Network Topology Inference from Non-stationary Graph Signals

Rasoul Shafipour

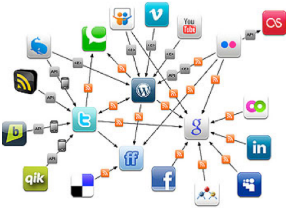
Dept. of Electrical and Computer Engineering
University of Rochester
rshafipo@ece.rochester.edu
http://www.ece.rochester.edu/~rshafipo/

Co-authors: Santiago Segarra, Antonio G. Marques and Gonzalo Mateos

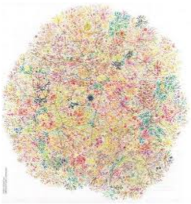
ICASSP, March 6, 2017

Network Science analytics

Online social media



Internet

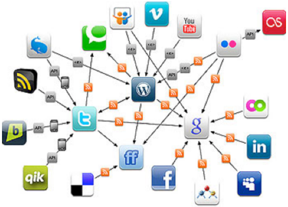


Clean energy and grid analytics

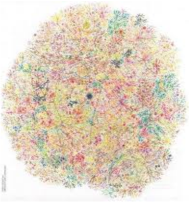
- Network as graph $G = (\mathcal{V}, \mathcal{E})$: encode pairwise relationships
- Desiderata: Process, analyze and learn from network data [Kolaczyk'09]

Network Science analytics

Online social media



Internet



Clean energy and grid analytics

- Network as graph $G = (\mathcal{V}, \mathcal{E})$: encode pairwise relationships
- Desiderata: Process, analyze and learn from network data [Kolaczyk'09]
- Interest here not in G itself, but in data associated with nodes in $\mathcal{V}$
  - $\Rightarrow$ The object of study is a graph signal
- Ex: Opinion profile, buffer congestion levels, neural activity, epidemic

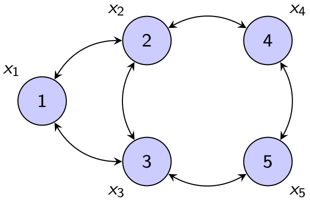
Graph signal processing (GSP)

- Undirected G with adjacency matrix $\mathbf{A}$

  $\Rightarrow A_{ij}$ = Proximity between i and j

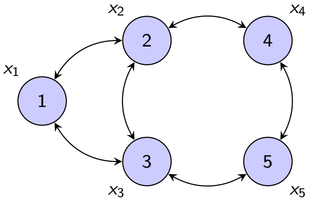
- Define a signal $\mathbf{x}$ on top of the graph

  $\Rightarrow x_i$ = Signal value at node i



Graph signal processing (GSP)

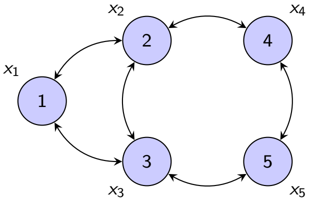
- Undirected G with adjacency matrix $\mathbf{A}$
  - $\Rightarrow A_{ij}$ = Proximity between i and j
- Define a signal $\mathbf{x}$ on top of the graph
  - $\Rightarrow x_i$ = Signal value at node i



- Associated with G is the graph-shift operator $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T \in \mathcal{M}^N$
  - $\Rightarrow S_{ij} = 0$ for $i \neq j$ and $(i,j) \notin \mathcal{E}$ (local structure in G)
  - $\Rightarrow$ Ex: $\mathbf{A}$, degree $\mathbf{D}$ and Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ matrices

Graph signal processing (GSP)

- Undirected G with adjacency matrix $\mathbf{A}$
  - $\Rightarrow A_{ij}$ = Proximity between i and j
- Define a signal $\mathbf{x}$ on top of the graph
  - $\Rightarrow x_i$ = Signal value at node i



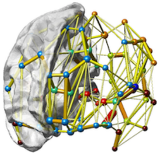
- Associated with G is the graph-shift operator $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T \in \mathcal{M}^N$
  - $\Rightarrow S_{ij} = 0$ for $i \neq j$ and $(i, j) \notin \mathcal{E}$ (local structure in G)
  - $\Rightarrow$ Ex: $\mathbf{A}$, degree $\mathbf{D}$ and Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ matrices
- Graph Signal Processing $\rightarrow$ Exploit structure encoded in $\mathbf{S}$ to process $\mathbf{x}$
  - $\Rightarrow$ Our view: GSP well suited to study (network) diffusion processes
- Take the reverse path. How to use GSP to infer the graph topology?

