Sparse Sampling for Statistical and Graph Signal Processing

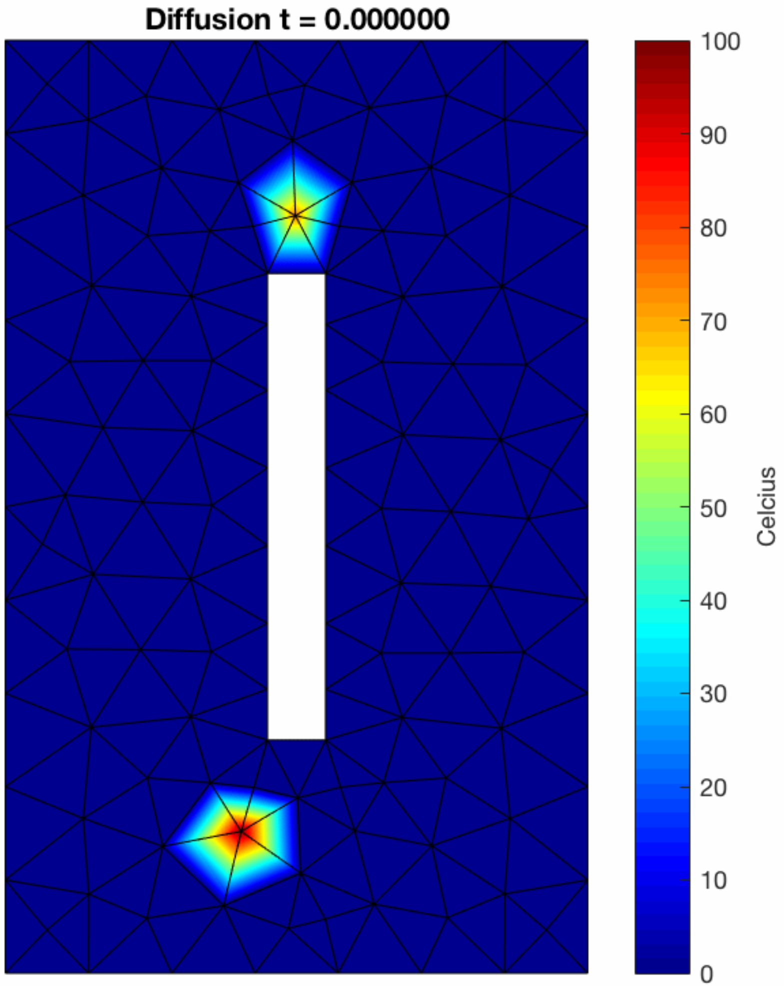
Day 3: Graph sampling

dr. ir. Sundeep Chepuri
email: s.p.chepuri@tudelft.nl

TU Delft
Delft University of Technology

Roadmap

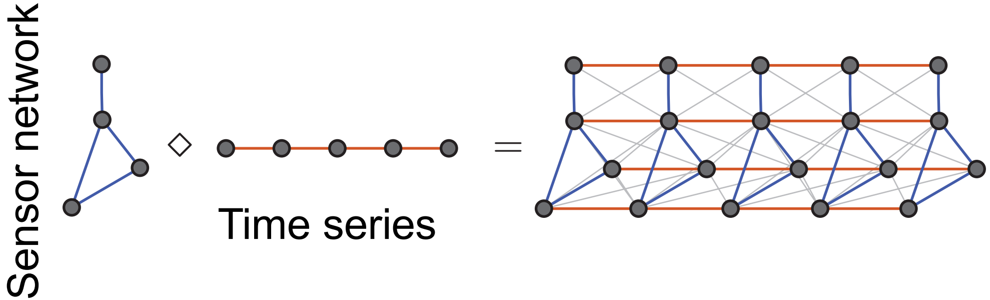
- ☐ Introduction and context
- ☐ Signal processing on graphs
- ☐ Signal reconstruction
- ☐ Multi-domain (tensor) signal reconstruction
- ☐ Covariance estimation
- ☐ Sparse sampler design
- ☐ Graph learning
- ☐ Conclusions, Q&A



Frozen metal plate with cavity excited with two hotspots

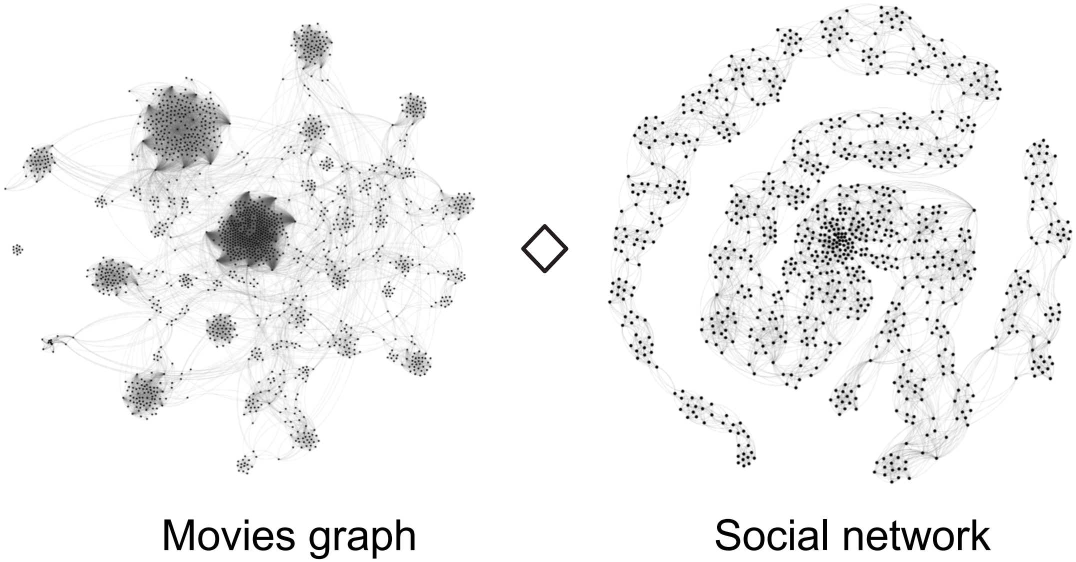
How to optimally deploy sensors?

Temperature on Earth's surface



Sensing networks

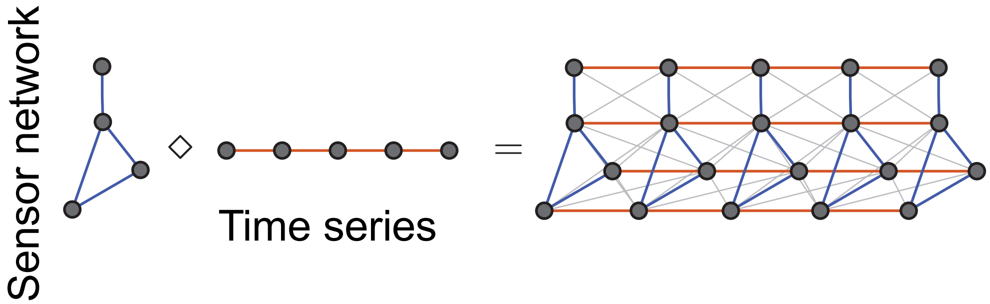
3D point clouds (Kinect, LiDAR)

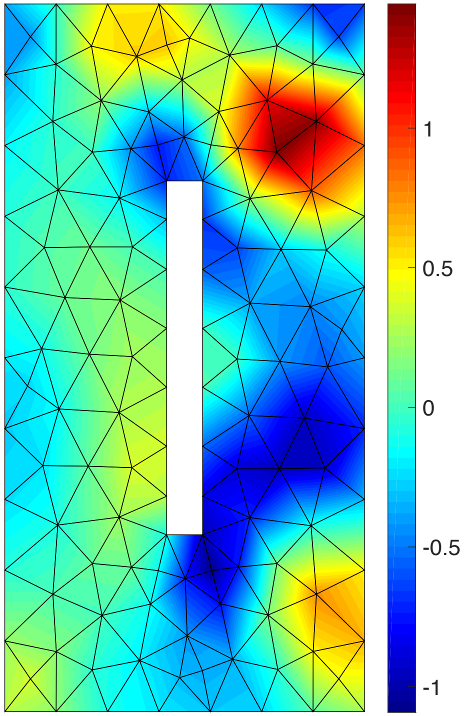


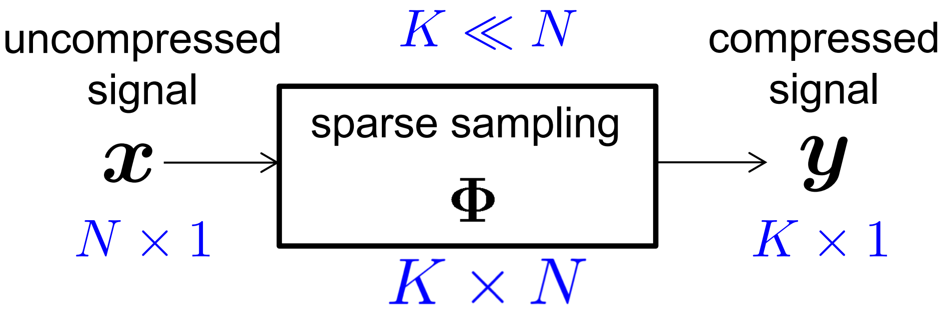
Recommender systems

Design sparse samplers taking into account the underlying topology

Sparse sampling on irregular domains







uncompressed signal

$\boldsymbol{x}$

$N \times 1$

$K \ll N$

sparse sampling

$\boldsymbol{\Phi}$

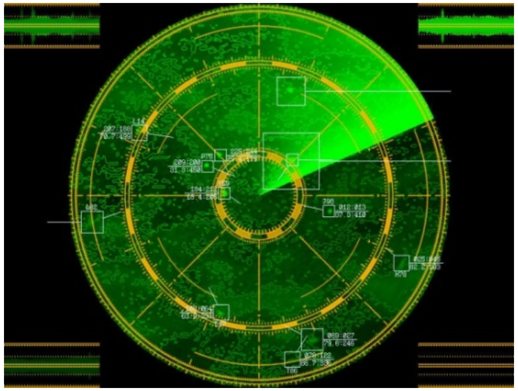
$K \times N$

compressed signal

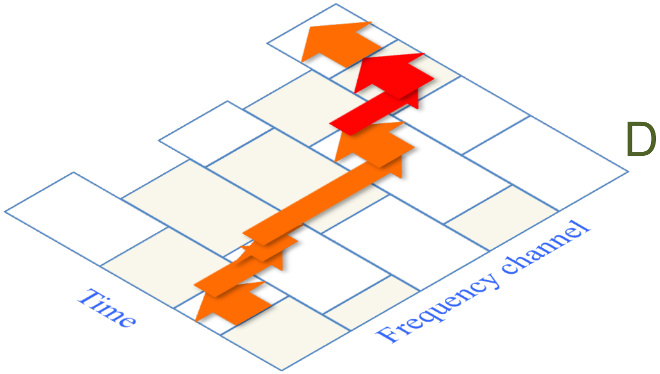
$\boldsymbol{y}$

$K \times 1$

Given y estimate x

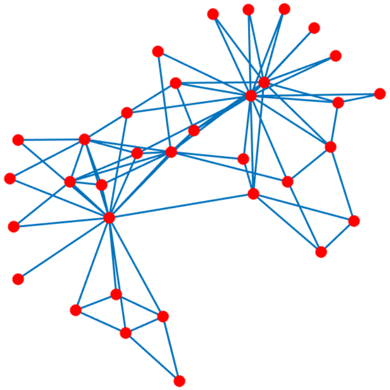


Radar
Doppler + angular spectra



Cognitive radio
frequency spectrum

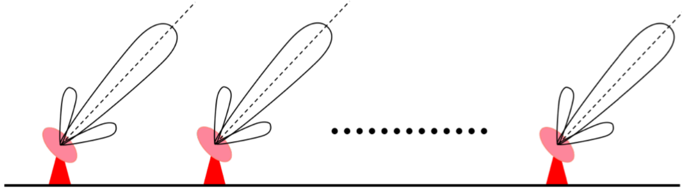
Radio astronomy
spatial spectrum

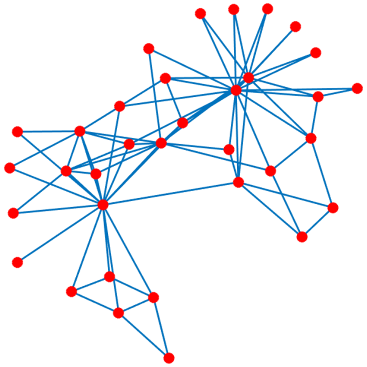


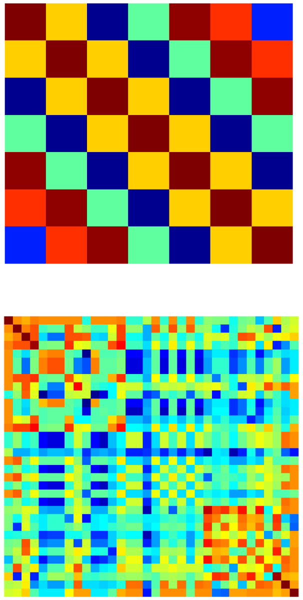
Graph-based inference
graph spectrum

Design sparse samplers taking into account the data structure

Sparse sampling on irregular domains



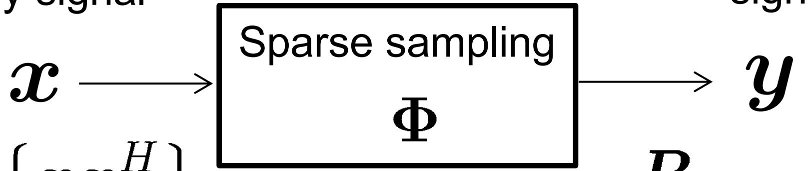




uncompressed stationary signal

$K \ll N$

compressed signal



Sparse sampling

$\boldsymbol{\Phi}$

$\boldsymbol{x}$

$\boldsymbol{y}$

$$\boldsymbol{R}_{\boldsymbol{x}} = \mathrm{E}\left\{\boldsymbol{x}\boldsymbol{x}^H\right\}$$

$N \times N$

$K \times N$

$$\boldsymbol{R}_{\boldsymbol{y}} = \mathrm{E}\left\{\boldsymbol{y}\boldsymbol{y}^H\right\}$$

$K \times K$

Given $\boldsymbol{R}_{\boldsymbol{y}}$ or several realizations of $\boldsymbol{y}$ estimate $\boldsymbol{R}_{\boldsymbol{x}}$

What is sparse sampling?

$$\boldsymbol{\Phi}(\boldsymbol{w}) \in \{0,1\}^{K\times N}$$

$$\boldsymbol{R}_{\boldsymbol{y}} = \mathrm{E}\left\{\boldsymbol{y}\boldsymbol{y}^H\right\}$$

$$\boldsymbol{R}_{\boldsymbol{x}} = \mathrm{E}\left\{\boldsymbol{x}\boldsymbol{x}^H\right\}$$

- Sampling matrix is determined by the sampling vector/set

$$\boldsymbol{w} = [w_1, w_2, \dots, w_N]^T \in \{0,1\}^N \quad \text{or} \quad \mathcal{S} = \{n | w_n = 1, n = 1, 2, \dots, N\}$$

$w_m = (0)1$ sample or vertex is (not) selected

- Sparse sampling structure
  - only one nonzero entry per row
  - many zero columns

Why sparse sampling?

- **Economical** constraints (hardware cost)
- Limited **physical space**
- Limited data **storage space**
- Reduce **communications bandwidth**
- Reduce **processing overhead**

Today's plan

We will cover the following two aspects:

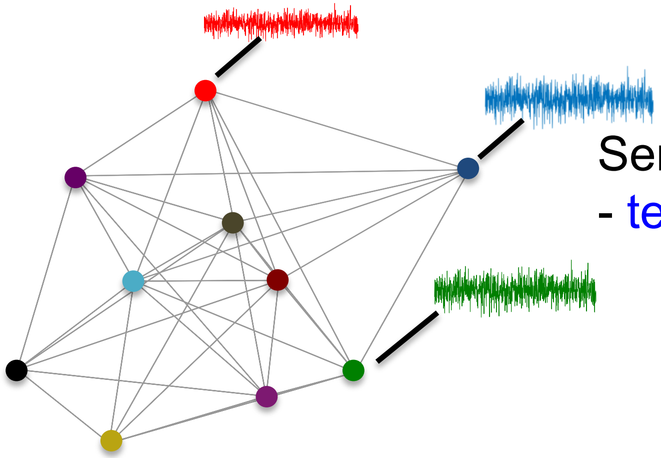
1. Reconstruction of signals and second-order statistics from subsampled measurements by taking into account the domain on which the data is defined as a prior information

2. Efficient near-optimal methods to design sparse samplers

Signal Processing on Graphs

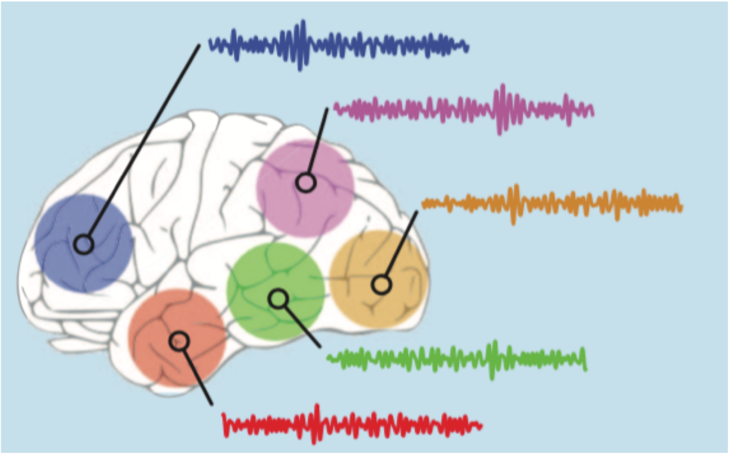
- D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” IEEE Signal Process. Mag., vol. 30, no. 3, pp. 83–98, 2013.

- A. Sandryhaila and J. M. Moura, “Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure,” IEEE Signal Process. Mag., vol. 31, no. 5, pp. 80–90, 2014.



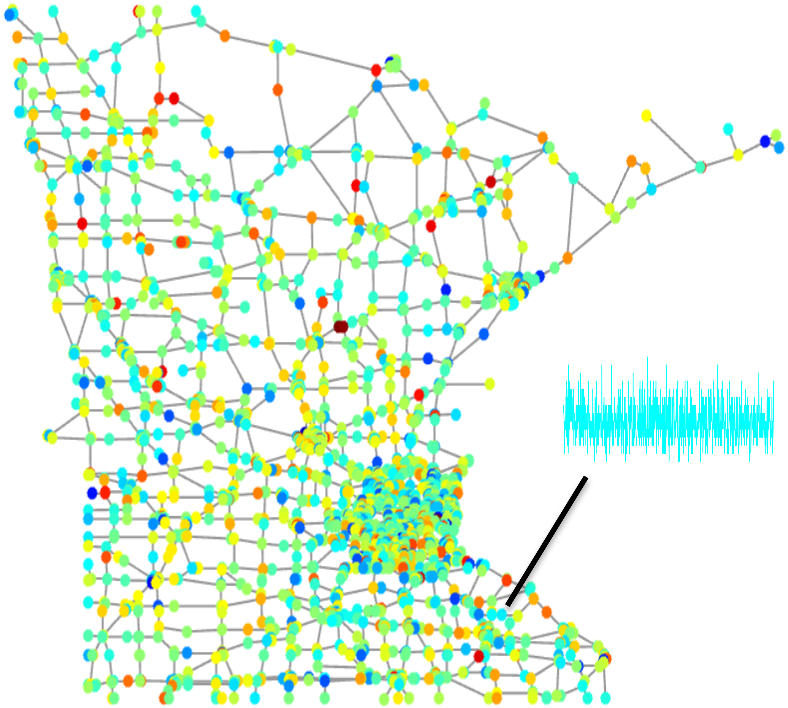
Sensing networks

- temp., pressure, air quality monitoring



Brain networks

- fMRI time series, EEG signals



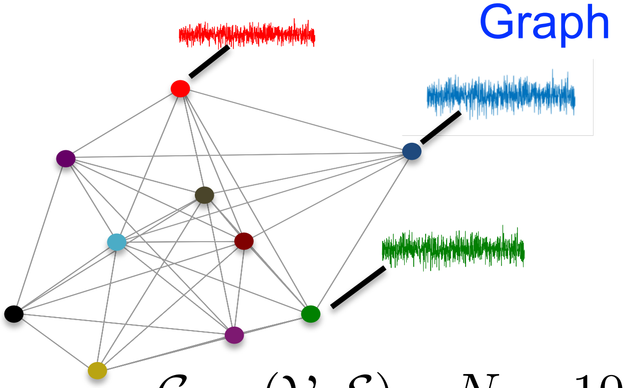
Transport networks

- # vehicles crossing a junction

Signals and random processes on graphs

Graphs and graph signals

- Datasets with *irregular support* can be represented using a graph



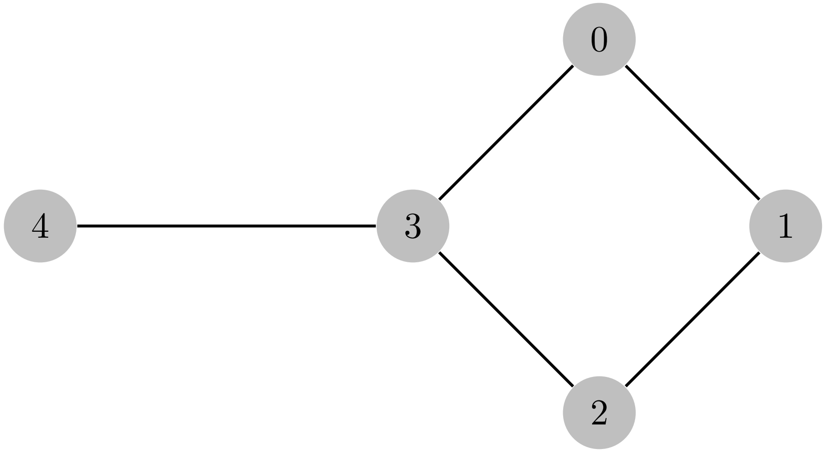
Graph signal

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}) \quad N = 10$$

- $\mathcal{V}$ is the set of nodes
- $\mathcal{E}$ is the set of edges
- $\boldsymbol{x} \in \mathbb{R}^N$ represents the graph signal

