\boldsymbol{x}^k \triangleq \boldsymbol{S}^k\boldsymbol{x} = \boldsymbol{S}(\boldsymbol{S}^{k-1}\boldsymbol{x}) = \boldsymbol{S}\boldsymbol{x}^{k-1}$$

[Sandryhaila 2013, TSP]

TU Delft Delft University of Technology

Distributed Graph Filters

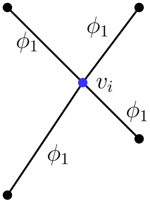


Distributed Graph Filters

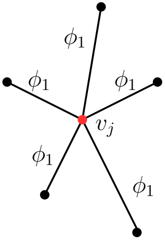
- **Classical FIR Graph Filter** (C-GF) [Sandryhaila 2013, TSP]

$$\boldsymbol{H}_{\mathrm{c}} \triangleq \sum_{k=0}^{K} \phi_k \boldsymbol{S}^k; \qquad \boldsymbol{y}_{\mathrm{c}} = \boldsymbol{H}_{\mathrm{c}} \boldsymbol{x}$$

One-hop neighborhood of v_i



One-hop neighborhood of v_j

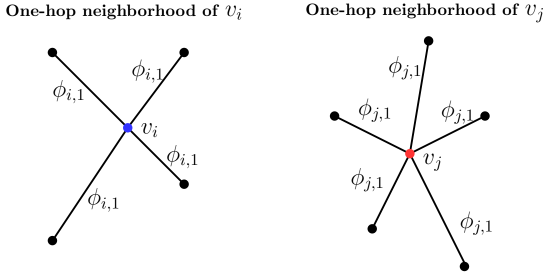


- distributable
- scalar coefficients $\{\phi_k\}$
- $\mathcal{O}(MK)$ communications
- $\boldsymbol{H}_{\mathrm{nv}} \boldsymbol{S} = \boldsymbol{S} \boldsymbol{H}_{\mathrm{nv}}$ (always) ☹

Distributed Graph Filters

- **Node-variant Graph Filter** (NV-GF) [Segarra 2017, TSP]

$$\boldsymbol{H}_{\mathrm{nv}} \triangleq \sum_{k=0}^{K} \mathrm{diag}(\boldsymbol{\phi}_k)\boldsymbol{S}^k; \qquad \boldsymbol{y}_{\mathrm{nv}} = \boldsymbol{H}_{\mathrm{nv}}\boldsymbol{x}$$



- distributable
- vectors of coeffs. $\{\boldsymbol{\phi}_k\}$
- $\mathcal{O}(MK)$ communications
- $\boldsymbol{H}_{\mathrm{nv}}\boldsymbol{S} \neq \boldsymbol{S}\boldsymbol{H}_{\mathrm{nv}}$ (in general)

TU Delft Delft University of Technology

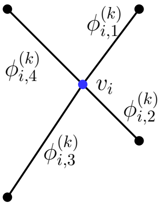
Edge-Variant Graph Filters

- **Edge-variant Graph Filter** (EV-GF)

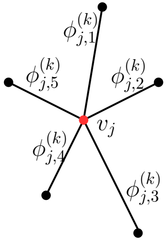
$$\boldsymbol{H}_{\text{ev}} \triangleq \boldsymbol{\Phi}_0 + \boldsymbol{\Phi}_1 \odot \boldsymbol{S} + (\boldsymbol{\Phi}_2 \odot \boldsymbol{S})(\boldsymbol{\Phi}_1 \odot \boldsymbol{S}) + \dots + (\boldsymbol{\Phi}_K \odot \boldsymbol{S})(\boldsymbol{\Phi}_{K-1} \odot \boldsymbol{S}) \cdots (\boldsymbol{\Phi}_1 \odot \boldsymbol{S})$$

$$= \sum_{k=1}^{K} \prod_{j=1}^{k} (\boldsymbol{\Phi}_j \odot \boldsymbol{S}) + \boldsymbol{\Phi}_0, \quad \text{with} \odot : \text{Hadamard product}$$