Topology inference: Motivation and context

- Network topology inference from nodal observations [Kolaczyk'09]
  - Partial correlations and conditional dependence [Dempster'74]
  - Sparsity [Friedman et al'07] and consistency [Meinshausen-Buhlmann'06]

- Key in neuroscience [Sporns'10]

  $\Rightarrow$ Functional net inferred from activity

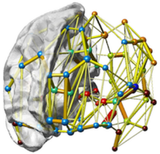


Topology inference: Motivation and context

- Network topology inference from nodal observations [Kolaczyk'09]
  - Partial correlations and conditional dependence [Dempster'74]
  - Sparsity [Friedman et al'07] and consistency [Meinshausen-Buhlmann'06]

- Key in neuroscience [Sporns'10]

  $\Rightarrow$ Functional net inferred from activity



- Noteworthy GSP-based approaches
  - Gaussian graphical models [Egilmez et al'16]
  - Smooth signals [Dong et al'15], [Kalofolias'16]
  - Stationary signals [Pasdeloup et al'15], [Segarra et al'16]
  - Directed graphs [Mei-Moura'15], [Shen et al'16]

- **Our contribution:** topology inference from non-stationary graph signals

Generating structure of a diffusion process

- Signal $\mathbf{y}$ is the response of a linear diffusion process to an input $\mathbf{x}$

$$\mathbf{y} = \alpha_0 \prod_{l=1}^{\infty} (\mathbf{I} - \alpha_l \mathbf{S}) \mathbf{x} = \sum_{l=0}^{\infty} \beta_l \mathbf{S}^l \mathbf{x}$$

$\Rightarrow$ Common generative model. Heat diffusion if α_k constant

- We say the graph shift $\mathbf{S}$ explains the structure of signal $\mathbf{y}$

Generating structure of a diffusion process

- Signal **y** is the response of a linear diffusion process to an input **x**

$$\mathbf{y} \;=\; \alpha_0 \prod_{l=1}^{\infty}(\mathbf{I} - \alpha_l \mathbf{S})\mathbf{x} \;=\; \sum_{l=0}^{\infty} \beta_l \mathbf{S}^l \mathbf{x}$$

$\Rightarrow$ Common generative model. Heat diffusion if α_k constant

- We say the graph shift **S** explains the structure of signal **y**

- Cayley-Hamilton asserts we can write diffusion as

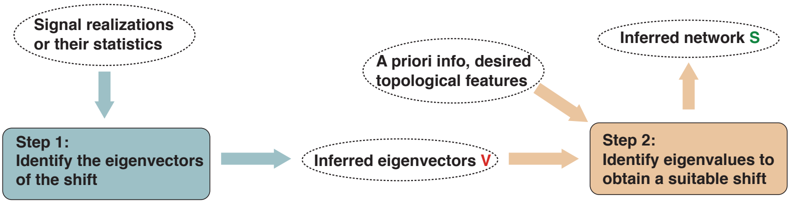
$$\mathbf{y} = \left(\sum_{l=0}^{N-1} h_l \mathbf{S}^l\right)\mathbf{x} := \mathbf{H}\mathbf{x}$$

$\Rightarrow$ Graph filter **H** is shift invariant [Sandryhaila-Moura'13]

$\Rightarrow$ **H** diagonalized by the eigenvectors **V** of the shift operator

Our approach for topology inference

- Two-step approach for graph topology identification



- Beyond diffusion $\rightarrow$ Alternative sources for spectral templates $\mathbf{V}$
  - Design of graph filters [Segarra et al'15]
  - Graph sparsification and network deconvolution [Feizi et al'13]

Step 2: Obtaining the eigenvalues

- We can use extra knowledge/assumptions to choose one graph

  $\Rightarrow$ Of all graphs, select one that is optimal in some sense

$$\mathbf{S}^* := \operatorname*{argmin}_{\mathbf{S},\boldsymbol{\lambda}} \; f(\mathbf{S}, \boldsymbol{\lambda}) \quad \text{s. to} \quad \mathbf{S} = \sum_{k=1}^{N} \lambda_k \mathbf{v}_k \mathbf{v}_k^T, \;\; \mathbf{S} \in \mathcal{S}$$