- Graph is represented using the matrix $\boldsymbol{S} \in \mathbb{R}^{N \times N}$
  - $[\boldsymbol{S}]_{i,j}$ is nonzero only if $i = j$ and/or $(i, j) \in \mathcal{E}$
  - $\boldsymbol{S}$ could be graph Laplacian, adjacency matrix, or ...
  - $\boldsymbol{S}$ is referred to as the graph-shift operator

Graph Laplacian



$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$$

$$= \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

diagonal degree matrix adjacency matrix

- For an *undirected graph*, $\boldsymbol{L}$ is symmetric

$$\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^H$$

$$= [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_N] \operatorname{diag}(\lambda_1, \cdots, \lambda_N) [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_N]^H$$

- $\boldsymbol{L}\mathbf{1} = \mathbf{0}$, so

$$0 = \lambda_1 \le \lambda_2 \le \cdots \le \lambda_N$$

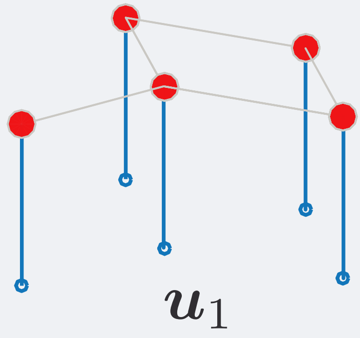
Graph Laplacian - eigenmodes

Frequency interpretation of the eigenvectors (viewed as signals on graphs)

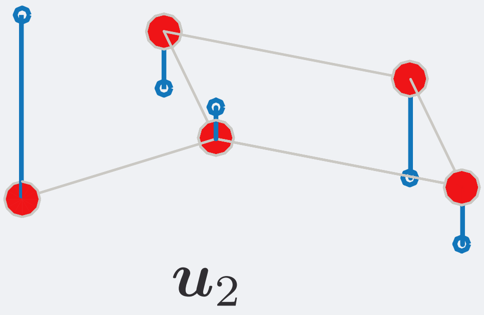
eigenvalues

eigenvectors

$$\boldsymbol{\lambda} = \begin{bmatrix} 0 \\ 0.8299 \\ 2 \\ 2.6889 \\ 4.4812 \end{bmatrix} \qquad \boldsymbol{U} = \begin{bmatrix} -0.4472 & -0.2560 & 0.7071 & 0.2422 & -0.4193 \\ -0.4472 & -0.4375 & 0 & -0.7031 & 0.3380 \\ -0.4472 & -0.2560 & -0.7071 & 0.2422 & -0.4193 \\ -0.4472 & 0.1380 & 0 & 0.5362 & 0.7024 \\ -0.4472 & 0.8115 & 0 & -0.3175 & -0.2018 \end{bmatrix}$$

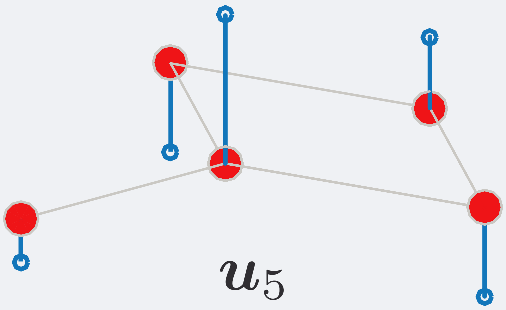


DC (no zero crossing)



two zero crossings

...



five zero crossings

Sign transitions of eigenvectors increase with eigenvalues

Fourier-like orthogonal basis

$$\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^H$$

$$= \underbrace{[\boldsymbol{u}_1,\cdots,\boldsymbol{u}_N]}_{\text{Fourier-like basis for the graph}}\ \underbrace{\mathrm{diag}(\lambda_1,\cdots,\lambda_N)}_{\text{Spectrum of the graph}}\ [\boldsymbol{u}_1,\cdots,\boldsymbol{u}_N]^H$$

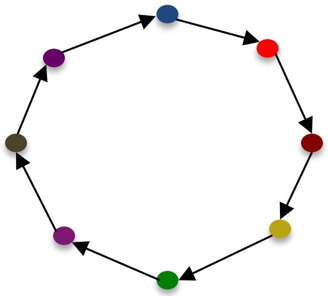
- Holds for graph Laplacians and adjacency matrices
  - Frequency interpretation based on zero crossings or total variation

- For undirected graphs
  - Eigenvalues are all real (*graph-shift operator is symmetric*)

- For directed graphs with normal S
  - Eigenvalues occur in complex conjugate pairs

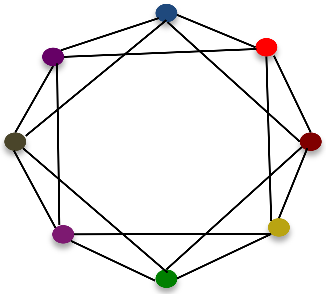
Time-domain as a graph

- The DFT and the traditional frequency grid is obtained by the adjacency matrix of the cycle graph



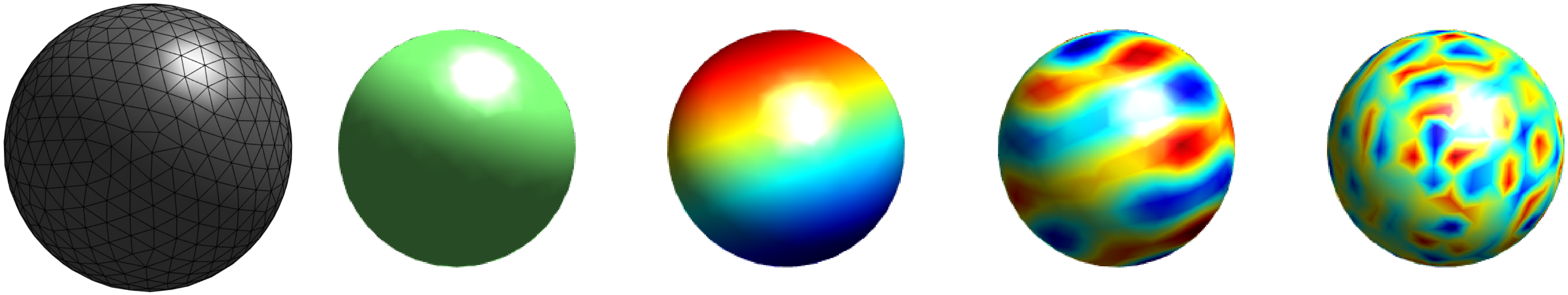
$$\boldsymbol{S} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- Any circulant graph in principle leads to the DFT as the graph Fourier transform

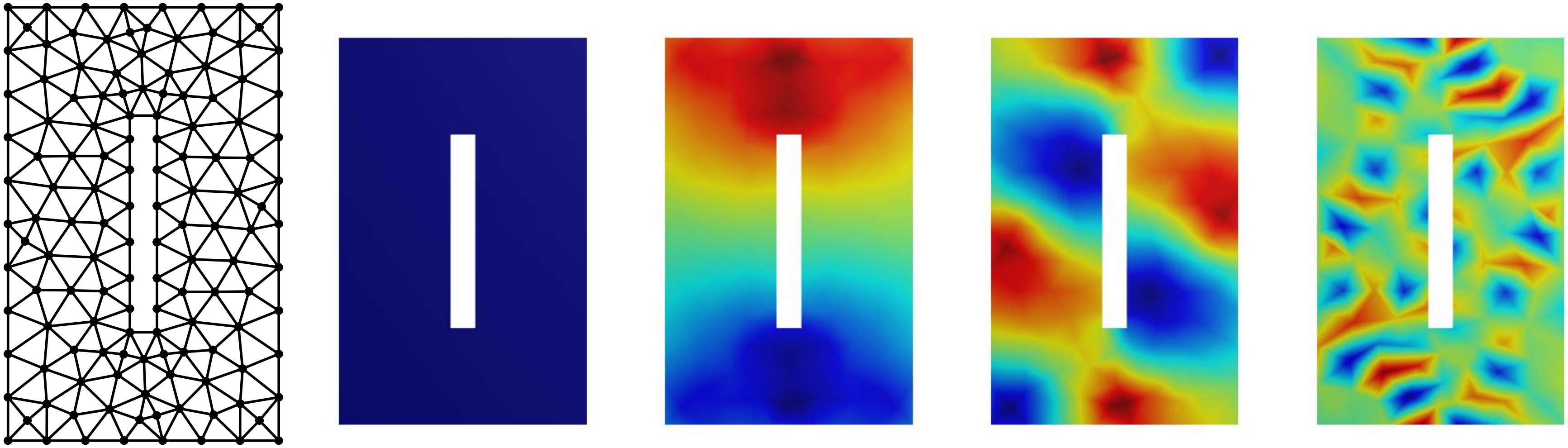


$$\boldsymbol{S} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Fourier-like basis on meshes



(Laplace's) spherical harmonics

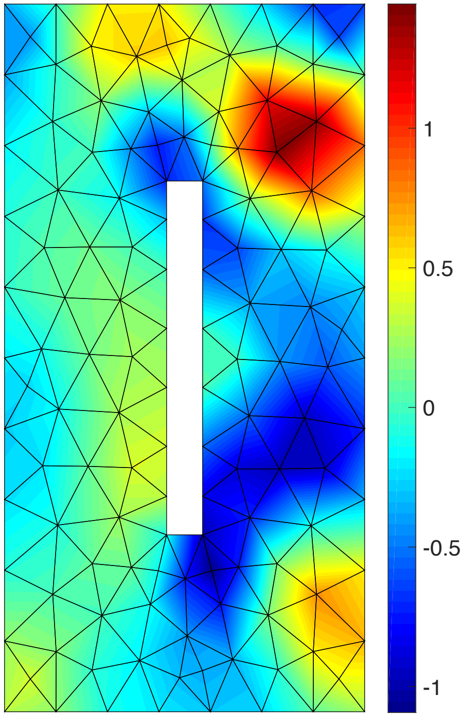


Fourier-like oscillating modes of the metal plate with cavity

Graph Fourier transform

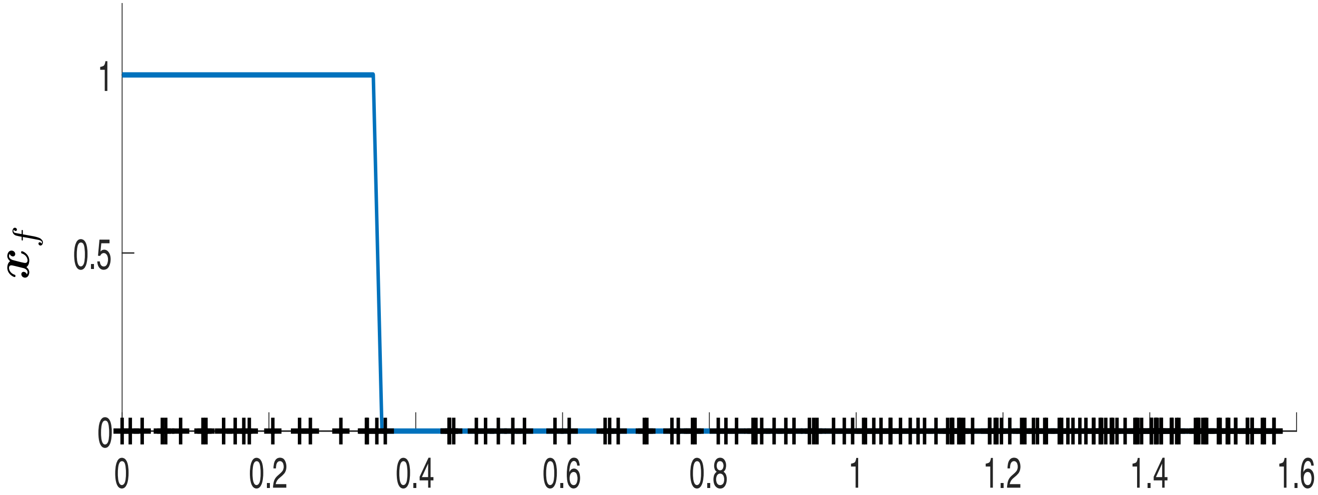
Decomposition of the (graph) signal $\boldsymbol{x} \in \mathbb{R}^N$ w.r.t. the orthonormal basis $\boldsymbol{U}$

(analysis) $$\boldsymbol{x}_f := \boldsymbol{U}^H \boldsymbol{x} \Leftrightarrow \boldsymbol{x} =: \boldsymbol{U} \boldsymbol{x}_f$$ (synthesis)



Field distribution

$\boldsymbol{x}$ is the field values measured at mesh points



Laplacian eigenvalues

(non-uniform discrete frequency grid)

Graph filters

- *Graph filters* (polynomial of the *graph-shift* operator) can be used to modify the frequency content of graph signals

$$\boldsymbol{H} = \sum_{l=0}^{L-1} h_l \boldsymbol{S}^l = \boldsymbol{U} \left(\sum_{l=0}^{L-1} h_l \boldsymbol{\Lambda}^l \right) \boldsymbol{U}^H = \boldsymbol{U} \text{diag}(\boldsymbol{h}_f) \boldsymbol{U}^H$$

Shift invariant: $\boldsymbol{HS} = \boldsymbol{SH}$ and distributable: $\boldsymbol{x}_l = \boldsymbol{S}\boldsymbol{x}_{l-1}$

- Vertex-domain vs. frequency-domain implementation

Vertex-domain implementation: $\boldsymbol{y} = \boldsymbol{Hx}$

Frequency-domain implementation: $\boldsymbol{y}_f = \text{diag}(\boldsymbol{h}_f)\boldsymbol{x}_f$

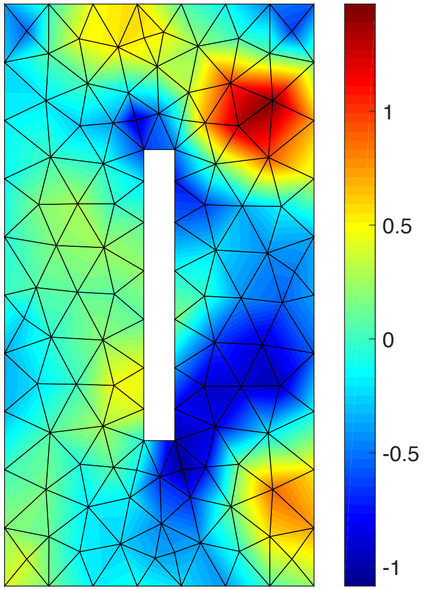
  - No fast GFT implementations
  - Parametrized filter implementation in the vertex-domain is possible

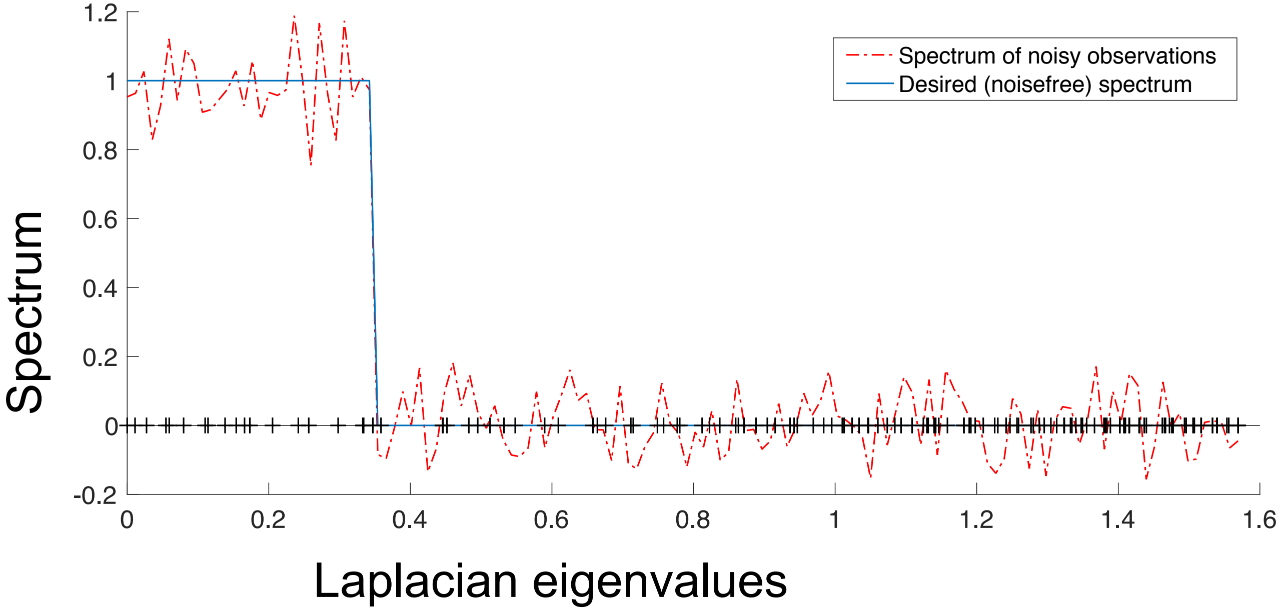
Graph filters

- *Graph filters* (polynomial of the *graph-shift* operator) can be used to modify the frequency content of graph signals

$$\boldsymbol{H} = \sum_{l=0}^{L-1} h_l \boldsymbol{S}^l = \boldsymbol{U}\left(\sum_{l=0}^{L-1} h_l \boldsymbol{\Lambda}^l\right)\boldsymbol{U}^H = \boldsymbol{U}\text{diag}(\boldsymbol{h}_f)\boldsymbol{U}^H$$

Denoising example:





Graph Signal Sampling

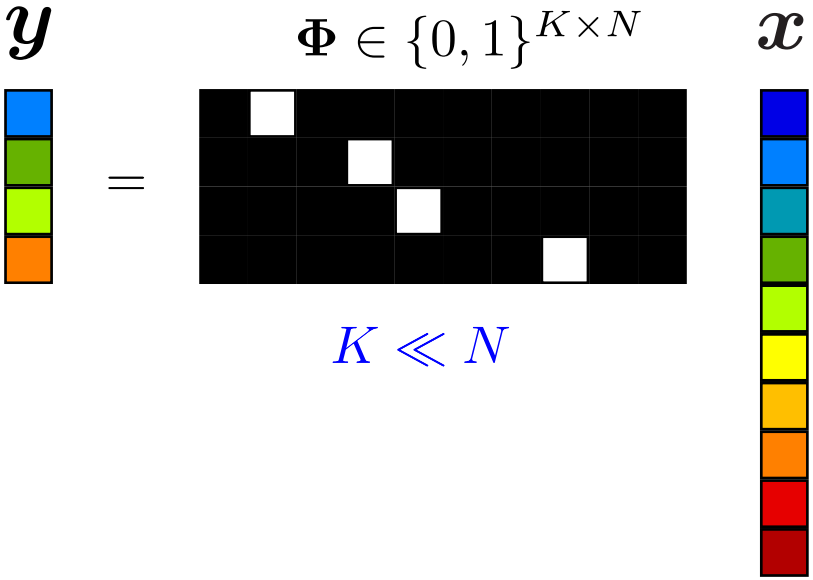
- S.P. Chepuri, Y. Eldar and G. Leus. Graph Sampling With and Without Input Priors. In Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2018), Calgary, Canada, April 2018.

- S. Chen, R. Varma, A. Sandryhaila, and J. Kovacevic, “Discrete signal processing on graphs: Sampling theory,” IEEE TSP, vol. 63, no. 24, pp. 6510–6523, Dec. 2015.

- D. Romero, M. Ma, and G.B. Giannakis. Kernel-Based Reconstruction of Graph Signals, IEEE TSP, vol. 65, no. 3, pp. 764–778, Feb 2017.

- A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, “Sampling of graph signals with successive local aggregations,” IEEE TSP, vol. 64, no. 7, pp. 1832–1834, Arp. 2016.