One-hop neighborhood of v_i



One-hop neighborhood of v_j



- distributable $\Longrightarrow$ $\boldsymbol{x}^{(k)} = (\boldsymbol{\Phi}_k \odot \boldsymbol{S})\boldsymbol{x}^{(k-1)}$
- $\{\boldsymbol{\Phi}_j\}$ coefficients matrices
- $\mathcal{O}(MK)$ communications
- $\boldsymbol{H}_{\text{nv}}\boldsymbol{S} \neq \boldsymbol{S}\boldsymbol{H}_{\text{nv}}$ (in general)

specializes to C-GF and NV-GF☺

Graph Filter Design

- **Filter Response Fitting**

$$\underset{\Theta}{\text{minimize}} \left\| \tilde{\boldsymbol{H}} - \boldsymbol{H}_{\text{fit}}(\Theta) \right\|,$$

where

- $\tilde{\boldsymbol{H}}$ desired filter response.
- $\| \cdot \|$ appropriate norm, e.g., Frobenius, spectral, etc.

and

$$\boldsymbol{H}_{\text{fit}}(\Theta) = \begin{cases} \boldsymbol{H}_{\text{c}}; & \Theta = \{\phi_k\} \\ \boldsymbol{H}_{\text{nv}}; & \Theta = \{\boldsymbol{\phi}_k\} \\ \boldsymbol{H}_{\text{ev}}; & \Theta = \{\boldsymbol{\Phi}_k\} \end{cases}$$

Graph Filter Design

- **Edge-variant Graph Filter** (EV-GF)

  Find $\{\boldsymbol{\Phi}_j\}_{j=1}^{K}$ by solving

$$\underset{\{\boldsymbol{\Phi}_j\}}{\text{minimize}} \quad \left\| \tilde{\boldsymbol{H}} - \sum_{k=1}^{K} \prod_{j=1}^{k} (\boldsymbol{\Phi}_j \odot \boldsymbol{S}) + \boldsymbol{\Phi}_0 \right\|,$$

High-dimensional nonconvex problem ☹

$\Longrightarrow$ suboptimal results due to its multiple local minima.

Two-step design

- decomposition of $\tilde{\boldsymbol{H}}$ by a finite matrix series
- sequential fitting of weighting matrices

Constrained Edge-Variant Graph Filter

Recall:

$$\boldsymbol{H}_{\mathrm{ev}} = \sum_{k=1}^{K} \prod_{j=1}^{k} (\boldsymbol{\Phi}_j \odot \boldsymbol{S}) + \boldsymbol{\Phi}_0$$

weighting the "one hop" neighbors, we can consider a different edge-variant graph filter as follows

- **Constrained Edge-Variant Graph Filters** (C-EV)

$$\boldsymbol{H}_{\mathrm{c-ev}} \triangleq \sum_{k=1}^{K} (\boldsymbol{\Phi}_k \odot \boldsymbol{S}) \boldsymbol{S}^{k-1} + \boldsymbol{\Phi}_0.$$

Constrained Edge-Variant Graph Filter

- **Constrained Edge-Variant Graph Filters** (C-EV)

$$\boldsymbol{H}_{\mathrm{c-ev}} \triangleq \sum_{k=1}^{K} (\boldsymbol{\Phi}_k \odot \boldsymbol{S})\boldsymbol{S}^{k-1} + \boldsymbol{\Phi}_0,$$

- distributable $\begin{cases} \boldsymbol{x}^{(k)} = \boldsymbol{S}\boldsymbol{x}^{(k-1)} & \text{(regular filter)} \\ \boldsymbol{y}^{(k)} = (\boldsymbol{\Phi}_k \odot \boldsymbol{S})\boldsymbol{x}^{(k-1)} & \text{(weighted filter)} \end{cases}$
- $\mathcal{O}(MK)$ communication complexity
- $\boldsymbol{H}_{\mathrm{nv}}\boldsymbol{S} \neq \boldsymbol{S}\boldsymbol{H}_{\mathrm{nv}}$ (in general)
- **specializes C-GF and NV-GF** ☺
- **accepts easy design** for $\{\boldsymbol{\Phi}_k\}$☺

- **Constrained Edge-Variant Graph Filters** (CEV-GF)

  Element-wise Fitting (Frobenius norm)

  Optimal weights from linear system

$$\underset{\{\phi_k\}}{\text{minimize}} \quad \left\| \tilde{\boldsymbol{h}} - \sum_{k=1}^{K} (\boldsymbol{S}^{k-1} \otimes \boldsymbol{I}) \text{diag}(\boldsymbol{s}) \phi_k + \phi_0 \right\|_2 ,$$

  where $\otimes$: Kronecker product and

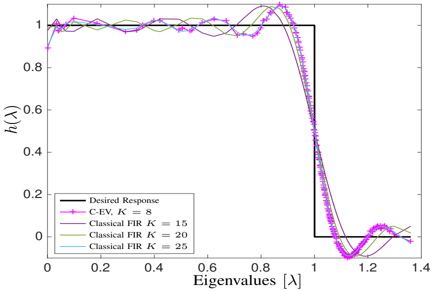
$$\boldsymbol{s} = \text{vec}(\boldsymbol{S}); \qquad \phi_k = \text{vec}(\boldsymbol{\Phi}_k); \qquad \tilde{\boldsymbol{h}} = \text{vec}(\tilde{\boldsymbol{H}})$$