- Set $\mathcal{S}$ contains all admissible scaled adjacency matrices

$$\mathcal{S} := \{\mathbf{S} \,|\, S_{ij} \geq 0, \;\; \mathbf{S} \in \mathcal{M}^N, \; S_{ii} = 0, \;\; \textstyle\sum_j S_{1j} = 1\}$$

  $\Rightarrow$ Can accommodate Laplacian matrices as well

Step 2: Obtaining the eigenvalues

- We can use extra knowledge/assumptions to choose one graph
  - $\Rightarrow$ Of all graphs, select one that is optimal in some sense

$$\mathbf{S}^* := \underset{\mathbf{S},\boldsymbol{\lambda}}{\operatorname{argmin}} \ f(\mathbf{S}, \boldsymbol{\lambda}) \quad \text{s. to} \quad \mathbf{S} = \sum_{k=1}^{N} \lambda_k \mathbf{v}_k \mathbf{v}_k^T, \ \ \mathbf{S} \in \mathcal{S}$$

- Set $\mathcal{S}$ contains all admissible scaled adjacency matrices

$$\mathcal{S} := \{\mathbf{S} \mid S_{ij} \geq 0, \ \ \mathbf{S} \in \mathcal{M}^N, \ S_{ii} = 0, \ \ \textstyle\sum_j S_{1j} = 1\}$$

  - $\Rightarrow$ Can accommodate Laplacian matrices as well

- Problem is convex if we select a convex objective $f(\mathbf{S}, \boldsymbol{\lambda})$

  Ex: Sparsity ($f(\mathbf{S}) = \|\mathbf{S}\|_1$), min. energy ($f(\mathbf{S}) = \|\mathbf{S}\|_F$), mixing ($f(\boldsymbol{\lambda}) = -\lambda_2$)

- Robust recovery from imperfect or incomplete $\hat{\mathbf{V}}$ [Segarra et al'16]

Step 1: Obtaining the eigenvectors

Stationary graph signal [Marques et al'16]

Def: A graph signal $\mathbf{y}$ is stationary with respect to the shift $\mathbf{S}$ if and only if $\mathbf{y} = \mathbf{H}\mathbf{x}$, where $\mathbf{H} = \sum_{l=0}^{L-1} h_l \mathbf{S}^l$ and $\mathbf{x}$ is white.

Step 1: Obtaining the eigenvectors

Stationary graph signal [Marques et al'16]

Def: A graph signal $\mathbf{y}$ is stationary with respect to the shift $\mathbf{S}$ if and only if $\mathbf{y} = \mathbf{H}\mathbf{x}$, where $\mathbf{H} = \sum_{l=0}^{L-1} h_l \mathbf{S}^l$ and $\mathbf{x}$ is white.

- The covariance matrix of the stationary signal $\mathbf{y}$ is

$$\mathbf{C}_y = \mathbb{E}\left[\mathbf{H}\mathbf{x}(\mathbf{H}\mathbf{x})^T\right] = \mathbf{H}\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]\mathbf{H}^T = \mathbf{H}\mathbf{H}^T$$

- Key: Since $\mathbf{H}$ is diagonalized by $\mathbf{V}$, so is the covariance $\mathbf{C}_y$

$$\mathbf{C}_y = \mathbf{V}\left|\sum_{l=0}^{L-1} h_l \boldsymbol{\Lambda}^l\right|^2 \mathbf{V}^T$$

$\Rightarrow$ Estimate $\mathbf{V}$ from $\{\mathbf{y}_i\}$ via Principal Component Analysis