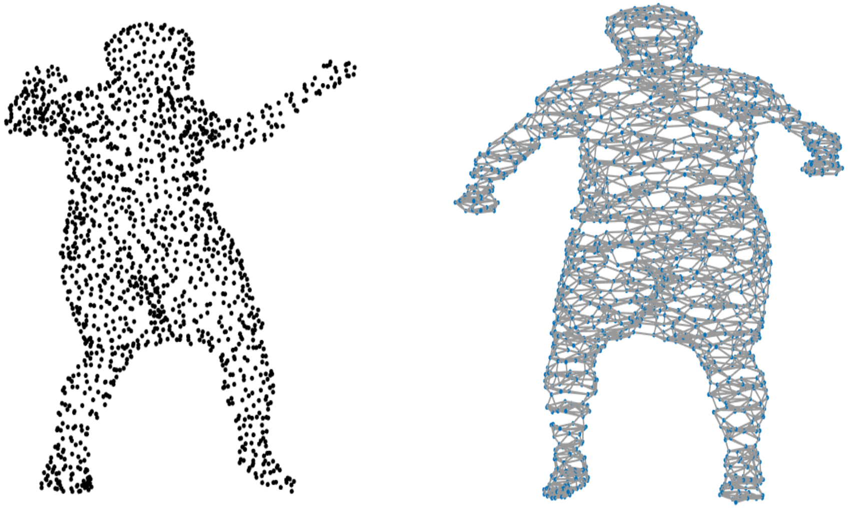
Sparse graph sampling



$\boldsymbol{y}$ $=$ $\boldsymbol{\Phi} \in \{0,1\}^{K \times N}$ $\boldsymbol{x}$

$K \ll N$

graph signal



Given y estimate x

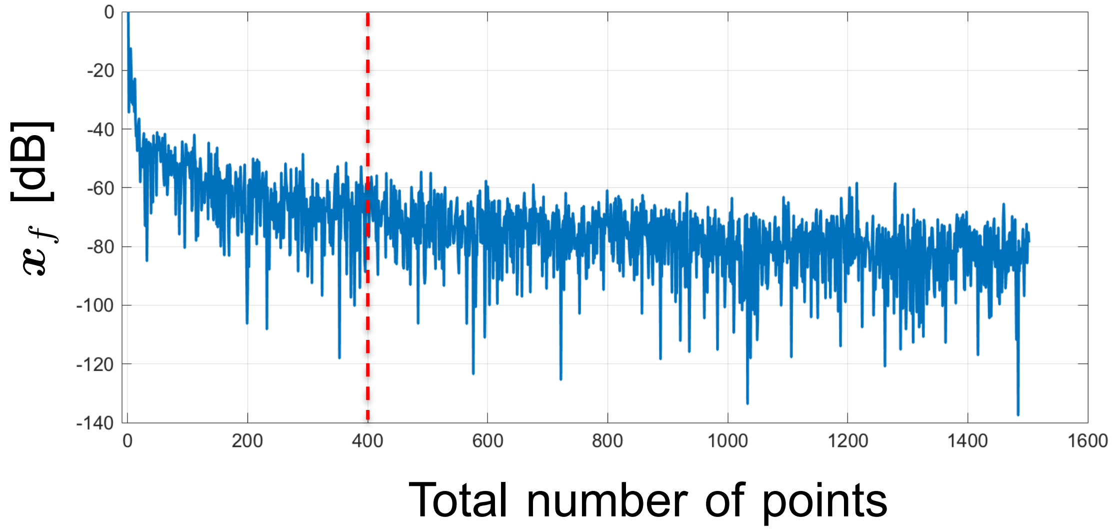
signal: 3D points, which are displacements of graph nodes

Bandlimited graph signals – subspace prior

Suppose the support of the sparse $\boldsymbol{x}_f$ is known

$$\boldsymbol{x} = \boldsymbol{U}\boldsymbol{x}_f = \left[\; \underset{N \times L}{\boldsymbol{U}_{\text{BL}}} \;\middle|\; \star \;\right] \begin{bmatrix} \overset{L \times 1}{\tilde{\boldsymbol{x}}_f} \\ \hline \boldsymbol{0} \end{bmatrix} \quad \Leftrightarrow \quad \boldsymbol{x} = \boldsymbol{U}_{\text{BL}}\tilde{\boldsymbol{x}}_f$$

$\boldsymbol{x} \in \text{range}(\boldsymbol{U}_{\text{BL}})$ —a known L-dimensional subspace



Bandlimited graph signals – subspace prior

With sparse sampling, we get K equations in L unknowns

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} = \boldsymbol{\Phi}\boldsymbol{U}_{\text{BL}}\tilde{\boldsymbol{x}}_f$$

If the matrix $\boldsymbol{\Phi}\boldsymbol{U}_{\text{BL}}$ has full column rank, i.e, range$(\boldsymbol{U}_{\text{BL}}) \cap$ null$(\boldsymbol{\Phi}) = \{0\}$:

Least squares solution: $\widehat{\tilde{\boldsymbol{x}}}_f = (\boldsymbol{\Phi}\boldsymbol{U}_{\text{BL}})^{\dagger}\boldsymbol{y}$

Design of $\boldsymbol{\Phi}$ crucial for the least-squares solution to be unique

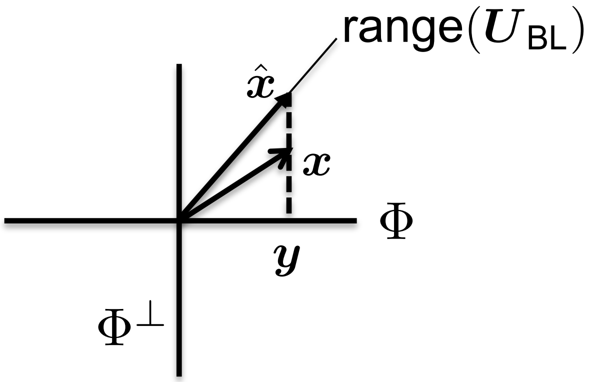
Bandlimited graph signals – subspace prior

- With sparse sampling, we get K equations in L unknowns

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} = \boldsymbol{\Phi}\boldsymbol{U}_{\text{BL}}\tilde{\boldsymbol{x}}_f$$

- *Oblique projection* of x onto the range$(\boldsymbol{U}_{\text{BL}})$ and along the null$(\boldsymbol{\Phi})$

$$\hat{\boldsymbol{x}} = \boldsymbol{U}_{\text{BL}}(\boldsymbol{U}_{\text{BL}}^H\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{U}_{\text{BL}})^{-1}\boldsymbol{U}_{\text{BL}}^H\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{x} = \boldsymbol{E}_{U_{\text{BL}}\Phi^{\perp}}\boldsymbol{x}$$

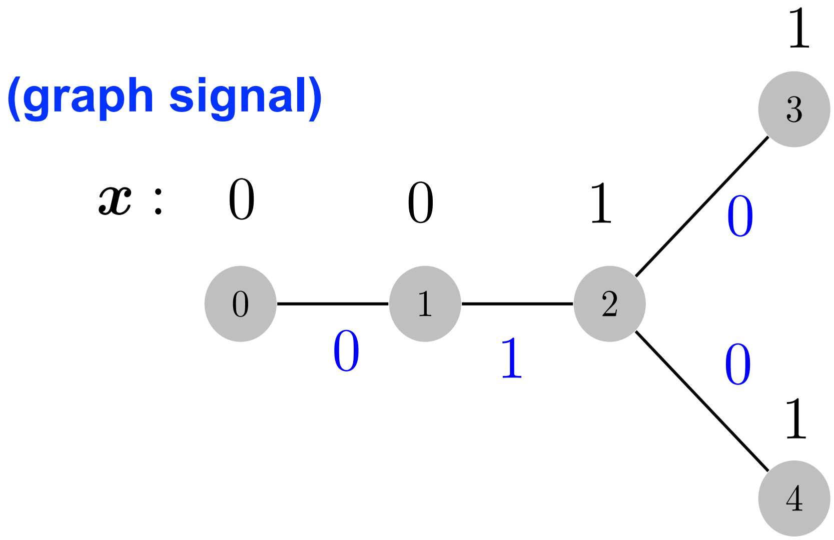


- A more interesting case, perhaps is, when the support is not known!

Reconstruction with smoothness prior

- Assume x is smooth with respect to the underlying graph or has small

$$\boldsymbol{x}^T \boldsymbol{L} \boldsymbol{x} = \sum_{(i,j)\in\mathcal{E}} (x_i - x_j)^2$$



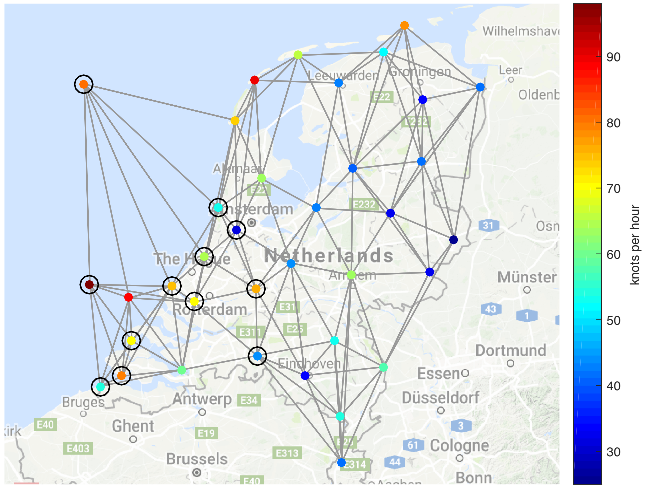
$$\boldsymbol{x}^T \boldsymbol{L} \boldsymbol{x} = \sum_{(i,j)\in\mathcal{E}} (x_i - x_j)^2$$

$$= 1$$

Sum of squares of differences across edges

Reconstruction with smoothness prior

- When the prior subspace is not known, we can be ***consistent*** (cf. interpolation)



$$\boldsymbol{\Phi x} = \boldsymbol{\Phi \hat{x}}$$

- Assume $\boldsymbol{x}$ is smooth with respect to the underlying graph or has small

- Equality constrained quadratic program

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \frac{1}{2}\boldsymbol{x}^H \boldsymbol{L x} \quad \text{subject to} \quad \boldsymbol{\Phi x} = \boldsymbol{y}$$

Solution:
$$\begin{bmatrix} \boldsymbol{L} + \boldsymbol{\Phi}^T\boldsymbol{\Phi} & \boldsymbol{\Phi}^T \\ \boldsymbol{\Phi} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}^T \boldsymbol{y} \\ \boldsymbol{y} \end{bmatrix}$$

If $\text{null}(\boldsymbol{L}) \cap \text{null}(\boldsymbol{\Phi}) = \{0\}$, then $\hat{\boldsymbol{x}} = \tilde{\boldsymbol{L}}(\boldsymbol{\Phi}\tilde{\boldsymbol{L}})^{-1}\boldsymbol{y}$

$$\tilde{\boldsymbol{L}} = (\boldsymbol{L} + \boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T$$

Sampling via graph filtering

Sparse sampling in spectral domain:

- Suppose sampling operator collects the first K contiguous frequencies

- Sampling and interpolation operations can be implemented via graph filters

$$\hat{\boldsymbol{x}} = \boldsymbol{H}_{\text{interp}} \boldsymbol{H}_{\text{samp}} \boldsymbol{x}.$$

- Subspace prior

diagonal matrix

$$\boldsymbol{\Phi} = \boldsymbol{E}_K \boldsymbol{U}^H \Rightarrow \boldsymbol{H}_{\text{samp}} = \boldsymbol{\Phi}^H \boldsymbol{\Phi} = \boldsymbol{U} \boldsymbol{E}_K^T \boldsymbol{E}_K \boldsymbol{U}^H$$

$$\boldsymbol{E}_K = [\boldsymbol{e}_1^T, \cdots, \boldsymbol{e}_K^T]^T \in \{0,1\}^{K \times N}$$

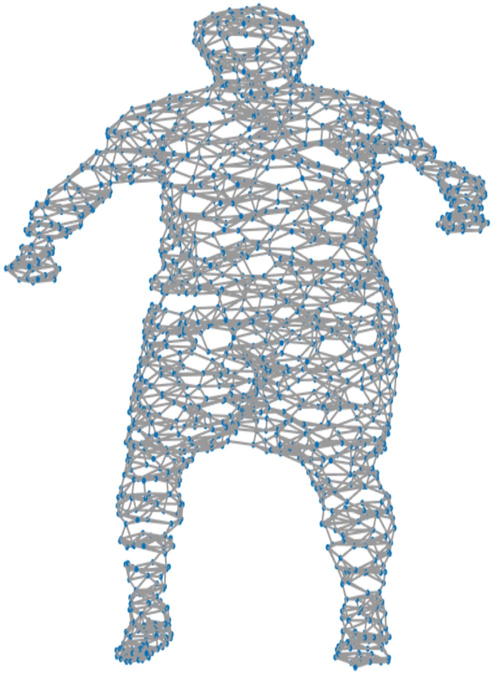
$$\boldsymbol{H}_{\text{interp}} = \boldsymbol{U}_{\text{BL}} \boldsymbol{H}_{f,\text{interp}} \boldsymbol{U}_{\text{BL}}^H \qquad \boldsymbol{H}_{f,\text{interp}}^{-1} = \boldsymbol{U}_{\text{BL}}^H \boldsymbol{H}_{\text{samp}} \boldsymbol{U}_{\text{BL}} \text{ (diagonal)}$$

- Smoothness prior

$$\boldsymbol{H}_{f,\text{samp}} = \boldsymbol{E}_K^T [\boldsymbol{E}_K (\boldsymbol{\Lambda} + \boldsymbol{E}_K^T \boldsymbol{E}_K)^{-1} \boldsymbol{E}_K^T]^{-1} \boldsymbol{E}_K \quad \text{(diagonal)}$$

$$\boldsymbol{H}_{\text{interp}} = \boldsymbol{U} (\boldsymbol{\Lambda} + \boldsymbol{E}_K^T \boldsymbol{E}_K)^{-1} \boldsymbol{U}^H$$

Numerical experiments



Graph (K-nearest neighbor)

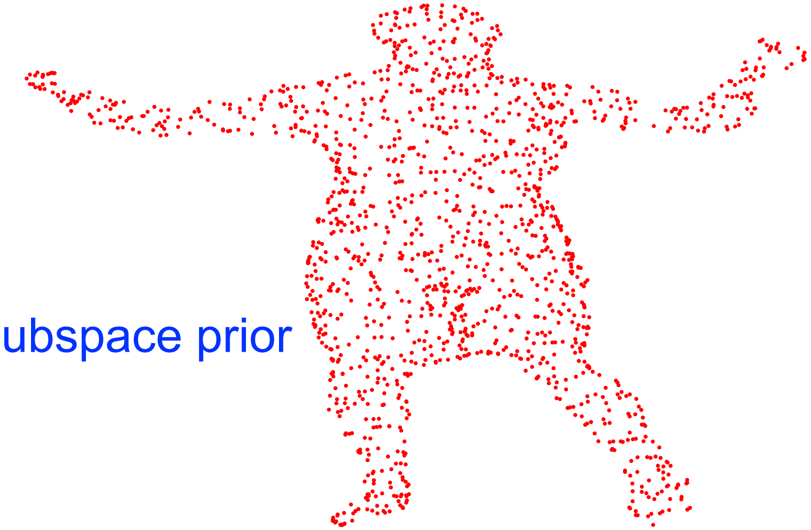
Original signal (3D points)

$N = 1502,\ K = 600,\ K/N \approx 40\%$ compression

Numerical experiments

Original signal

$N = 1502,\ K = 600,\ K/N \approx 40\%$ compression



Subspace prior

Smoothness prior

Kernel-based reconstruction

- Popular within machine learning for nonlinear function estimation

- Kernel methods seek an estimation of a function in a reproducing kernel Hilbert space (RKHS)

$$\mathcal{H} = \left\{ x : x(v) = \sum_{n=1}^{N} \alpha_n k(v, v_n), \ \alpha_n \in \mathbb{R} \right\}$$

basis functions

Kernel map $k : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$

$k(v_n, v_m)$ measures similarity between signal values at v_n and v_m

- Any graph signal can be assumed to be in RKHS

$$\boldsymbol{x} = \boldsymbol{K\alpha}$$

$[\boldsymbol{K}]_{n,m} = k(v_n, v_m)$

Kernel-based reconstruction

RKHS inner product of $x(v) = \sum_{n=1}^{N} \alpha_n k(v, v_n)$ and $x'(v) = \sum_{n=1}^{N} \alpha'_n k(v, v_n)$

$$\langle x, x' \rangle_{\mathcal{H}} = \sum_{n=1}^{N} \sum_{n=1}^{N} \alpha_n \alpha'_n k(v_n, v'_n) = \boldsymbol{\alpha}'^T \boldsymbol{K} \boldsymbol{\alpha}$$

RKHS-based function estimator can be used to reconstruct signals

$$\hat{\boldsymbol{x}} = \boldsymbol{K}\boldsymbol{\alpha}$$

promotes smoothness

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \mathcal{L}(\boldsymbol{y}, \boldsymbol{\Phi}\boldsymbol{K}\boldsymbol{\alpha}) + \mu \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha}$$

Or, equivalently

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x} \in \mathcal{H}} \mathcal{L}(\boldsymbol{y}, \boldsymbol{\Phi}\boldsymbol{x}) + \mu \boldsymbol{x}^T \boldsymbol{K}^\dagger \boldsymbol{x}$$

$$\boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha} = \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{K}^\dagger \boldsymbol{K} \boldsymbol{\alpha}$$

$\mathcal{L}(\cdot)$ is a loss function

Kernel-based reconstruction – ridge regression

- Parameterization via *representer theorem*

$$\hat{\boldsymbol{x}} = \boldsymbol{K}\boldsymbol{\alpha} = \boldsymbol{K}\boldsymbol{\Phi}^T\bar{\boldsymbol{\alpha}} \qquad \bar{\boldsymbol{\alpha}} \in \mathbb{R}^K$$

Terms corresponding to unobserved vertices play no role in kernel expansion

$$\hat{\bar{\boldsymbol{\alpha}}} = \arg\min_{\bar{\boldsymbol{\alpha}}\in\mathbb{R}^K} \mathcal{L}(\boldsymbol{y}, \bar{\boldsymbol{K}}\bar{\boldsymbol{\alpha}}) + \mu\bar{\boldsymbol{\alpha}}^T\bar{\boldsymbol{K}}\bar{\boldsymbol{\alpha}} \qquad \bar{\boldsymbol{K}} = \boldsymbol{\Phi}\boldsymbol{K}\boldsymbol{\Phi}^T$$

- Kernel ridge regression

$$\begin{aligned}\hat{\bar{\boldsymbol{\alpha}}} &= \arg\min_{\bar{\boldsymbol{\alpha}}\in\mathbb{R}^K} \frac{1}{K}\|\boldsymbol{y} - \bar{\boldsymbol{K}}\bar{\boldsymbol{\alpha}}\|^2 + \mu\bar{\boldsymbol{\alpha}}^T\bar{\boldsymbol{K}}\bar{\boldsymbol{\alpha}} \\ &= (\bar{\boldsymbol{K}} + \mu K\mathbf{I})^{-1}\boldsymbol{y}\end{aligned}$$

$$\hat{\boldsymbol{x}} = \boldsymbol{K}\boldsymbol{\Phi}^T(\bar{\boldsymbol{K}} + \mu K\mathbf{I})^{-1}\boldsymbol{y}$$

- D. Romero, M. Ma, and G.B. Giannakis. Kernel-Based Reconstruction of Graph Signals, IEEE TSP, vol. 65, no. 3, pp. 764–778, Feb 2017.

Kernel-based reconstruction

Choice of kernels

- Graph bandlimited kernels

$$\boldsymbol{K}\boldsymbol{\Phi}^T = \boldsymbol{U}_{\text{BL}} \qquad \boldsymbol{x} = \boldsymbol{U}_{\text{BL}}\tilde{\boldsymbol{x}}_f$$

- Other topology-based kernel (promotes smooth signal estimates)

$$\boldsymbol{K} = r^{\dagger}(\boldsymbol{L}) = \boldsymbol{U}r^{\dagger}(\boldsymbol{\Lambda})\boldsymbol{U}^T$$

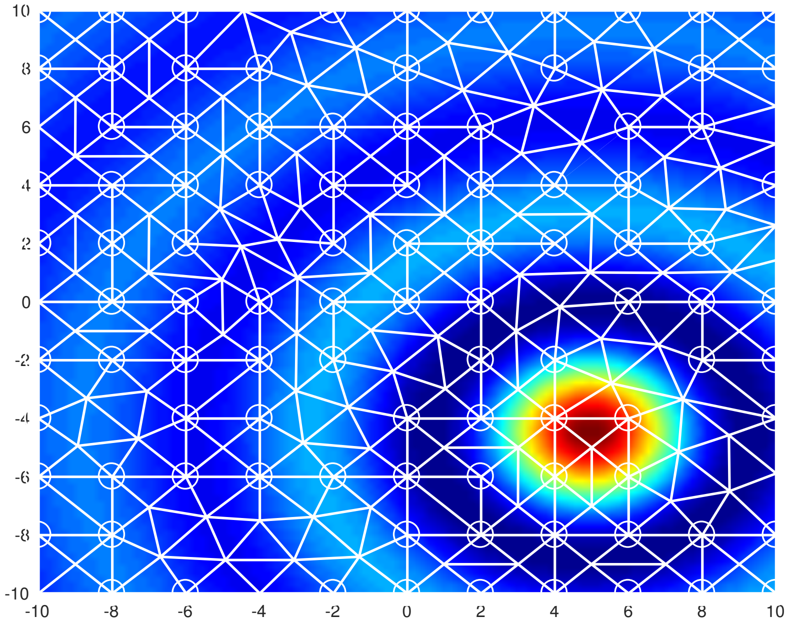
$r : \mathbb{R} \rightarrow \mathbb{R}_+$

Diffusion kernel: $r(\lambda) = exp\{\sigma^2\lambda/2\}$

p-step random walk kernel: $r(\lambda) = (a-\lambda)^{-p}, a \geq 2$

Laplacian (regularization) kernel: $r(\lambda) = 1 + \sigma^2\lambda$

Numerical experiments



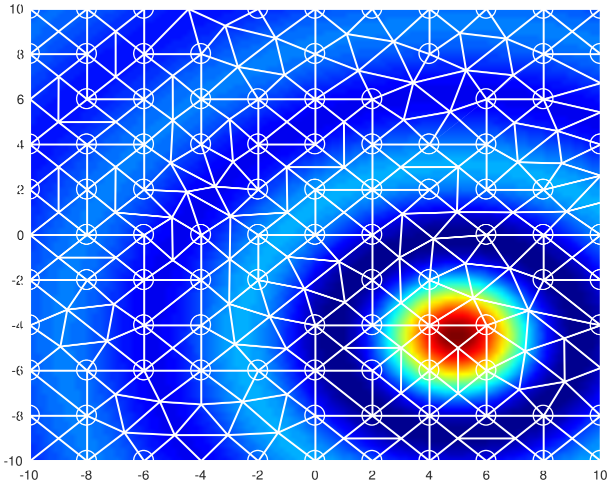
Wave field

- 2-D field estimation
- Rectangular domain of 10×10m
- Source located at coordinates $(x, y) = (5, -4.5)$
- Noise covariance $\mathbf{\Sigma} = \text{Toeplitz}\{1, \rho, \ldots, \rho^{N-1}\}$.
- Gaussian radial basis kernel with $\sigma = 0.8$.