$\Longrightarrow$ **solution through least squares** ☺

Numerical Results (1/3)

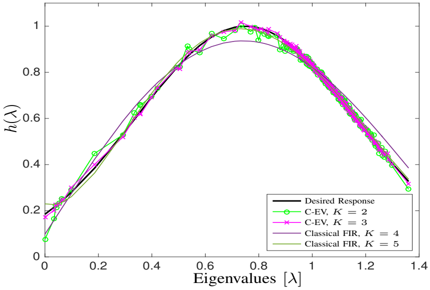
Ideal Low-pass Filter

$$\tilde{h}(\lambda) = \begin{cases} 1 & 0 \leq \lambda \leq \lambda_c \\ 0 & \text{otherwise}, \end{cases}$$



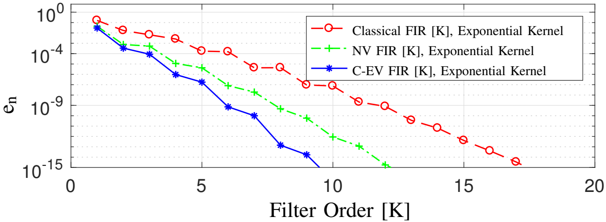
Exponential Kernel

$$\tilde{h}(\lambda) = e^{-\gamma(\lambda-\mu)^2},$$

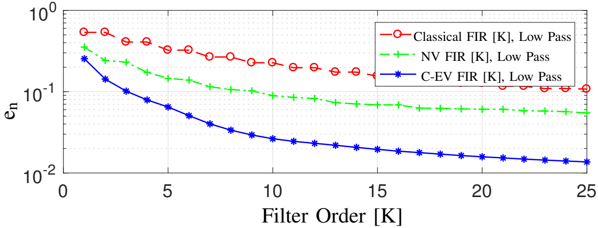


$$\{\mu = 0.75; \quad \gamma = 3\}$$

TU Delft Delft University of Technology



Exponential Kernel



Ideal Low-pass Filter

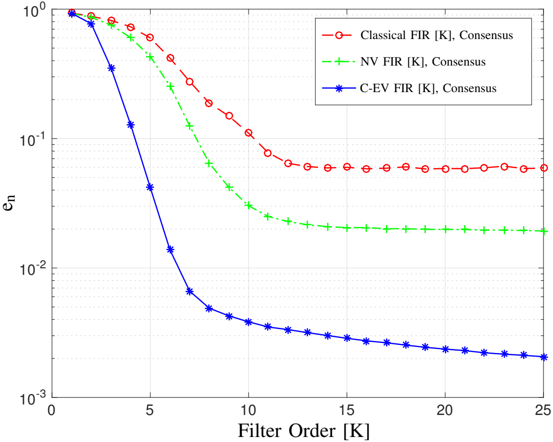
$$e_n = \|\tilde{\boldsymbol{H}} - \boldsymbol{H}_{\text{fit}}\|_F^2 / \|\tilde{\boldsymbol{H}}\|_F^2$$

TUDelft Delft University of Technology

Numerical Results (3/3)

- **Consensus** (No Eigendecomposition)

$$\tilde{\boldsymbol{H}} = \frac{1}{N}\mathbf{1}\mathbf{1}^T; \text{ Community Graph}; \quad N = 256$$



Conclusions

- Edge-variant graph filter reduces the *communication* and *computational* complexity.
- Information from neighbors is *weighted* (possibly) asymmetrically.
- $\tilde{\boldsymbol{H}}$ outside the subspace of $\boldsymbol{S}$ are possible to approximate ☺
- C-EV still generalizes C and NV graph filters ☺
- Least squares design for C-EV ☺
  - provides same approximation accuracy at reduced complexity ☺
- *Consensus*:

  C-EV saturates at much lower error with reduced order ☺

TUDelft Delft University of Technology

Thank you!

Questions?

[m.a.coutinominguez@tudelft.nl]