Non-stationary graph signals

- **Q:** What if the signal $\mathbf{y} = \mathbf{H}\mathbf{x}$ is not stationary (i.e., $\mathbf{x}$ colored)?
  - $\Rightarrow$ Matrices $\mathbf{S}$ and $\mathbf{C}_y$ no longer simultaneously diagonalizable since

$$\mathbf{C}_y = \mathbf{H}\mathbf{C}_x\mathbf{H}^T$$

Non-stationary graph signals

- **Q:** What if the signal $\mathbf{y} = \mathbf{H}\mathbf{x}$ is not stationary (i.e., $\mathbf{x}$ colored)?
  $\Rightarrow$ Matrices $\mathbf{S}$ and $\mathbf{C}_y$ no longer simultaneously diagonalizable since

$$\mathbf{C}_y = \mathbf{H}\mathbf{C}_x\mathbf{H}^T$$

- Key: still $\mathbf{H} = \sum_{l=0}^{L-1} h_l \mathbf{S}^l$ diagonalized by the eigenvectors $\mathbf{V}$ of $\mathbf{S}$
  $\Rightarrow$ Infer $\mathbf{V}$ by estimating the unknown diffusion (graph) filter $\mathbf{H}$
  $\Rightarrow$ Step 1 boils down to system identification + eigendecomposition
- Leverage different sources of information on the input signal $\mathbf{x}$
  (a) Input-output graph signal realization pairs $\{\mathbf{y}_m, \mathbf{x}_m\}$
  (b) Input covariance $\mathbf{C}_x$ and positive semidefinite filter $\mathbf{H} \succcurlyeq \mathbf{0}$
  (c) Input covariance $\mathbf{C}_x$ and generic filter $\mathbf{H}$

Input-output graph signal realization pairs

- Consider M diffusion processes on G, where $\mathbf{y}_m = \mathbf{H}\mathbf{x}_m$ ($\mathbf{x}_m$ colored)
  - $\Rightarrow$ Assume that realizations $\{\mathbf{y}_m, \mathbf{x}_m\}_{m=1}^{M}$ are available
- Filter $\mathbf{H}$ and, as byproduct, its eigenvectors $\mathbf{V}$ can be estimated as

$$\hat{\mathbf{H}} = \underset{\mathbf{H}}{\text{argmin}} \sum_{m=1}^{M} \|\mathbf{y}_m - \mathbf{H}\mathbf{x}_m\|^2$$

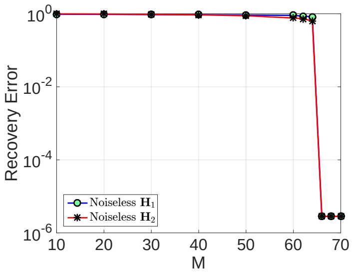
- Define $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_M]$ and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_M]$. Then, $\hat{\mathbf{H}}$ given by

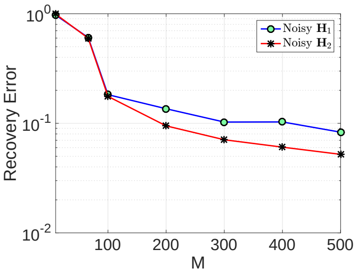
$$\text{vec}(\hat{\mathbf{H}}) = \left((\mathbf{X}^T)^\dagger \otimes \mathbf{I}_N\right)\text{vec}(\mathbf{Y})$$

$\Rightarrow$ If $M \geq N$ and $\mathbf{X}$ is full rank, the minimizer $\hat{\mathbf{H}}$ is unique

Inferring a brain network

- Consider a structural brain graph with $N = 66$ neural regions
  - Signals diffused either by $\mathbf{H}_1 = \sum_{l=0}^{2} h_l \mathbf{A}^l$ or $\mathbf{H}_2 = (\mathbf{I} + \alpha \mathbf{A})^{-1}$
  - Observe realizations $\{\mathbf{y}_m, \mathbf{x}_m\}_{m=1}^{M}$ and vary M
- Also noisy signals $\mathbf{y}_m = \mathbf{H}_i \mathbf{x}_m + \mathbf{w}_m$, with $\mathbf{w}_m \sim \mathcal{N}(\mathbf{0}, 10^{-2}\mathbf{I})$





- Recovery error $\|\mathbf{A} - \hat{\mathbf{A}}\|_F / \|\mathbf{A}\|_F$ small for $M \geq 66$, even with noise