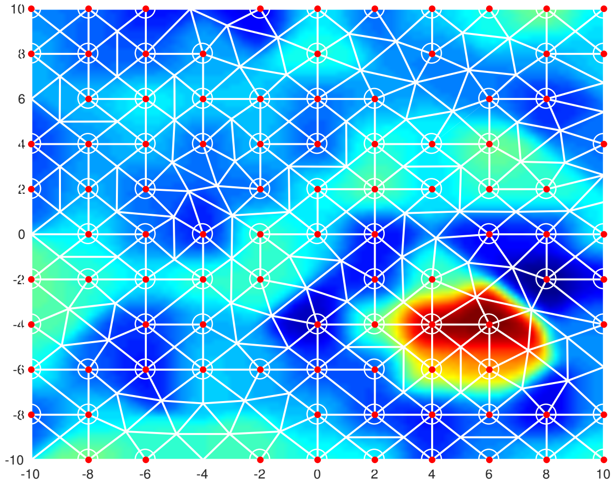
[Coutino-Chepuri-Leus-2018]

- M. Coutino, S.P. Chepuri and G. Leus. Subset Selection for Kernel-based Reconstruction. In Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2018), Calgary, Canada, April 2018.

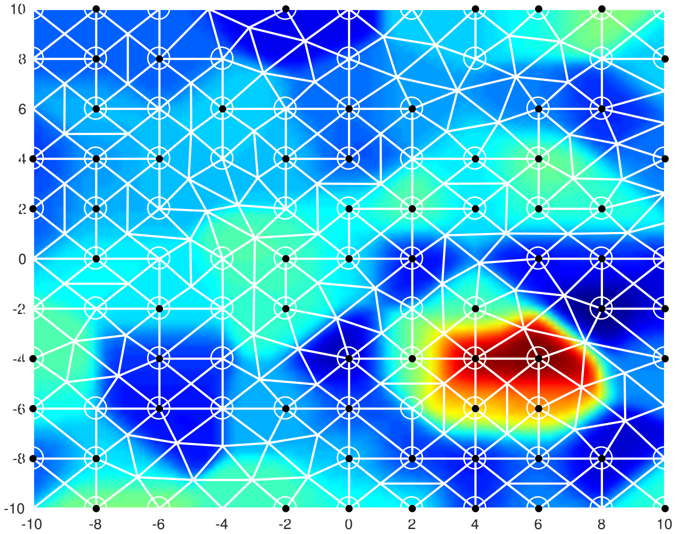
Numerical experiments



Ground truth

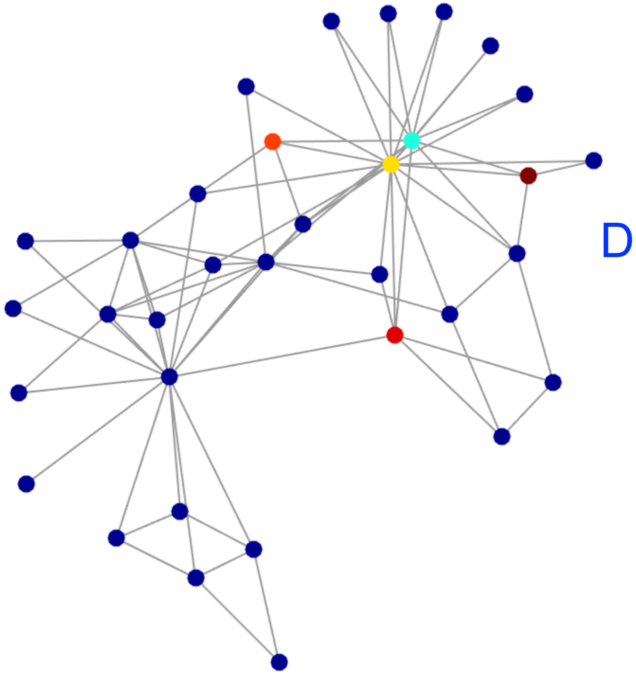


No subsampling (N=97)

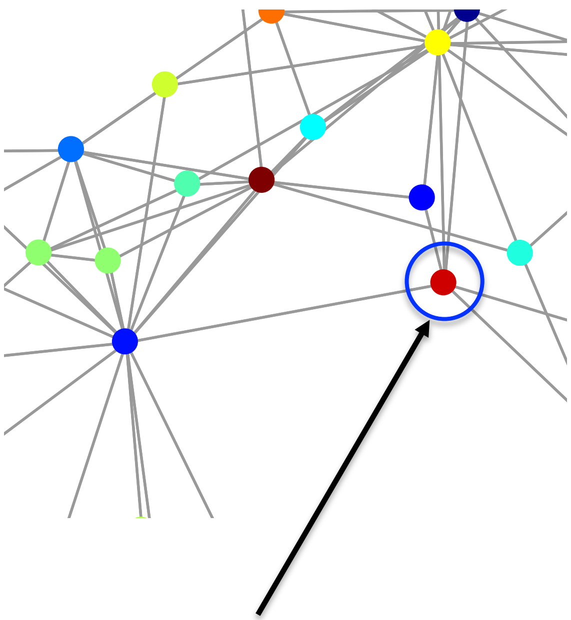


Measured 67 out of 97 mesh points

Diffusion processes on networks



Diffusion on networks



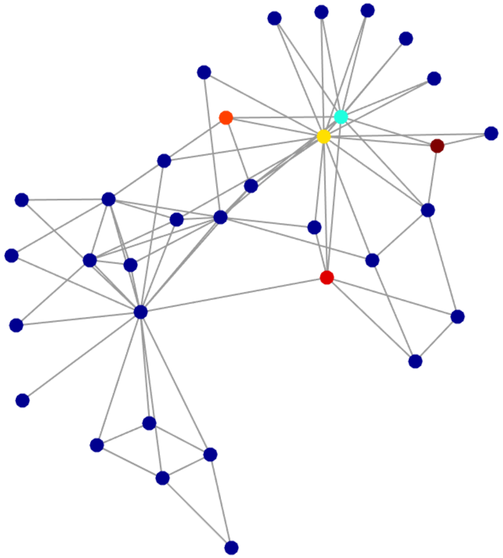
Can we reconstruct a graph signal from observations at a single node?

Linear dynamics on networks

- Information flow to a node from its neighbors

$$\underbrace{\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{S}\boldsymbol{x}_{k-1} + \boldsymbol{x}u_{k-1} \\ y_k &= \boldsymbol{e}_i^T \boldsymbol{x}_k \end{aligned}}_{\text{sample } node\ i}$$

Linear network dynamics



$\boldsymbol{x}_{-1} = 0$ and $\boldsymbol{x}_0 = \boldsymbol{x}$

$u_{k-1} = \delta[k]$ (Kronecker delta)

$\boldsymbol{e}_i$ is the ith column of the identity matrix

- Given observations $\boldsymbol{y} = \{y_0, \dots, y_{K-1}\}$ estimate $\boldsymbol{x}$

K is the number of shifts applied

Linear dynamics on networks

- At the observed node

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{e}_i^T \\ \boldsymbol{e}_i^T \boldsymbol{S} \\ \vdots \\ \boldsymbol{e}_i^T \boldsymbol{S}^{K-1} \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} \boldsymbol{e}_i^T \\ \boldsymbol{e}_i^T \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U}^H \\ \vdots \\ \boldsymbol{e}_i^T \boldsymbol{U} \boldsymbol{\Lambda}^{K-1} \boldsymbol{U}^H \end{bmatrix} \boldsymbol{x}$$

$$= \boldsymbol{V} \text{diag}[\underline{\boldsymbol{u}}] \boldsymbol{U}^H \boldsymbol{x} = \boldsymbol{V} \text{diag}[\underline{\boldsymbol{u}}] \boldsymbol{x}_f$$

Spectral response

$\underline{\boldsymbol{u}} = \boldsymbol{e}_i^T \boldsymbol{U}$ and $[\boldsymbol{V}]_{i,j} = \lambda_j^{i-1}$ (Vandermonde)

[Marques et al.-2016]

- A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, “Sampling of graph signals with successive local aggregations,” TSP 2016.

Linear dynamics on networks

Recall bandlimitedness:

- Suppose the support of the sparse $\boldsymbol{x}_f$ is known

$$\boldsymbol{x} = \boldsymbol{U}\boldsymbol{x}_f = \left[\ \boldsymbol{U}_{\text{BL}}\ \middle|\ \star\ \right] \left[\begin{array}{c} \tilde{\boldsymbol{x}}_f \\ \hline \boldsymbol{0} \end{array}\right] \quad \Leftrightarrow \quad \boldsymbol{x} = \boldsymbol{U}_{\text{BL}}\tilde{\boldsymbol{x}}_f$$

- The observations at *node i* will then be

K equations in L unknowns

$$\boldsymbol{y} = \boldsymbol{V}\text{diag}[\underline{\boldsymbol{u}}]\boldsymbol{x}_f = \boldsymbol{V}\text{diag}[\underline{\boldsymbol{u}}]\boldsymbol{E}_L\tilde{\boldsymbol{x}}_f = \boldsymbol{V}_{\text{BL}}\tilde{\boldsymbol{x}}_f$$

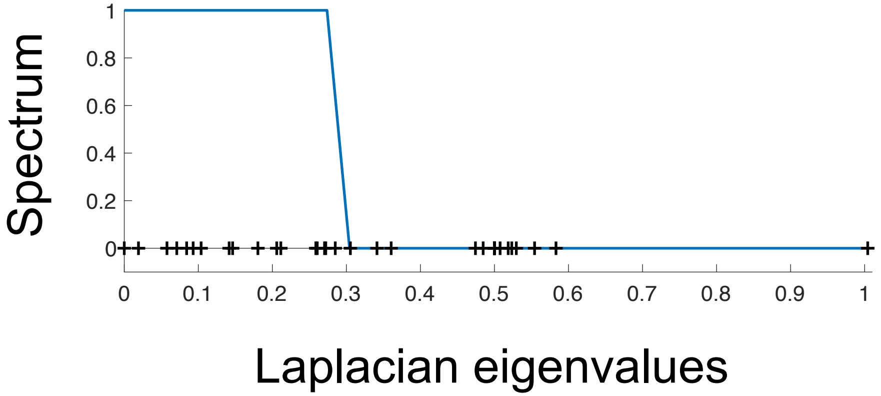
$$\boldsymbol{E}_L = [\boldsymbol{e}_1, \cdots, \boldsymbol{e}_L]$$

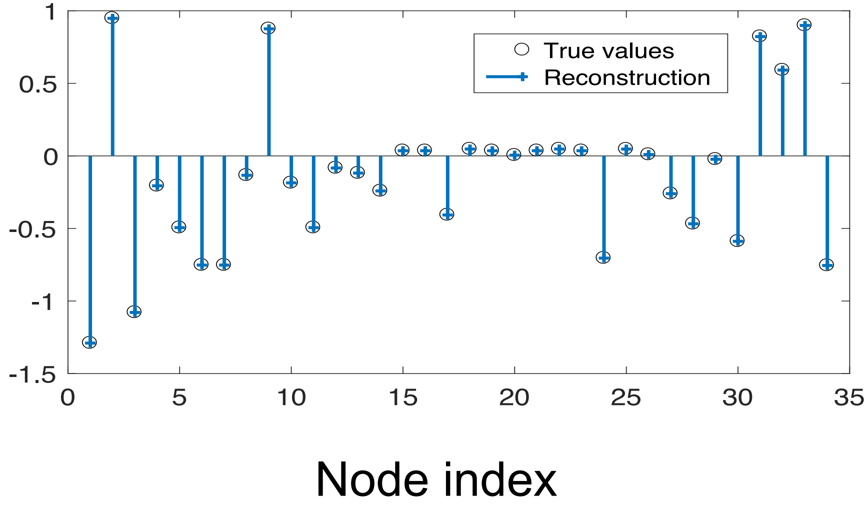
# of shifts

- If the matrix $\boldsymbol{V}_{\text{BL}}$ has full column rank, which requires $K \geq L$:

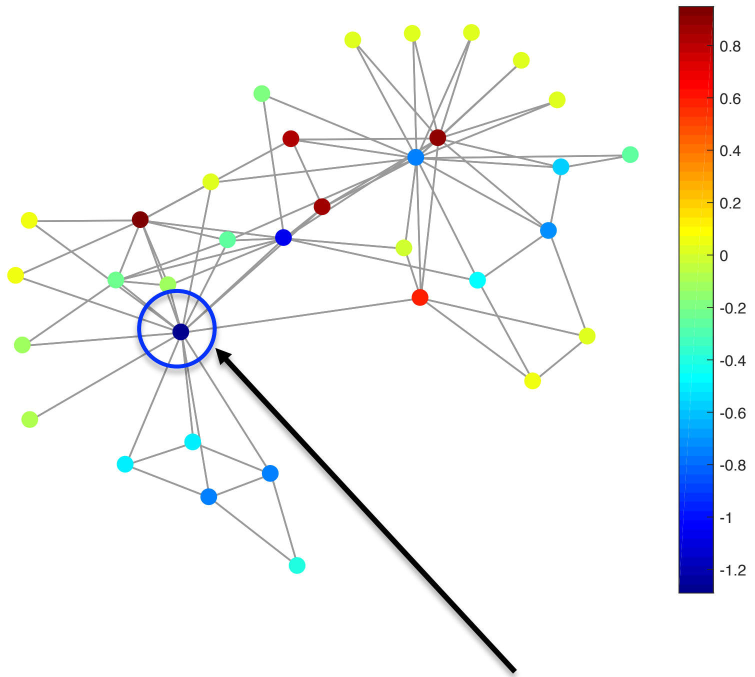
Least squares solution: $\widehat{\tilde{\boldsymbol{x}}}_f = \boldsymbol{V}_{\text{BL}}^{\dagger}\boldsymbol{y}$

Numerical experiments





Karate club network



Observed node for *K* shifts

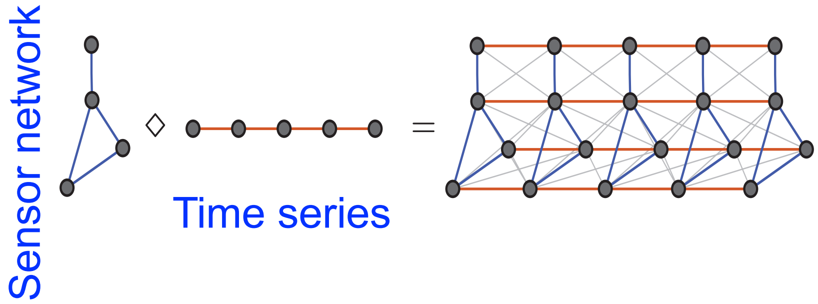
- ➢ Although reconstruction possible by observing a single node, system gets quickly ill conditioned (very sensitive to noise).
- ➢ Combining observations from a few more nodes might improve conditioning

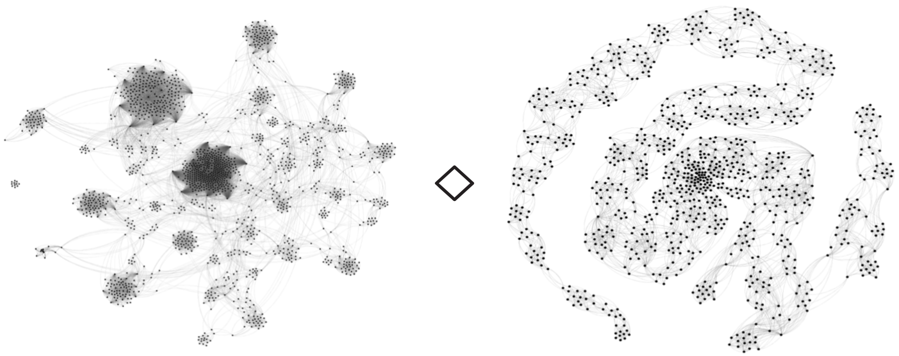
Product Graph Sampling

- G. Ortiz-Jiménez, M. Coutino, S.P. Chepuri, and G. Leus. Sampling and Reconstruction of Signals on Product Graphs. *GlobalSIP 2018*, Anaheim, USA. (available on arXiv:1807.00145).

- G. Ortiz-Jiménez, M. Coutino, S.P. Chepuri, and G. Leus. Sparse Sampling for Inverse Problems with Tensors. *IEEE TSP (under review)*, June 2018. (available as arXiv:1806.10976).

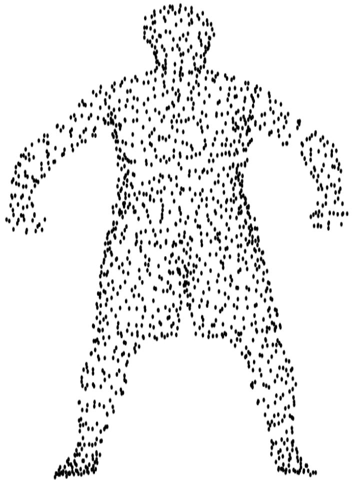
Sparse sampling on multigraph domains



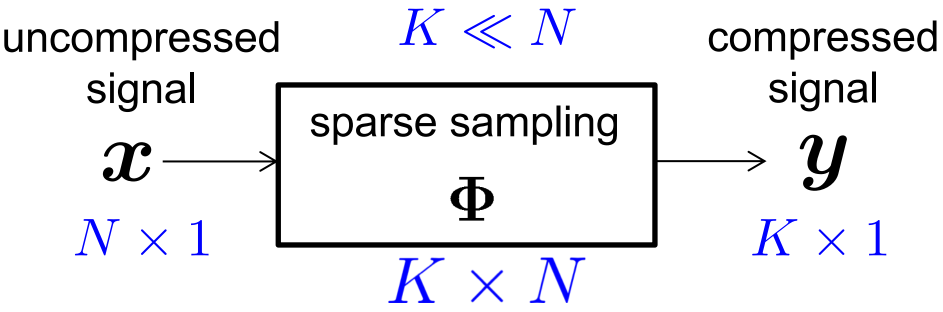


Movie graph

Social network



Dynamic 3D point cloud



uncompressed signal $\boldsymbol{x}$ $N \times 1$

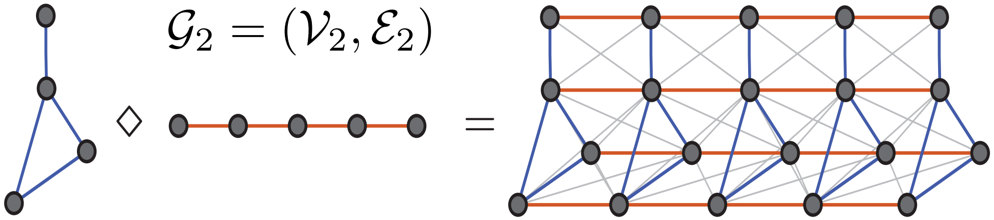
$K \ll N$

sparse sampling $\boldsymbol{\Phi}$ $K \times N$

compressed signal $\boldsymbol{y}$ $K \times 1$

Given y estimate x

Product graphs



N_2 nodes

$\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$

$\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ $\diamond$ $\mathcal{G}_2$ $=$ $\mathcal{G}_\diamond = (\mathcal{V}_1 \times \mathcal{V}_2, \mathcal{E}_\diamond)$

N_1 nodes

$N = N_1 N_2$ nodes

- Cartesian product (colored edges)
- Kronecker product (gray edges)
- Strong product (all edges)

- Let us represent $\mathcal{G}_1$ and $\mathcal{G}_2$ with the graph-shift operators