  $\Rightarrow$ Performance roughly independent of the filter type

Input covariance and positive semidefinite filters

- Realizations of the input may be challenging to acquire
  - $\Rightarrow$ Consider instead that $\mathbf{C}_{\mathbf{x},m} = \mathbb{E}[\mathbf{x}_m \mathbf{x}_m{}^T]$ are known
  - $\Rightarrow$ Estimate output covariance $\hat{\mathbf{C}}_{\mathbf{y},m}$ from realizations $\{\mathbf{y}_m^{(p)}\}_{p=1}^{P_m}$
- Goal is to find $\mathbf{H}$ such that $\hat{\mathbf{C}}_{\mathbf{y},m}$ and $\mathbf{H}\mathbf{C}_{\mathbf{x},m}\mathbf{H}^T$ are close
  - $\Rightarrow$ Least squares yields a fourth-order cost in $\mathbf{H}$ $\rightarrow$ Challenging

Input covariance and positive semidefinite filters

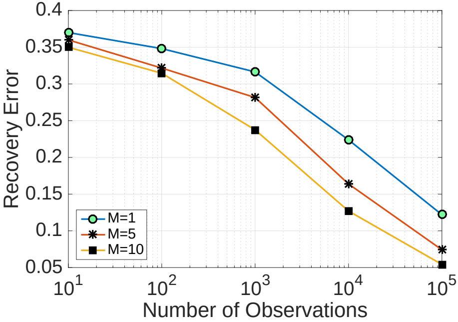
- Realizations of the input may be challenging to acquire
  - $\Rightarrow$ Consider instead that $\mathbf{C}_{\mathbf{x},m} = \mathbb{E}[\mathbf{x}_m \mathbf{x}_m{}^T]$ are known
  - $\Rightarrow$ Estimate output covariance $\hat{\mathbf{C}}_{\mathbf{y},m}$ from realizations $\{\mathbf{y}_m^{(p)}\}_{p=1}^{P_m}$
- Goal is to find $\mathbf{H}$ such that $\hat{\mathbf{C}}_{\mathbf{y},m}$ and $\mathbf{H}\mathbf{C}_{\mathbf{x},m}\mathbf{H}^T$ are close
  - $\Rightarrow$ Least squares yields a fourth-order cost in $\mathbf{H}$ $\rightarrow$ Challenging
- Assume $\mathbf{H}$ is PSD, e.g, in Laplacian diffusion $\mathbf{y} = (\sum_{l=0}^{\infty} \beta_l \mathbf{L}^l)\mathbf{x}$, $\beta_l > 0$
  - $\Rightarrow$ Well-defined square roots, hence $\mathbf{H}$ can be identified as

$$\hat{\mathbf{H}} = \operatorname*{argmin}_{\mathbf{H} \in \mathcal{M}_{++}^N} \sum_{m=1}^{M} \|(\mathbf{C}_{\mathbf{x},m}{}^{1/2} \hat{\mathbf{C}}_{\mathbf{y},m} \mathbf{C}_{\mathbf{x},m}{}^{1/2})^{1/2} - \mathbf{C}_{\mathbf{x},m}{}^{1/2} \mathbf{H} \mathbf{C}_{\mathbf{x},m}{}^{1/2}\|_F^2$$

- If $\mathbf{C}_{y,1}$ known, even with $M = 1$ PSD assumption renders $\mathbf{H}$ identifiable

Inferring Zachary's karate club network

- Social network with $N = 34$ club members
  - Model opinion diffusion with $\mathbf{S} = \mathbf{I} - \alpha\mathbf{L}$, where $\alpha = \lambda_{max}^{-1}(\mathbf{L})$
  - For $M = 1, 5, 10$ input covariances $\mathbf{C}_{x,m}$ assumed given
  - Estimate $\mathbf{C}_{y,m}$ from $\{\mathbf{y}_m^{(p)}\}_{p=1}^{P_m}$ via sample averaging, varying P_m



- With imperfect estimates $\hat{\mathbf{C}}_{\mathbf{y},m}$, performance improves with M

Input covariance and generic filters

- **Q:** What about identifying a generic symmetric filter $\mathbf{H}$?
- Filter is no longer PSD, square roots not prudent $\Rightarrow$ Try to solve

$$\hat{\mathbf{H}} = \underset{\mathbf{H}\in\mathcal{M}^N}{\operatorname{argmin}} \sum_{m=1}^{M} \|\hat{\mathbf{C}}_{\mathbf{y},m} - \mathbf{H}\mathbf{C}_{\mathbf{x},m}\mathbf{H}^T\|_F^2$$

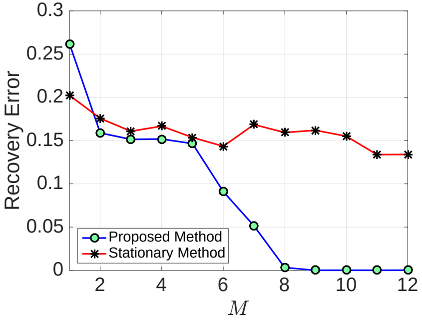
- Non-convex problem can be tackled by gradient descent or ADMM

$$\{\mathbf{H}_L^*, \mathbf{H}_R^*\} = \underset{\mathbf{H}_\mathbf{L},\mathbf{H}_\mathbf{R}\in\mathcal{M}^N}{\operatorname{argmin}} \sum_{m=1}^{M} ||\mathbf{C}_{\mathbf{y},m} - \mathbf{H}_\mathbf{L}\mathbf{C}_{\mathbf{x},m}\mathbf{H}_\mathbf{R}^T||_F^2 \quad \text{s. to } \mathbf{H}_\mathbf{L} = \mathbf{H}_\mathbf{R}$$

$\Rightarrow$ In general, identifiability cannot be guaranteed. Larger M helps

Inferring a brain network

- Consider a structural brain graph with $N = 66$ neural regions
  - Signals diffused by $\mathbf{H} = \sum_{l=0}^{2} h_l \mathbf{A}^l$, $h_l \sim \mathcal{U}[0,1]$
- Performance comparison against counterpart in [Segarra et al'16]
  - Assumes $\mathbf{y}_m$ stationary $\Rightarrow$ Estimates $\mathbf{V}$ directly from $\hat{\mathbf{C}}_{\mathbf{y},m}$



- Error decays with M, almost all edges in $\mathbf{S}$ recovered for $M = 9$

  $\Rightarrow$ Outperforms algorithm agnostic to signal non-stationarities

Closing remarks

- Network topology inference from diffused non-stationary graph signals
  - Graph shift $\mathbf{S}$ and covariance $\mathbf{C}_y$ are not simultaneously diagonalizable
- Diffusion filter $\mathbf{H}$ and graph shift $\mathbf{S}$ still share spectral templates $\mathbf{V}$

  $\Rightarrow$ Two step approach for topology inference

  i) Obtain $\hat{\mathbf{H}} \Rightarrow \hat{\mathbf{V}}$; ii) Given $\hat{\mathbf{V}}$, estimate $\hat{\mathbf{S}}$ via convex optimization
- Estimate $\hat{\mathbf{H}}$ under different settings
  - Input-output graph signal realization pairs $\{\mathbf{y}_m, \mathbf{x}_m\}$
  - Input covariance $\mathbf{C}_x$ and positive semidefinite filter $\mathbf{H} \succcurlyeq \mathbf{0}$
  - Input covariance $\mathbf{C}_x$ and generic filter $\mathbf{H}$

Closing remarks

- Network topology inference from diffused non-stationary graph signals
  - Graph shift $\mathbf{S}$ and covariance $\mathbf{C}_y$ are not simultaneously diagonalizable
- Diffusion filter $\mathbf{H}$ and graph shift $\mathbf{S}$ still share spectral templates $\mathbf{V}$

  $\Rightarrow$ Two step approach for topology inference

  i) Obtain $\hat{\mathbf{H}} \Rightarrow \hat{\mathbf{V}}$; ii) Given $\hat{\mathbf{V}}$, estimate $\hat{\mathbf{S}}$ via convex optimization
- Estimate $\hat{\mathbf{H}}$ under different settings
  - Input-output graph signal realization pairs $\{\mathbf{y}_m, \mathbf{x}_m\}$
  - Input covariance $\mathbf{C}_x$ and positive semidefinite filter $\mathbf{H} \succcurlyeq \mathbf{0}$
  - Input covariance $\mathbf{C}_x$ and generic filter $\mathbf{H}$
- Ongoing work and future directions
  - Identifiability and convergence guarantees for generic $\mathbf{H}$
  - Extensions to directed graphs
  - Inference of time-varying networks