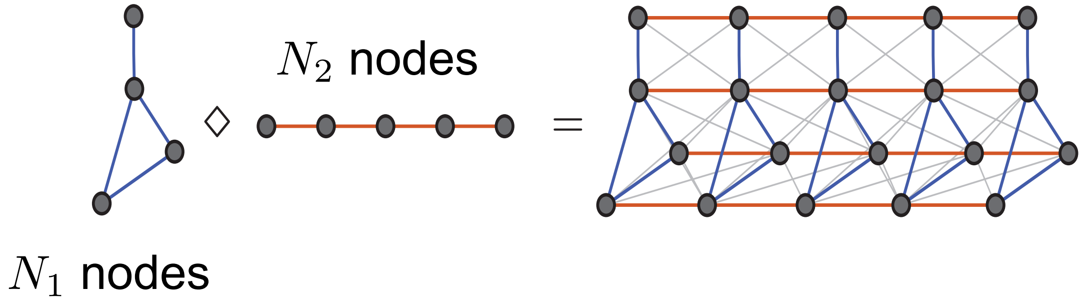
$$\boldsymbol{S}_1 = \boldsymbol{U}_1 \boldsymbol{\Lambda}_1 \boldsymbol{U}_1^H \in \mathbb{R}^{N_1 \times N_1} \quad \text{and} \quad \boldsymbol{S}_2 = \boldsymbol{U}_2 \boldsymbol{\Lambda}_2 \boldsymbol{U}_2^H \in \mathbb{R}^{N_2 \times N_2}$$

- The product graph $\mathcal{G}_\diamond$ has the graph-shift operator

$$\boldsymbol{S}_\diamond = (\boldsymbol{U}_1 \otimes \boldsymbol{U}_2) \boldsymbol{\Lambda}_\diamond (\boldsymbol{U}_1 \otimes \boldsymbol{U}_2)^H \in \mathbb{R}^{N \times N}$$

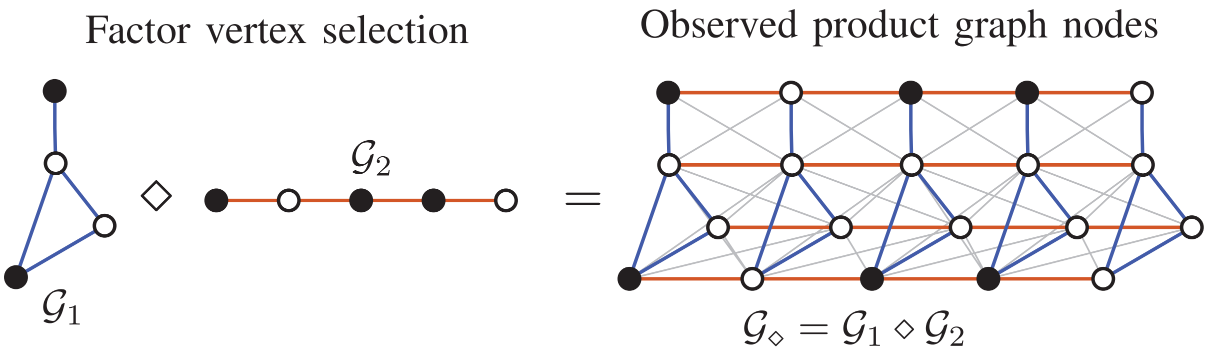
$\boldsymbol{\Lambda}_\diamond$ is a diagonal matrix that depends on $\mathcal{G}_1$ and $\mathcal{G}_2$, and the type of product

Product graph signals: The sampling problem



product graph signal

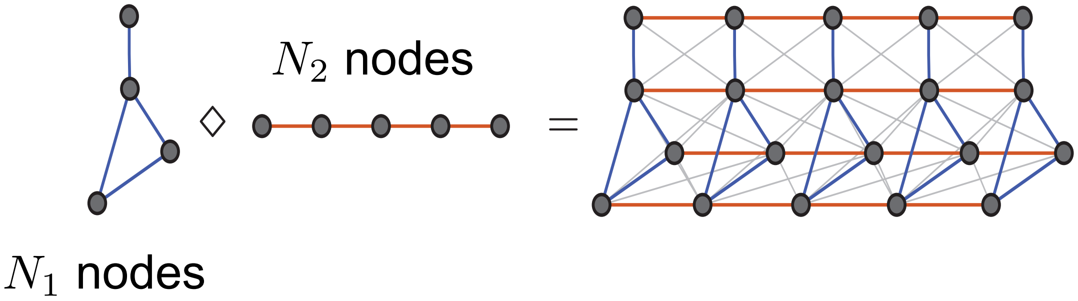
$\boldsymbol{X} \in \mathbb{R}^{N_1 \times N_2}$



uncompressed signal $\boldsymbol{X}$ ($N_1 \times N_2$) → **sparse sampling $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$** → compressed signal $\boldsymbol{Y} = \boldsymbol{\Phi}_1 \boldsymbol{X} \boldsymbol{\Phi}_2^T$ ($K_1 \times K_2$)

Given $\boldsymbol{Y}$ estimate $\boldsymbol{X}$

Product graph signal



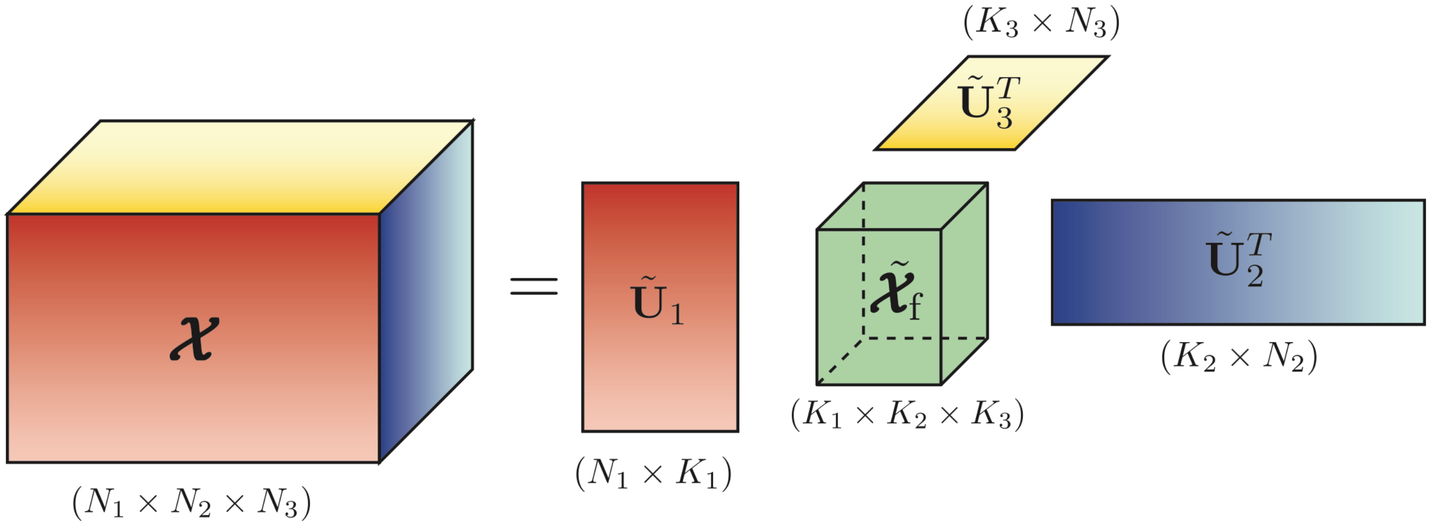
product graph signal
$\boldsymbol{X} \in \mathbb{R}^{N_1 \times N_2}$

- ➢ Product graph signal $\boldsymbol{X}$ may be decomposed w.r.t. $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ as

$$\boldsymbol{X} = \boldsymbol{U}_1 \boldsymbol{X}_f \boldsymbol{U}_2^T \quad \Leftrightarrow \quad \boldsymbol{x} = (\boldsymbol{U}_1 \otimes \boldsymbol{U}_2)\boldsymbol{x}_f$$

(synthesis)

Multilinear extension



$$\boldsymbol{\mathcal{X}} = \tilde{\boldsymbol{\mathcal{X}}}_\mathrm{f} \bullet_1 \tilde{\mathbf{U}}_1 \bullet_2 \cdots \bullet_R \tilde{\mathbf{U}}_R \Longleftrightarrow \mathbf{x} = (\tilde{\mathbf{U}}_1 \otimes \cdots \otimes \tilde{\mathbf{U}}_R)\tilde{\mathbf{x}}_\mathrm{f}$$

More generally, for Rth-order product graph, we have a graph (tensor) signal

$$\mathcal{X} = \mathcal{X}_f \bullet_1 \boldsymbol{U}_1 \bullet_2 \boldsymbol{U}_2 \cdots \bullet \boldsymbol{U}_R \quad \Leftrightarrow \quad \boldsymbol{x} = (\boldsymbol{U}_1 \otimes \boldsymbol{U}_2 \cdots \otimes \boldsymbol{U}_R)\boldsymbol{x}_f$$

$$\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \cdots \times N_R}$$

- G. Ortiz-Jiménez, M. Coutino, S.P. Chepuri, and G. Leus. Sparse Sampling for Inverse Problems with Tensors. IEEE TSP (under review), June 2018. (available as arXiv:1806.10976).

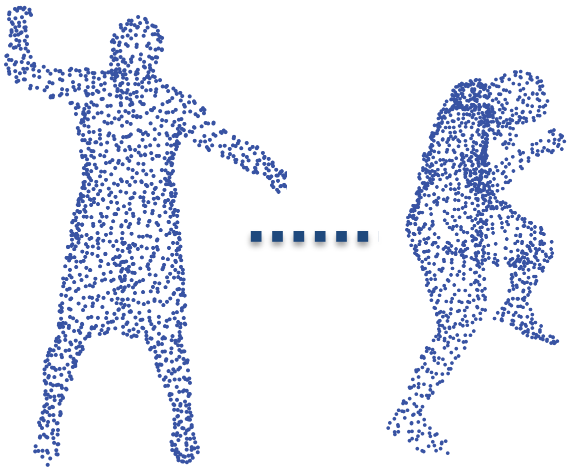
Bandlimited product graph signals

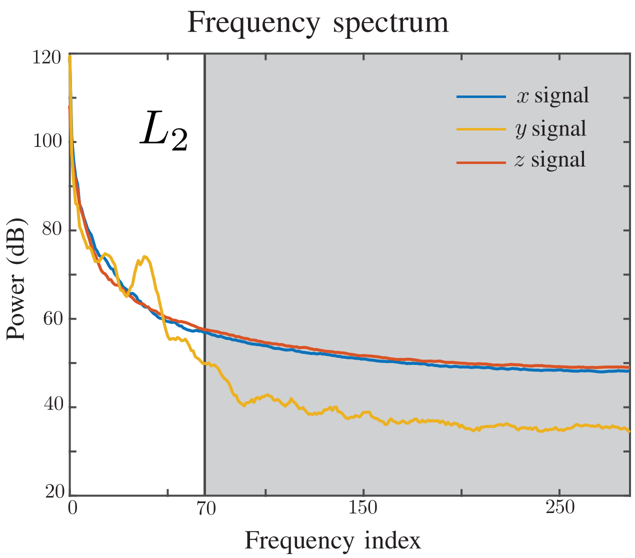
- Suppose the support of the sparse $\boldsymbol{x}_f$ is known

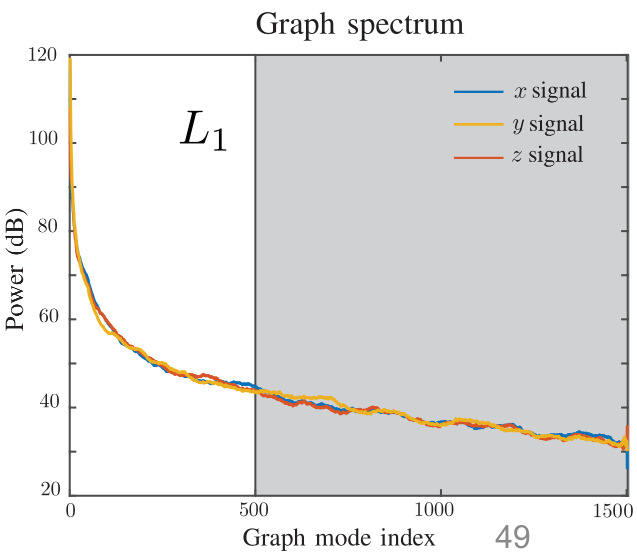
$$\boldsymbol{X} = \boldsymbol{U}_1 \boldsymbol{X}_f \boldsymbol{U}_2^T = \left[\ \overset{N_1 \times L_1}{\tilde{\boldsymbol{U}}_1} \ \middle|\ \star\ \right] \left[\begin{array}{c|c} \tilde{\boldsymbol{X}}_f & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{0} \end{array}\right] \left[\begin{array}{c} \overset{L_2 \times N_2}{\tilde{\boldsymbol{U}}_2^T} \\ \hline \star \end{array}\right]$$

or

$$\boldsymbol{x} = (\boldsymbol{U}_1 \otimes \boldsymbol{U}_2)\boldsymbol{x}_f = \left[\ (\tilde{\boldsymbol{U}}_1 \otimes \tilde{\boldsymbol{U}}_2)\ \middle|\ \star\ \right] \left[\begin{array}{c} \tilde{\boldsymbol{x}}_f \\ \hline \boldsymbol{0} \end{array}\right]$$







Bandlimited product graph signals

- Suppose the support of the sparse $\boldsymbol{x}_f$ is known

$$\boldsymbol{X} = \boldsymbol{U}_1 \boldsymbol{X}_f \boldsymbol{U}_2^T = \left[\ \overset{N_1 \times L_1}{\tilde{\boldsymbol{U}}_1} \ \middle|\ \star\ \right] \left[\begin{array}{c|c} \tilde{\boldsymbol{X}}_f & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array}\right] \left[\begin{array}{c} \overset{L_2 \times N_2}{\tilde{\boldsymbol{U}}_2^T} \\ \hline \star \end{array}\right]$$

or

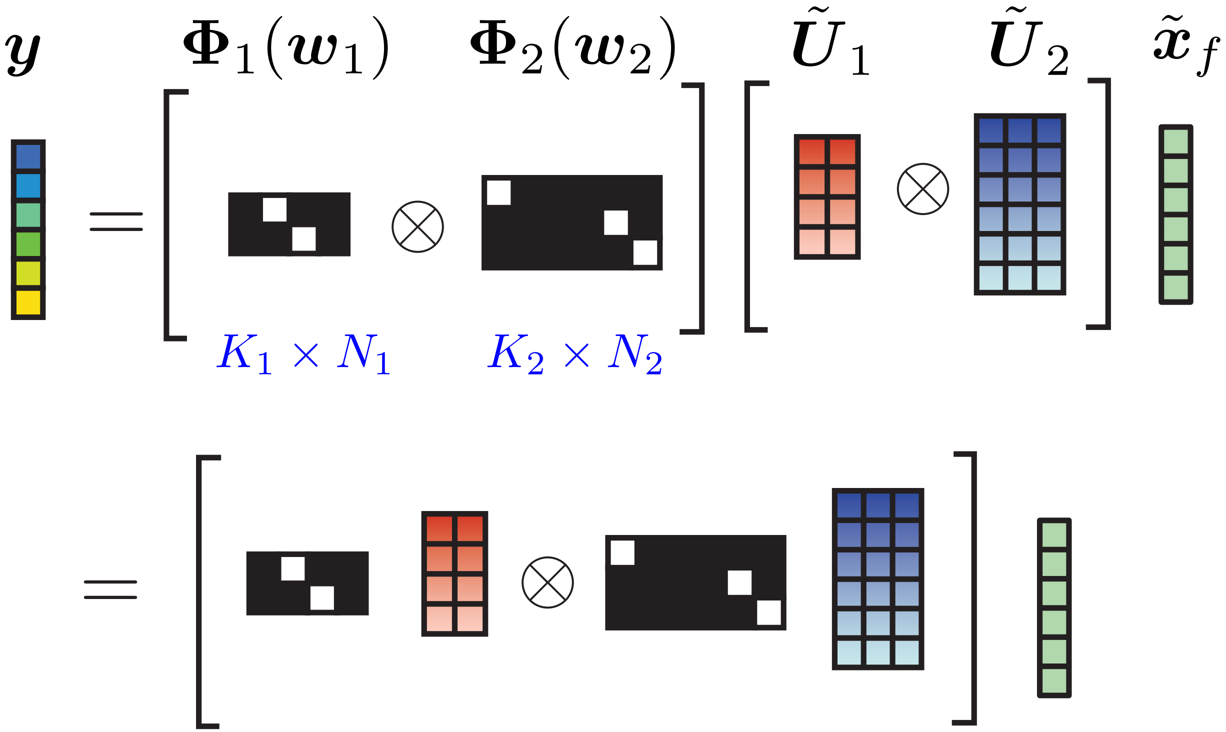
$$\boldsymbol{x} = (\boldsymbol{U}_1 \otimes \boldsymbol{U}_2)\boldsymbol{x}_f = \left[\ (\tilde{\boldsymbol{U}}_1 \otimes \tilde{\boldsymbol{U}}_2) \ \middle|\ \star\ \right] \left[\begin{array}{c} \tilde{\boldsymbol{x}}_f \\ \hline \mathbf{0} \end{array}\right]$$

- We can reconstruct the product graph signal from subsampled observations since

$$N_1 N_2 \gg L_1 L_2 \text{ and } \operatorname{rank}(\tilde{\boldsymbol{U}}_1 \otimes \tilde{\boldsymbol{U}}_2) = \operatorname{rank}(\tilde{\boldsymbol{U}}_1)\operatorname{rank}(\tilde{\boldsymbol{U}}_2)$$

Reconstruction with subspace prior

With sparse sampling, we get K_1K_2 equations in L_1L_2 unknowns



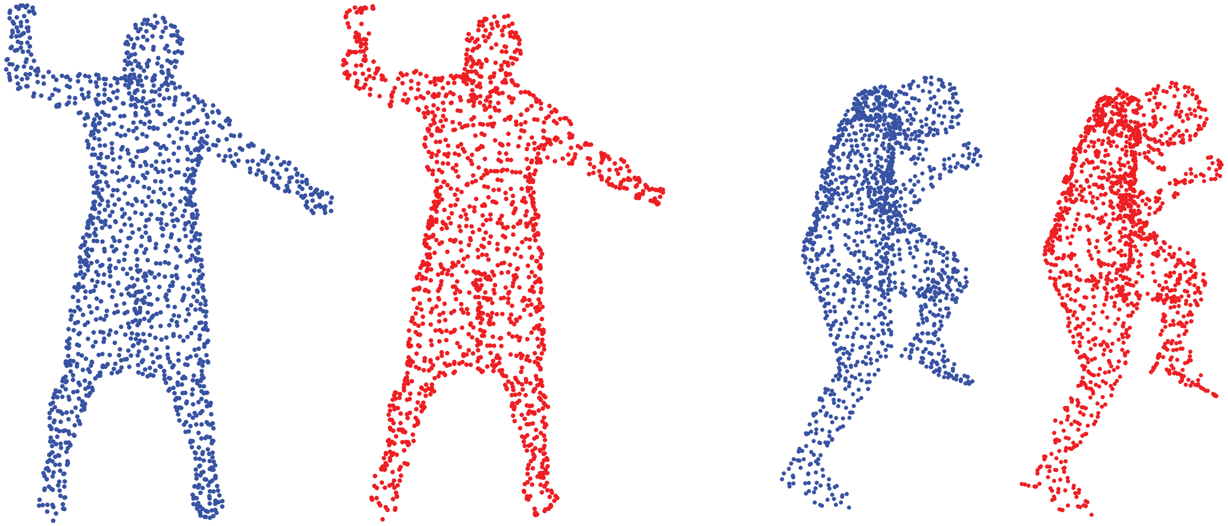
For unique reconstruction, we require $K_1 \geq L_1$ and $K_2 \geq L_2$

Least squares solution: $\hat{\tilde{\boldsymbol{x}}}_f = [(\boldsymbol{\Phi}_1\boldsymbol{U}_1)^\dagger \otimes (\boldsymbol{\Phi}_2\boldsymbol{U}_2)^\dagger]\boldsymbol{y}$

Design of $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$ is crucial for the least-squares solution to be unique

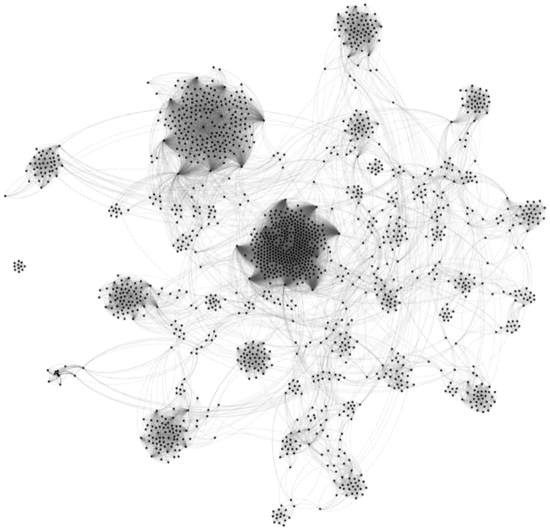
Numerical experiments – dynamic 3D point cloud

- 1502 markers, 573 frames. Product graph has 850000 vertices
- We sample 500 spatial points, and 70 time frames



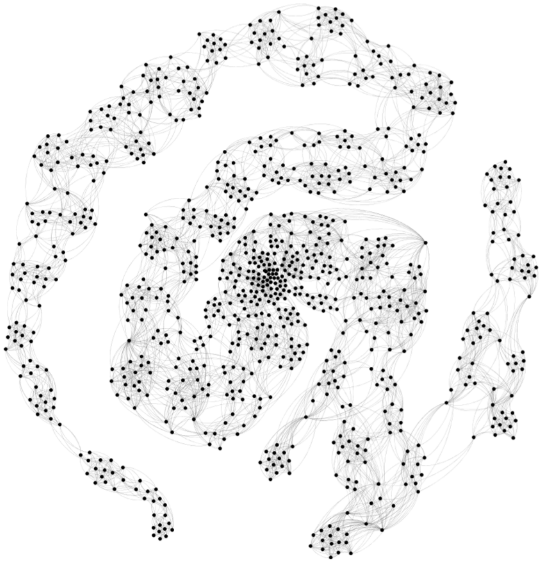
Numerical experiments – recommender system

MovieLens 100k dataset



Movie graph (1682 movies)

$\diamond$



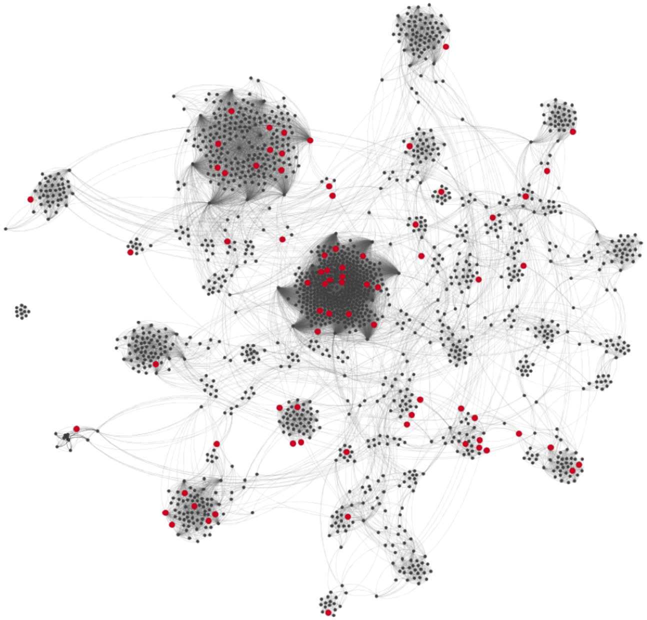
User graph (942 users)

- Product graph has about 1.6 million nodes
- Features used to build both the graphs (available with the dataset)
- Standard problem: Complete rating matrix using graph prior.
- *Active learning: Which users to probe for which movies?*

Numerical experiments – recommender system

MovieLens 100k dataset

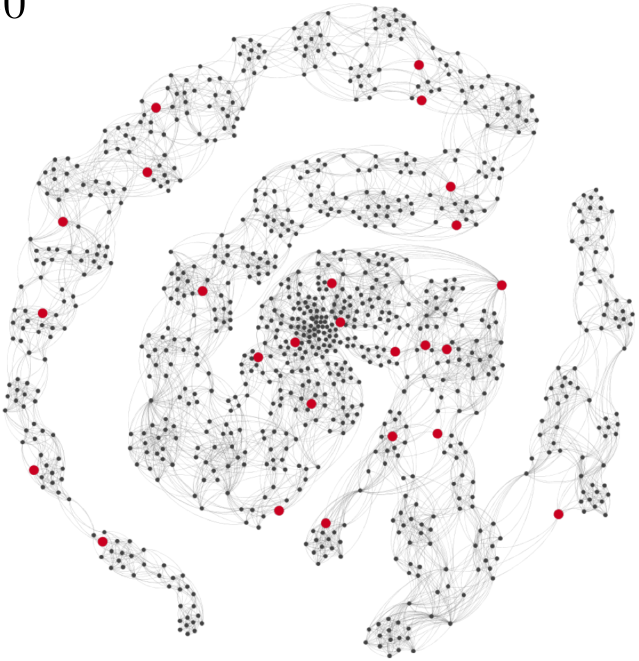
$$L_1 = L_2 = 20$$



Movie graph

75 movies sampled out of **1682 movies**

$\diamond$



User graph

25 users sampled out of **942 users**

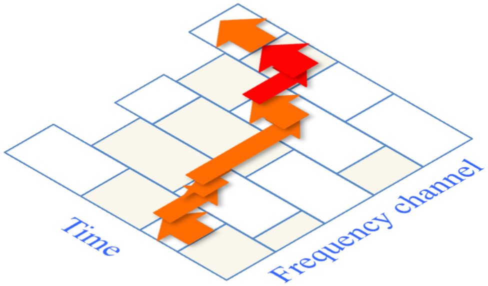
State-of-the-art matrix completion methods

Method	Number of samples	RMSE
GMC [26]	80,000	0.996
GRALS [27]	80,000	0.945
sRGCNN [29]	80,000	0.929
GC-MC [30]	80,000	**0.905**
Our method	**1,875**	0.9347

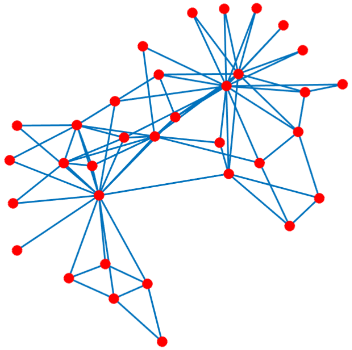
Graph Covariance Sampling

- S.P. Chepuri and G. Leus. Graph Sampling for Covariance Estimation. *IEEE Journ. on Sel. Topics in Sig. Proc. and IEEE Trans. on Sig. and Info. Proc. over Networks, joint special issue on Graph Signal Processing, July 2017.*

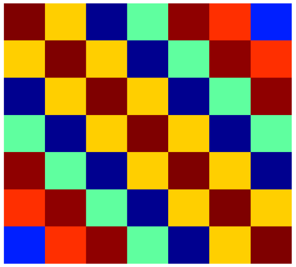
spatial spectrum



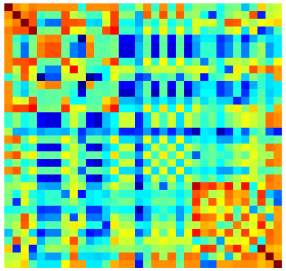
frequency spectrum



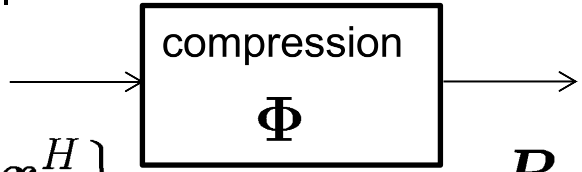
graph spectrum



structured (Toeplitz)



no apparent structure



uncompressed stationary signal

$K \ll N$

compressed signal

$$\boldsymbol{x} \longrightarrow \boxed{\text{compression } \boldsymbol{\Phi}} \longrightarrow \boldsymbol{y}$$

$$\boldsymbol{R}_{\boldsymbol{x}} = \mathrm{E}\left\{\boldsymbol{x}\boldsymbol{x}^H\right\} \qquad K \times N \qquad \boldsymbol{R}_{\boldsymbol{y}} = \mathrm{E}\left\{\boldsymbol{y}\boldsymbol{y}^H\right\}$$

$N \times N$

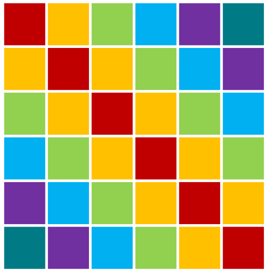
$K \times K$

Given $\boldsymbol{R}_{\boldsymbol{y}}$ or several realizations of $\boldsymbol{y}$ estimate $\boldsymbol{R}_{\boldsymbol{x}}$

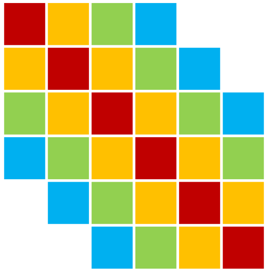
Compressive covariance sensing

$$\underbrace{\boldsymbol{r_y}}_{K^2 \times 1} = \mathrm{vec}(\boldsymbol{R_y}) = \mathrm{vec}(\boldsymbol{\Phi} \boldsymbol{R_x} \boldsymbol{\Phi}^T) = (\boldsymbol{\Phi} \otimes \boldsymbol{\Phi}) \underbrace{\mathrm{vec}(\boldsymbol{R_x})}_{N^2 \times 1}$$

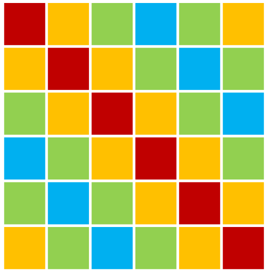
- Suppose the covariance matrix $\boldsymbol{R_x}$ has a linear structure



Toeplitz



Banded



Circulant

$$\boldsymbol{R_x}(\boldsymbol{\theta}) = \sum_{i=1}^{Q} \theta_i \boldsymbol{Q}_i \longrightarrow \boxed{\begin{array}{c}\text{compression}\\ \boldsymbol{\Phi}\end{array}} \longrightarrow \boldsymbol{R_y}(\boldsymbol{\theta}) = \sum_{i=1}^{Q} \theta_i \boldsymbol{\Phi} \boldsymbol{Q}_i \boldsymbol{\Phi}^T$$

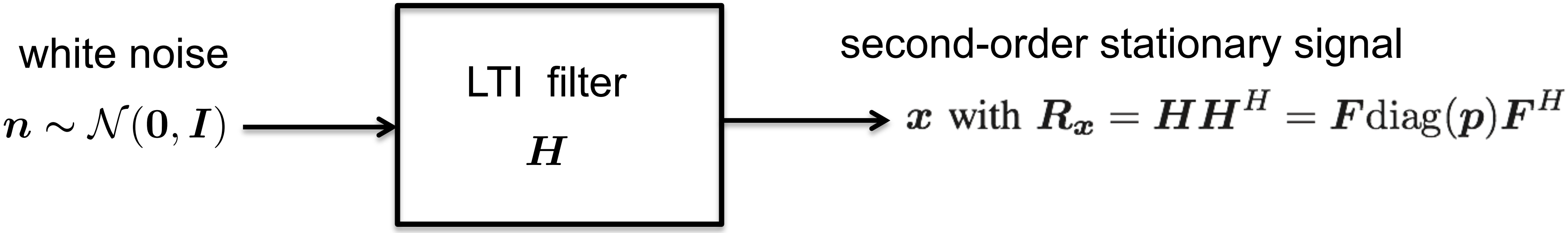
- If $K^2 > Q$: $\boldsymbol{r_y} = (\boldsymbol{\Phi} \otimes \boldsymbol{\Phi}) \boldsymbol{\Psi} \boldsymbol{\theta} \xrightarrow{\text{least squares}} \boldsymbol{\theta} = [(\boldsymbol{\Phi} \otimes \boldsymbol{\Phi}) \boldsymbol{\Psi}]^{\dagger} \boldsymbol{r_y}$

Design of $\boldsymbol{\Phi}$ crucial for the solution to be unique

Second-order stationarity in time

Filtering white noise:

- Signal is the output of an LTI filter excited with white noise



white noise

$\boldsymbol{n} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

LTI filter

$\boldsymbol{H}$

second-order stationary signal

$\boldsymbol{x}$ with $\boldsymbol{R}_{\boldsymbol{x}} = \boldsymbol{H}\boldsymbol{H}^H = \boldsymbol{F}\mathrm{diag}(\boldsymbol{p})\boldsymbol{F}^H$

- The covariance matrix is diagonalized by the Fourier matrix

$$\boldsymbol{R}_{\boldsymbol{x}} = \boldsymbol{F}\mathrm{diag}(\boldsymbol{p})\boldsymbol{F}^H$$

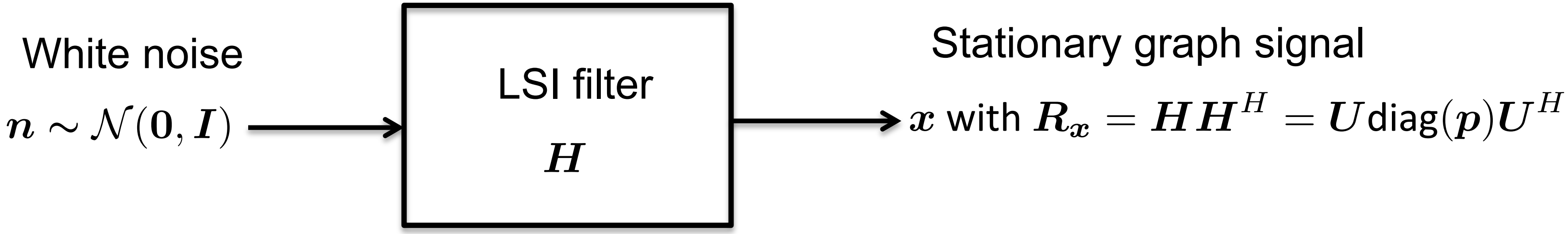
The process has power spectral density

$$\boldsymbol{p} = \mathrm{diag}(\boldsymbol{F}^H \boldsymbol{R}_{\boldsymbol{x}} \boldsymbol{F})$$

Stationary graph signals

Filtering white noise:

- A random graph signal $\boldsymbol{x} \in \mathbb{R}^N$ is second-order stationary:



White noise

$\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

LSI filter

$\boldsymbol{H}$

Stationary graph signal

$\boldsymbol{x}$ with $\boldsymbol{R}_{\boldsymbol{x}} = \boldsymbol{H}\boldsymbol{H}^H = \boldsymbol{U}\text{diag}(\boldsymbol{p})\boldsymbol{U}^H$

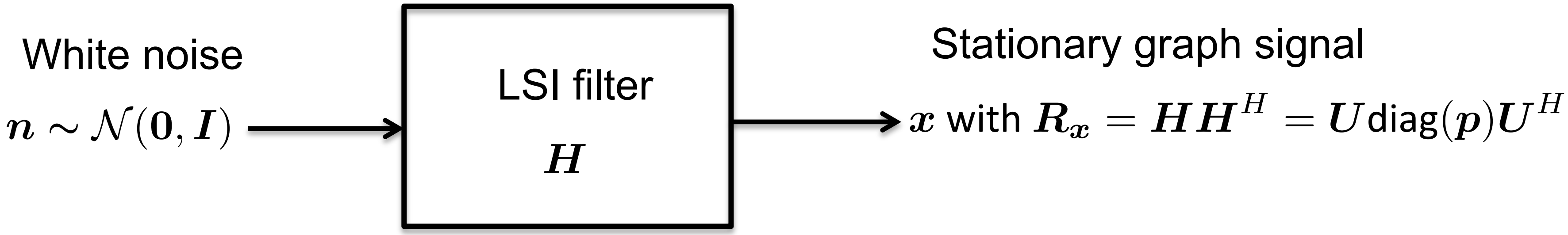
- The filter should be shift invariant $\boldsymbol{H}(\boldsymbol{S}\boldsymbol{x}) = \boldsymbol{S}(\boldsymbol{H}\boldsymbol{x}) \Leftrightarrow \boldsymbol{H} = \boldsymbol{U}\text{diag}(\boldsymbol{h}_f)\boldsymbol{U}^H$

- N. Perraudin and P. Vandergheynst, “Stationary signal processing on graphs,” *IEEE TSP, Jul. 2017.*
- A. Marques, S. Segarra, G. Leus, and A. Ribeiro, “Stationary graph processes and spectral estimation,” *IEEE TSP, Nov. 2017.*

Stationary graph signals

Filtering white noise:

- A random graph signal $\boldsymbol{x} \in \mathbb{R}^N$ is second-order stationary:



White noise

$\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

LSI filter

$\boldsymbol{H}$

Stationary graph signal

$\boldsymbol{x}$ with $\boldsymbol{R_x} = \boldsymbol{H}\boldsymbol{H}^H = \boldsymbol{U}\text{diag}(\boldsymbol{p})\boldsymbol{U}^H$

Simultaneous diagonalization:

$$\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^H \qquad \boldsymbol{R_x} = \boldsymbol{U}\text{diag}(\boldsymbol{p})\boldsymbol{U}^H$$

- The process has power spectral density

$$\boldsymbol{p} = \text{diag}(\boldsymbol{U}^H\boldsymbol{R_x}\boldsymbol{U})$$

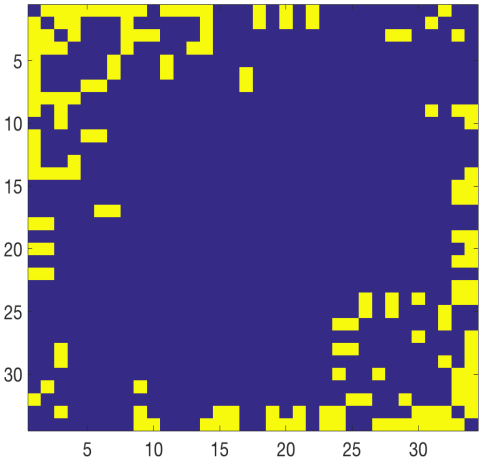
***Remark (second-order stationarity in time)*:**

$\boldsymbol{R_x}$ is a circulant matrix, which can be diagonalized by the DFT matrix

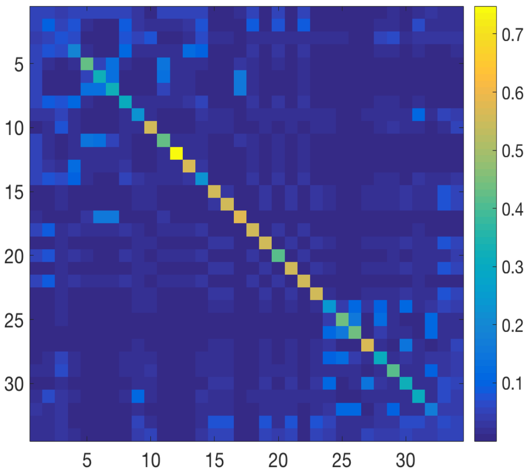
- N. Perraudin and P. Vandergheynst, “Stationary signal processing on graphs,” *IEEE TSP, Jul. 2017.*
- A. Marques, S. Segarra, G. Leus, and A. Ribeiro, “Stationary graph processes and spectral estimation,” *IEEE TSP, Nov. 2017.*

Stationary graph signals

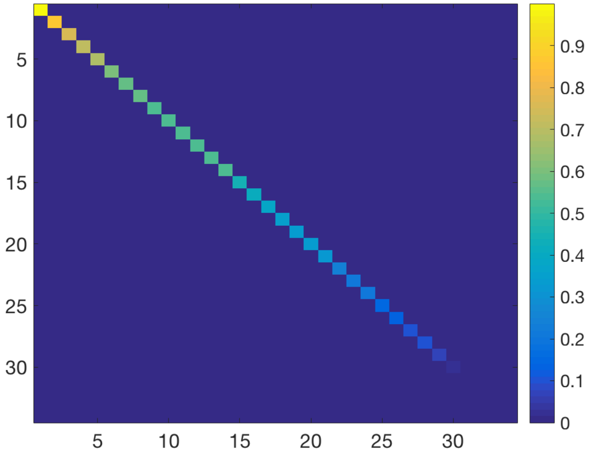
- Stationary process $\boldsymbol{x} \in \mathbb{R}^N$ on a graph shift $\boldsymbol{S}$



Adjacency matrix
(Karate club network)



Covariance matrix



Spectral domain
$\boldsymbol{U}^H \boldsymbol{R}_{\boldsymbol{x}} \boldsymbol{U}$

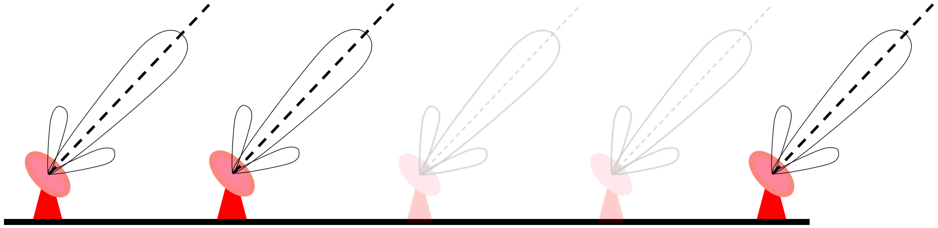
Power spectrum estimation is crucial for statistical inference
smoothing, prediction, deconvolution

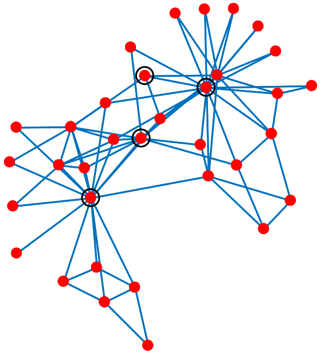
Power spectrum estimation

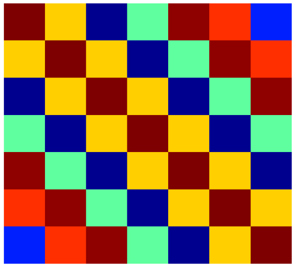
Estimate the power spectrum

a. by observing a reduced subset of nodes/sensors (i.e., subsample)

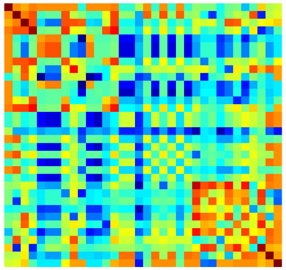
b. without using spectral priors (e.g., sparsity, bandlimited with known support)







structured (Toeplitz)



no apparent structure

Non-parametric method

- The covariance again admits a linear structure

$$\boldsymbol{R_x} = \boldsymbol{U}\mathrm{diag}(\boldsymbol{p})\boldsymbol{U}^H \qquad \boldsymbol{R_x} = \sum_{i=1}^{N} p_i \boldsymbol{u}_i \boldsymbol{u}_i^H = \sum_{i=1}^{N} p_i \boldsymbol{Q}_i$$

- After compression:

$$\boldsymbol{R_x} = \sum_{i=1}^{N} p_i \boldsymbol{Q}_i \longrightarrow \boxed{\begin{array}{c}\text{compression}\\ \boldsymbol{\Phi}\end{array}} \longrightarrow \boldsymbol{R_y} = \sum_{i=1}^{N} p_i \boldsymbol{\Phi}\boldsymbol{Q}_i\boldsymbol{\Phi}^T$$

- We have K^2 equations in N unknowns

vec$(\boldsymbol{A}\mathrm{diag}(\boldsymbol{d})\boldsymbol{B}) = (\boldsymbol{B}^T \circ \boldsymbol{A})\boldsymbol{d}$

$$\begin{aligned}\boldsymbol{r}_y = \mathrm{vec}(\boldsymbol{R}_y) &= (\boldsymbol{\Phi} \otimes \boldsymbol{\Phi})\mathrm{vec}(\boldsymbol{R}_x) \\ &= (\boldsymbol{\Phi} \otimes \boldsymbol{\Phi})(\boldsymbol{U} \circ \boldsymbol{U})\boldsymbol{p} \\ &= (\boldsymbol{\Phi} \otimes \boldsymbol{\Phi})\boldsymbol{\Psi}_{\mathrm{NP}}\boldsymbol{p}\end{aligned}$$

- If the matrix $(\boldsymbol{\Phi} \otimes \boldsymbol{\Phi})\boldsymbol{\Psi}_{\mathrm{NP}}$ has full column rank, which requires $K^2 \geq N$

$$\hat{\boldsymbol{p}} = [(\boldsymbol{\Phi} \otimes \boldsymbol{\Phi})\boldsymbol{\Psi}_{\mathrm{NP}}]^{\dagger}\boldsymbol{r_y}$$

Parametric method (moving average)

- Graph signal is a moving average graph process of order $L-1$

$$\boldsymbol{x} = \boldsymbol{H}(\boldsymbol{h})\boldsymbol{n} = \textstyle\sum_{l=0}^{L-1} h_l \boldsymbol{S}^l \boldsymbol{n} = \boldsymbol{U}\left(\textstyle\sum_{l=0}^{L-1} h_l \boldsymbol{\Lambda}^l\right)\boldsymbol{U}^H \boldsymbol{n}$$

with covariance matrix

$$\boldsymbol{R}_{\boldsymbol{x}} = \boldsymbol{H}(\boldsymbol{h})\boldsymbol{H}^H(\boldsymbol{h}) = \boldsymbol{U}\left(\textstyle\sum_{l=0}^{L-1} h_l \boldsymbol{\Lambda}^l\right)^2 \boldsymbol{U}^H$$

- We can express $\boldsymbol{R}_{\boldsymbol{x}}$ as a *matrix polynomial* of the *graph-shift* operator

$$\boldsymbol{R}_{\boldsymbol{x}}(\boldsymbol{b}) = \textstyle\sum_{k=0}^{Q-1} b_k \boldsymbol{S}^k$$

Covariance matching (*basis expansion*): $Q = \underbrace{\min\{2L-1, N\}}_{\text{degree of minimal polynomial of the \textit{graph-shift}}}$

For, $L=2$, $\boldsymbol{R}_{\boldsymbol{x}} = h_0^2\mathbf{I} + 2h_0h_1\boldsymbol{S} + h_1^2\boldsymbol{S}^2$

Parametric method (moving average)

- For a moving average graph process on an undirected graph we have

$$\boldsymbol{R_x} = \sum_{k=0}^{Q-1} b_k \boldsymbol{S}^k \qquad\qquad Q = \min\{2L-1, N\}$$

- After compression:

$$\boldsymbol{R_x} = \sum_{k=0}^{Q-1} b_k \boldsymbol{S}^k \longrightarrow \boxed{\begin{array}{c}\text{compression}\\ \boldsymbol{\Phi}\end{array}} \longrightarrow \boldsymbol{R_y} = \sum_{k=0}^{Q-1} b_k \boldsymbol{\Phi} \boldsymbol{S}^k \boldsymbol{\Phi}^T$$

- We have K^2 equations in Q unknowns

$$\begin{aligned} \boldsymbol{r}_y = \text{vec}(\boldsymbol{R}_y) &= (\boldsymbol{\Phi} \otimes \boldsymbol{\Phi})\text{vec}(\boldsymbol{R}_x) \\ &= (\boldsymbol{\Phi} \otimes \boldsymbol{\Phi})[\text{vec}(\boldsymbol{S}^0), \ldots, \text{vec}(\boldsymbol{S}^{Q-1})]\boldsymbol{b} \\ &= (\boldsymbol{\Phi} \otimes \boldsymbol{\Phi})\boldsymbol{\Psi}_{\text{MA}}\boldsymbol{b} \end{aligned}$$

- If the matrix $(\boldsymbol{\Phi} \otimes \boldsymbol{\Phi})\boldsymbol{\Psi}_{\text{MA}}$ has full column rank, which requires $K^2 \geq Q$

$$\hat{\boldsymbol{b}} = [(\boldsymbol{\Phi} \otimes \boldsymbol{\Phi})\boldsymbol{\Psi}_{\text{MA}}]^{\dagger}\boldsymbol{r_y}$$

Parametric approach (AR)

- For an autoregressive graph process we have (cf. Yule-Walker)

$$\boldsymbol{R_x} = \sum_{k=1}^{P} a_k \boldsymbol{S}^k \boldsymbol{R_x} + \boldsymbol{R_{nx}} \approx \sum_{k=1}^{P} a_k \boldsymbol{S}^k \boldsymbol{R_x}$$

- After compression:

$$\boldsymbol{R_x} \approx \sum_{k=1}^{P} a_k \boldsymbol{S}^k \boldsymbol{R_x} \longrightarrow \boxed{\begin{array}{c}\text{compression}\\ \boldsymbol{\Phi}\end{array}} \longrightarrow \boldsymbol{R_y} \approx \sum_{k=1}^{P} a_k \boldsymbol{\Phi} \boldsymbol{S}^k \boldsymbol{R_x} \boldsymbol{\Phi}^T$$

- We have K^2 equations in Q unknowns

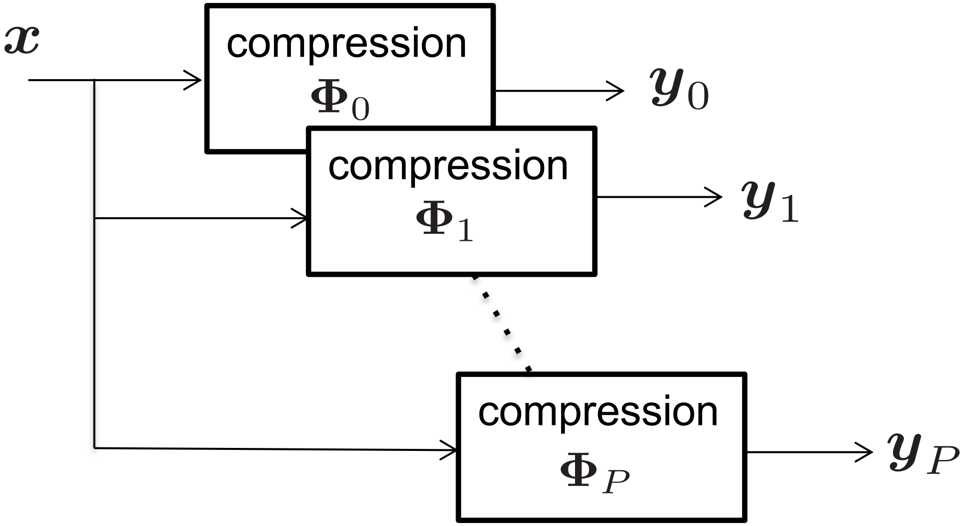
$$\begin{aligned} \boldsymbol{r}_y = \operatorname{vec}(\boldsymbol{R}_y) &= (\boldsymbol{\Phi} \otimes \boldsymbol{\Phi}) \operatorname{vec}(\boldsymbol{R}_x) \\ &= (\boldsymbol{\Phi} \otimes \boldsymbol{\Phi}) [\operatorname{vec}(\boldsymbol{S}\boldsymbol{R_x}), \ldots, \operatorname{vec}(\boldsymbol{S}^P \boldsymbol{R_x})] \boldsymbol{a} \\ &= (\boldsymbol{\Phi} \otimes \boldsymbol{\Phi}) \boldsymbol{\Psi}_{\mathrm{AR}} \boldsymbol{a} \end{aligned}$$

- If the matrix $(\boldsymbol{\Phi} \otimes \boldsymbol{\Phi}) \boldsymbol{\Psi}_{\mathrm{AR}}$ has full column rank, which requires $K^2 \geq P$

$$\hat{\boldsymbol{a}} = [(\boldsymbol{\Phi} \otimes \boldsymbol{\Phi}) \boldsymbol{\Psi}_{\mathrm{AR}}]^{\dagger} \boldsymbol{r_y}$$

Parametric Approach (AR)

- The system matrix $\boldsymbol{\Psi}_{\mathrm{AR}}$ depends on $\boldsymbol{R_x}$ and not only on $\boldsymbol{R_y}$
- Solution is to devise a new type of compression scheme
  - ✓ We sample K_0 nodes using $\boldsymbol{\Phi}_0$
  - ✓ We then sample a P-hop neighborhood of this set of nodes



$$\boldsymbol{y}_0 = \sum_{k=1}^{P} a_k \boldsymbol{\Phi}_0 \boldsymbol{S}^k \boldsymbol{x} + \boldsymbol{\Phi}_0 \boldsymbol{n}$$

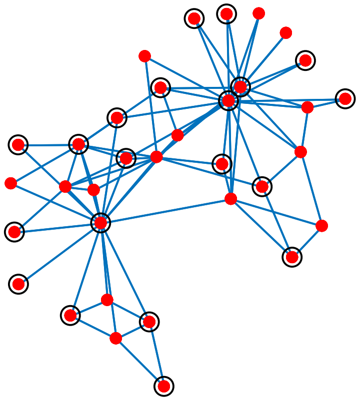
$$= \sum_{k=1}^{P} a_k \boldsymbol{\Phi}_0 \boldsymbol{S}^k \boldsymbol{\Phi}_k^T \boldsymbol{y}_k + \boldsymbol{\Phi}_0 \boldsymbol{n}$$

- In the time domain, this means we observe series of P consecutive samples

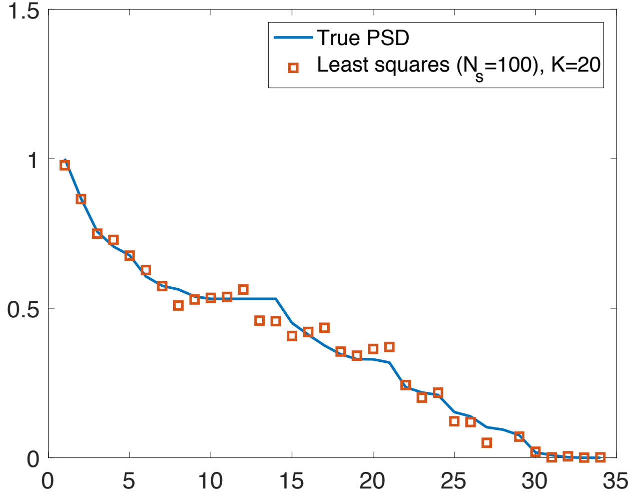
Illustration – Karate club network

Non-parametric approach

Sample 20 out of 34 nodes

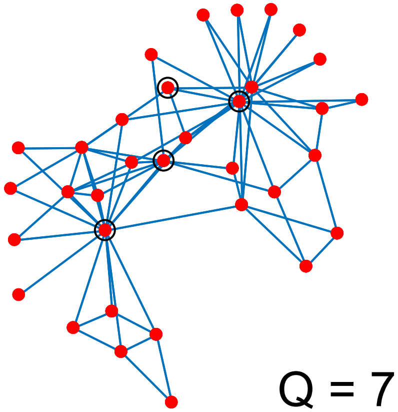


PSD estimation from subset of nodes



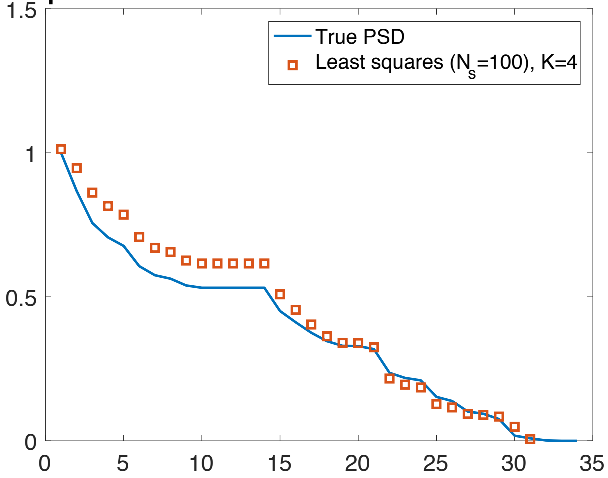
Parametric approach

Sample 4 out of 34 nodes



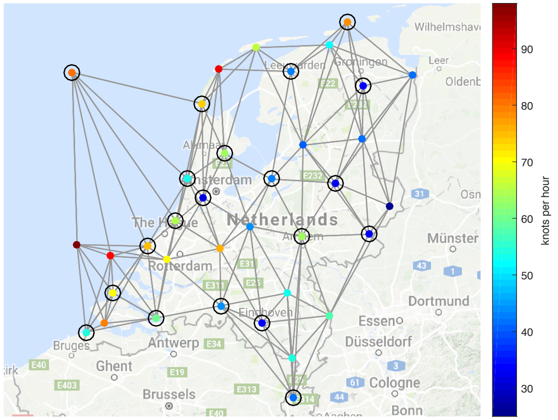
Q = 7

MA parametric PSD estimation from subset of nodes

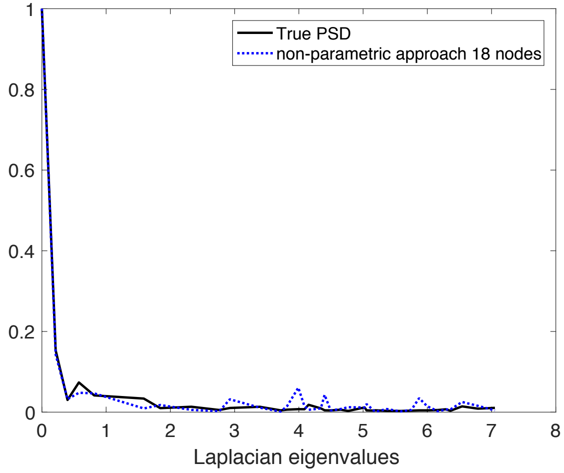


Wind speed dataset

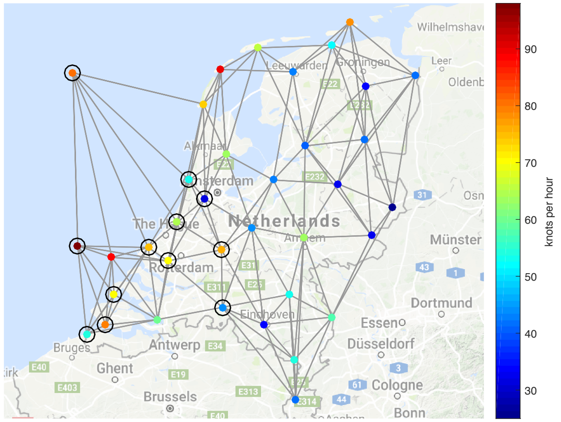
Non-parametric approach



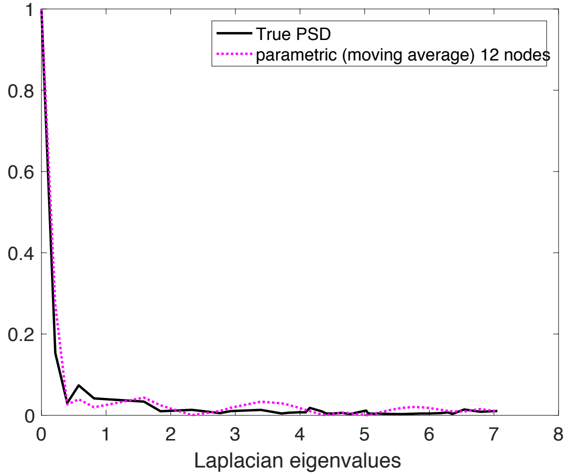
Sample 18 out of 36 stations



Moving average approach

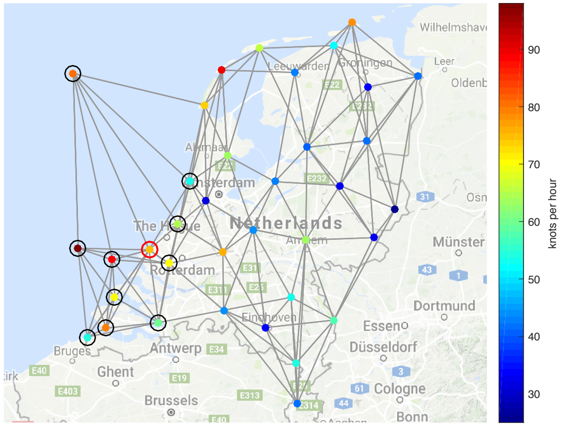


12 out of 36 stations

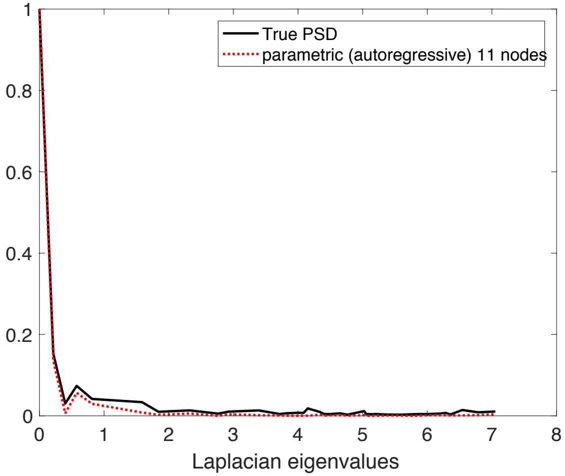


L=6 => Q = 11

Autoregressive approach



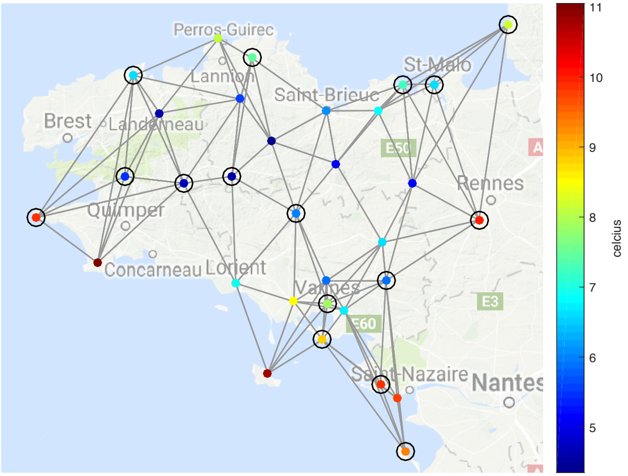
11 out of 36 stations



P = 1

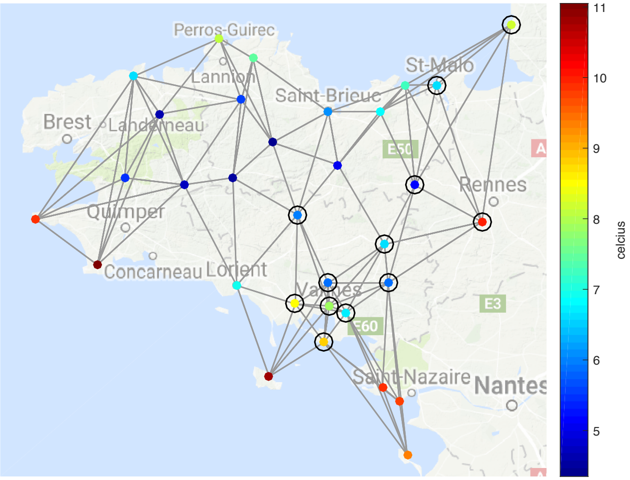
Temperature dataset

Non-parametric approach



Sample 16 out of 32 nodes

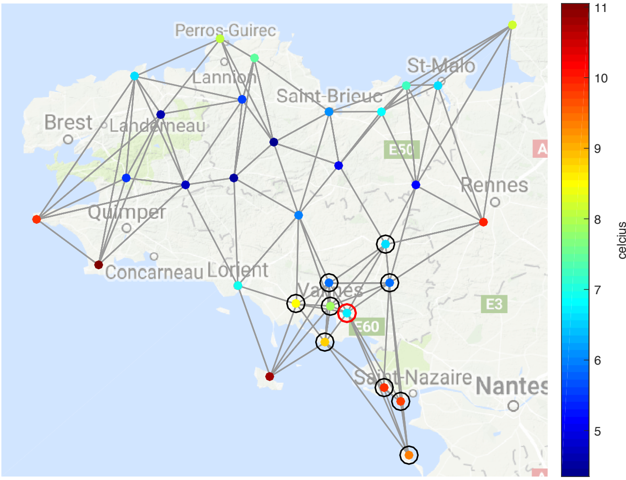
Moving average approach



12 out of 32 nodes

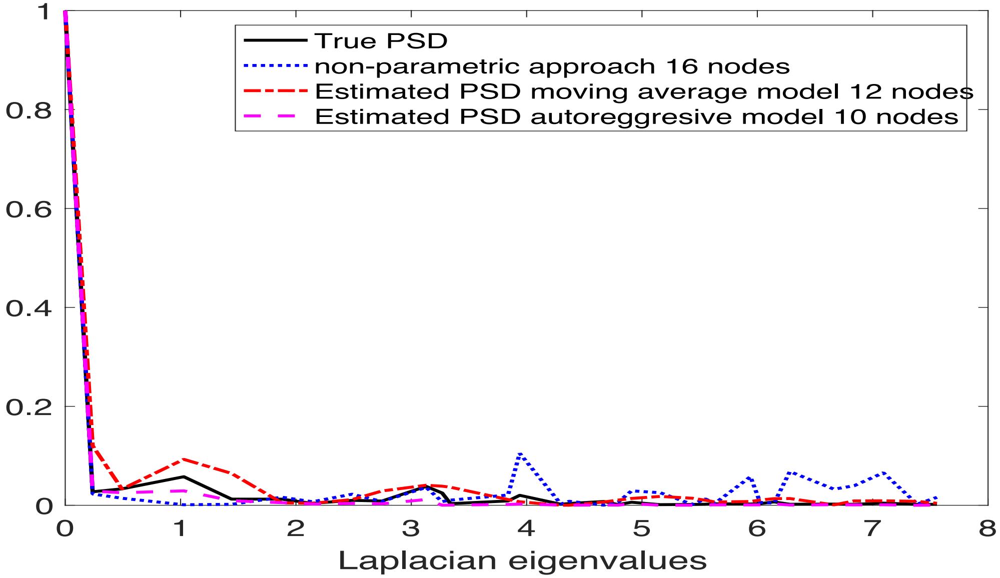
Q = 11

Autoregressive approach



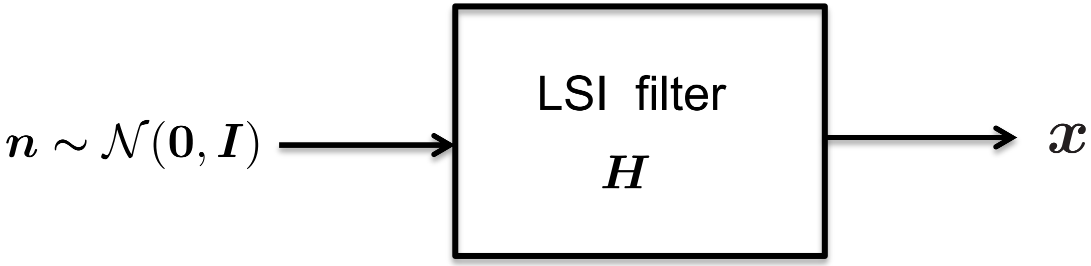
10 out of 32 nodes

P = 1



Generate digits

- Nearest neighbor graph built using digit 3 (16 x 16 pixels) from the USPS dataset.
- Graph signal (pixel intensity) is of length 256

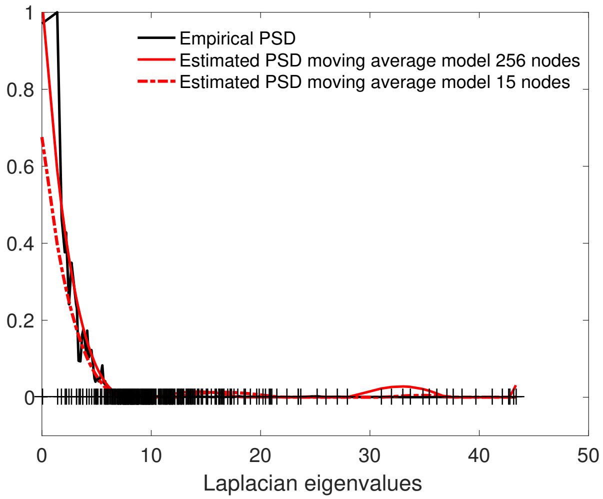


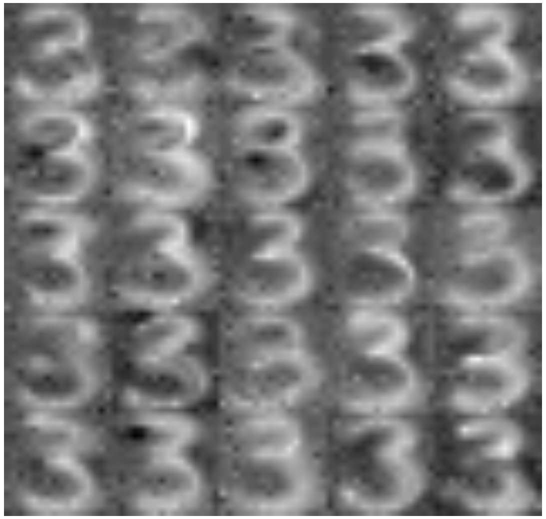
$\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

LSI filter

$\boldsymbol{H}$

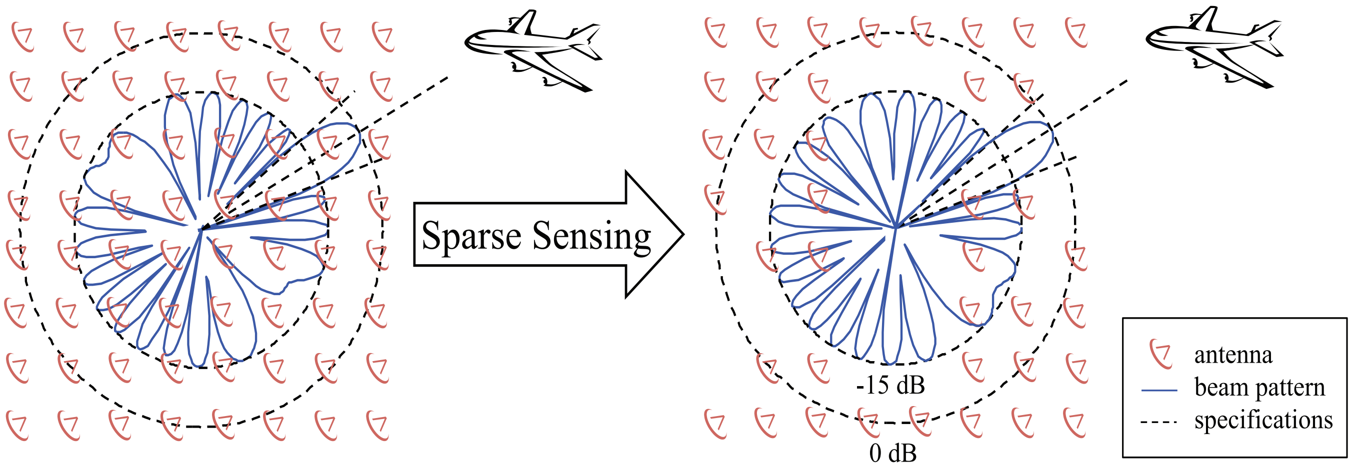
$\boldsymbol{x}$





25 realizations

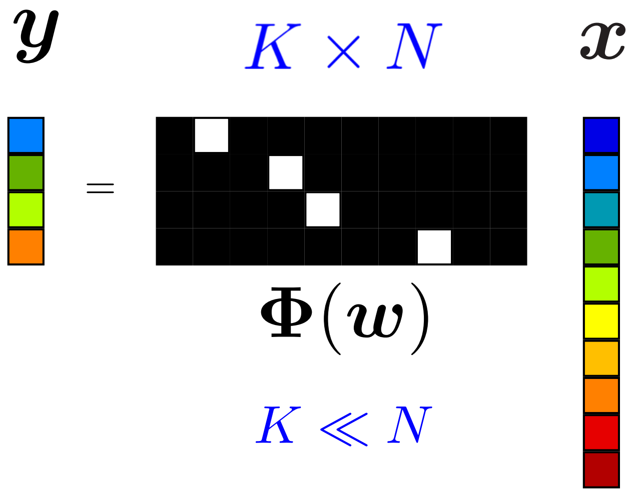
Sparse Sampler Design



S.P. Chepuri and G. Leus. Sparse Sensing for Statistical Inference. *Foundations and Trends in Signal Processing, Vol. 9: No. 3–4, pp 233-368, Dec. 2016.*

Sparse sensing models

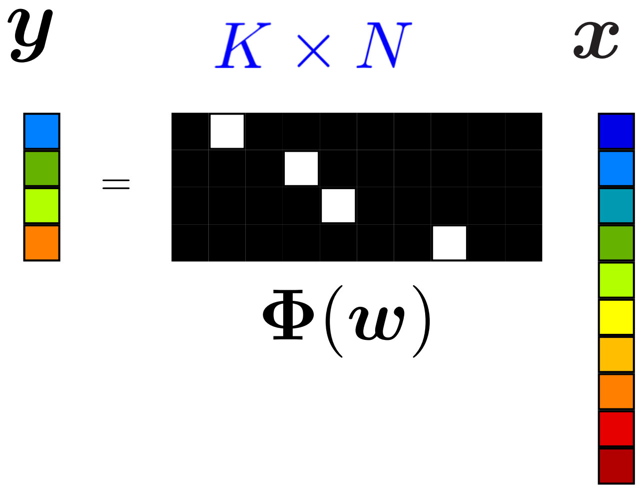
Sparsely sensed signals



Least squares solution: $[\boldsymbol{\Phi} U_{\text{BL}}]^{\dagger}\boldsymbol{y}$

Sparse sensing models

Sparsely sensed statistics



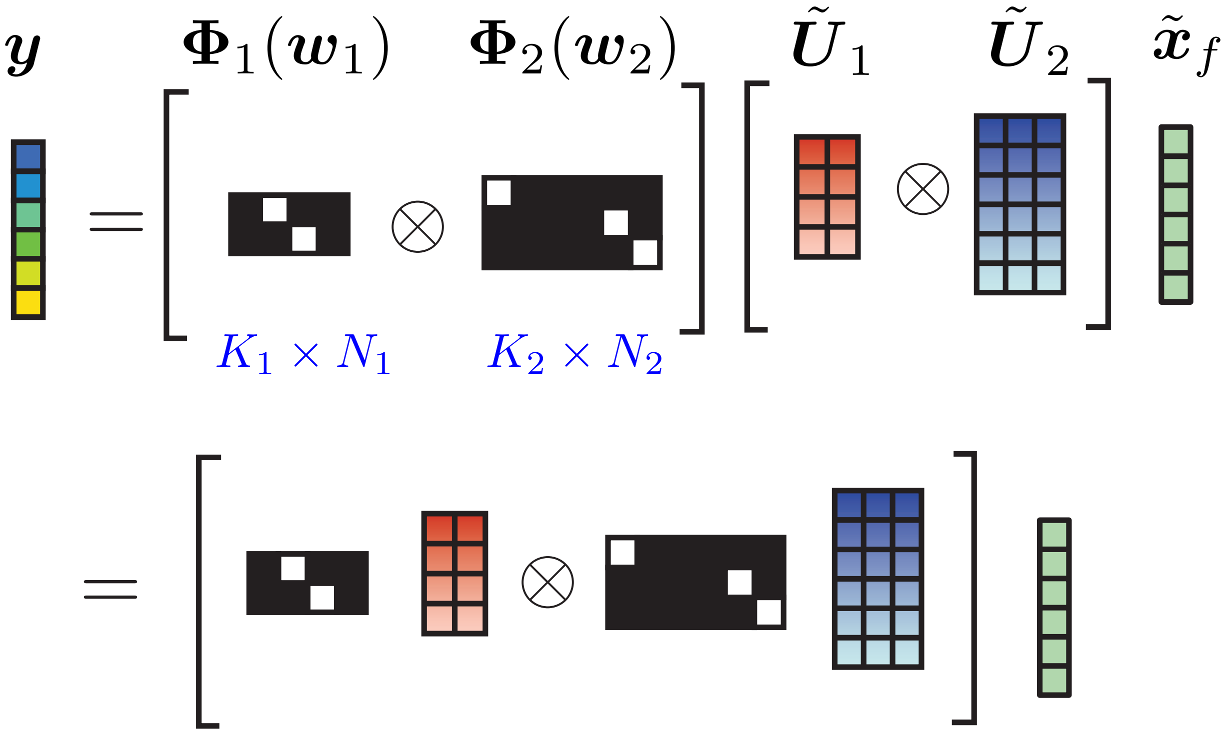
$$\boldsymbol{R_y} = \mathrm{E}\left\{\boldsymbol{y}\boldsymbol{y}^H\right\}$$

$$\boldsymbol{R_x} = \mathrm{E}\left\{\boldsymbol{x}\boldsymbol{x}^H\right\}$$

Least squares solution: $[(\boldsymbol{\Phi} \otimes \boldsymbol{\Phi})\boldsymbol{\Psi}]^{\dagger}\boldsymbol{r_y}$

Sparse sensing models

Sparsely sensed multidomain signals



Least squares solution: $[(\boldsymbol{\Phi}_1 \boldsymbol{U}_1)^\dagger \otimes (\boldsymbol{\Phi}_2 \boldsymbol{U}_2)^\dagger]\boldsymbol{y}$

What is sparse sampling?

$$\boldsymbol{\Phi}(\boldsymbol{w}) \in \{0,1\}^{K\times N}$$

$\boldsymbol{y}$

$$\boldsymbol{R}_{\boldsymbol{y}} = \mathrm{E}\left\{\boldsymbol{y}\boldsymbol{y}^H\right\}$$

$\boldsymbol{x}$

$$\boldsymbol{R}_{\boldsymbol{x}} = \mathrm{E}\left\{\boldsymbol{x}\boldsymbol{x}^H\right\}$$

- Sampling matrix is determined by the sampling vector/set

$$\boldsymbol{w} = [w_1, w_2, \ldots, w_N]^T \in \{0,1\}^N \quad \text{or} \quad \mathcal{S} = \{n | w_n = 1, n = 1, 2, \ldots, N\}$$

$w_m = (0)1$ sample or vertex is (not) selected

- Sparse sampling structure
  - only one nonzero entry per row
  - many zero columns

Design problem

Select the “best” subset of vertices out of the candidate vertices that guarantee a certain desired reconstruction accuracy.

$$\begin{aligned} \underset{\boldsymbol{w}}{\text{optimize}} \quad & f(\boldsymbol{w}) \\ \text{s.to} \quad & \text{card}(\boldsymbol{w}) = K \\ & \boldsymbol{w} \in \{0,1\}^N \end{aligned}$$

or

$$\begin{aligned} \underset{\mathcal{S} \subset \mathcal{N}}{\text{optimize}} \quad & f(\mathcal{S}) \\ \text{s.to} \quad & |\mathcal{S}| = K \end{aligned}$$

$f(\boldsymbol{w})$ reconstruction performance metric

K sample size

$\boldsymbol{w} = [w_1, w_2, \ldots, w_N]^T \in \{0,1\}^N$

$\mathcal{S} = \{n | w_n = 1, n = 1, 2, \ldots, N\}$

$w_m = (0)1$ sample or vertex is (not) selected

Design problem

Select the “best” subset of vertices out of the candidate vertices that guarantee a certain desired reconstruction accuracy.

$$\begin{aligned} \underset{\boldsymbol{w}}{\text{optimize}} \quad & f(\boldsymbol{w}) \\ \text{s.to} \quad & \text{card}(\boldsymbol{w}) = K \\ & \boldsymbol{w} \in \{0,1\}^N \end{aligned}$$

or

$$\begin{aligned} \underset{\mathcal{S} \subset \mathcal{N}}{\text{optimize}} \quad & f(\mathcal{S}) \\ \text{s.to} \quad & |\mathcal{S}| = K \end{aligned}$$

Nonconvex Boolean problem

Solutions to the combinatorial problem

Exact solutions:

- Exhaustive search over
  - ☐ $\binom{N}{K}$ possible candidates
- Branch-and-bound methods

  [Lawler-Wood-1966], [Nguyen-Miller-1992]
  - ☐ long runtimes even for a modest sized problem

- E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Oper. Res.*, vol. 14, pp. 699–719, 1966.
- N. Nguyen and A. Miller, "A review of some exchange algorithms for constructing discrete D-optimal designs," *Comput. Statist. Data Anal.*, vol. 14, pp. 489–498, 1992

Solutions to the combinatorial problem

Suboptimal solutions:

- **Convex** optimization (polynomial time)

  [Joshi-Boyd-2009], [Chepuri-Leus-2015]

  - ☐ convex relaxation for $\{0, 1\}, f(\boldsymbol{w})$
  - ☐ thresholding, randomization to get back a Boolean solution
  - ☐ Semidefinite program (typically)

- S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. Signal Process*., vol. 57, no. 2, pp. 451–462, Feb. 2009
- S.P. Chepuri and G. Leus. "Sparsity-Promoting Sensor Selection for Non-linear Measurement Models," *IEEE Trans. on Signal Processing*, vol. 63, no. 3, pp. 684-698, Feb. 2015.

Solutions to the combinatorial problem

Suboptimal solutions:

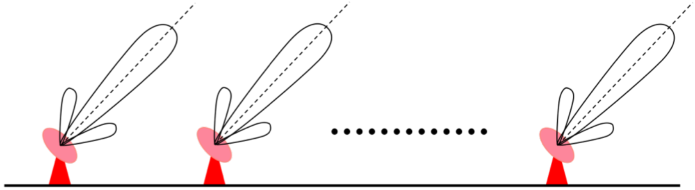
- ➢ **Submodular** optimization (linear search time)

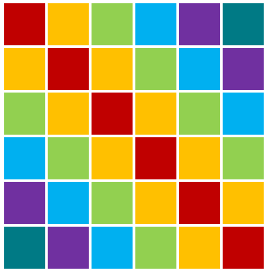
[Krause-Singh-Guestrin-2008], [Ranieri-Chebira-Vetteri-2014]

- ☐ Submodularity of $f(\mathcal{S})$
- ☐ greedy search
- ☐ solution is near optimal

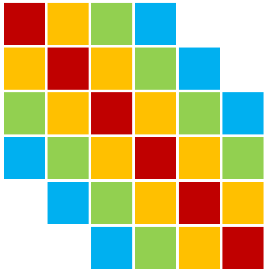
- A. Krause, A. Singh, and C. Guestrin, “Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies,” *J. Machine Learn. Res.*, vol. 9, pp. 235–284, Feb. 2008.
- J. Ranieri, A. Chebira, and M. Vetterli, “Near-optimal sensor placement for linear inverse problems,” *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1135–1146, Mar. 2014

Compressive covariance sensing

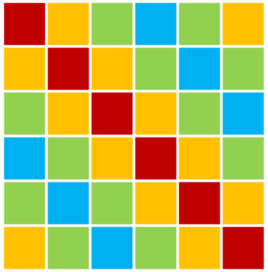




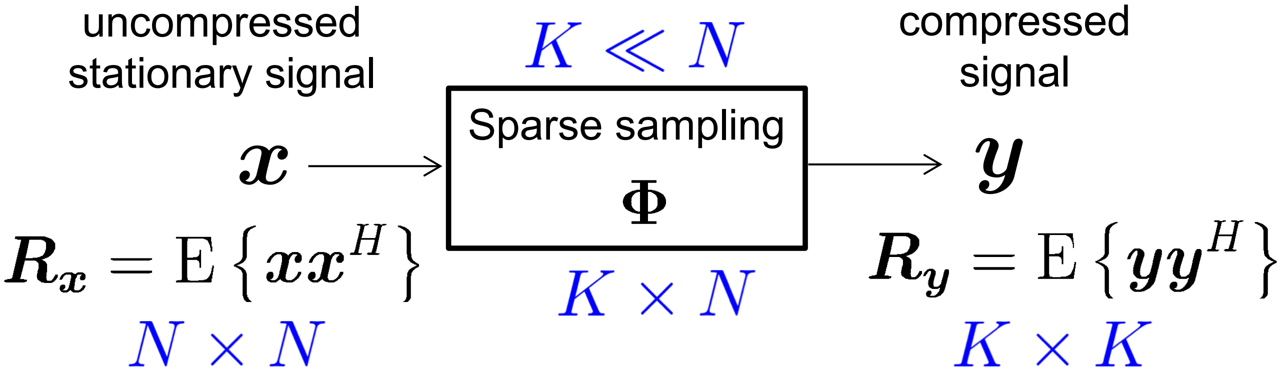
Toeplitz



Banded



Circulant



uncompressed
stationary signal

$$K \ll N$$

compressed
signal

$$\boldsymbol{x} \longrightarrow \boxed{\text{Sparse sampling } \boldsymbol{\Phi}} \longrightarrow \boldsymbol{y}$$

$$\boldsymbol{R_x} = \mathrm{E}\left\{\boldsymbol{x}\boldsymbol{x}^H\right\} \qquad K \times N \qquad \boldsymbol{R_y} = \mathrm{E}\left\{\boldsymbol{y}\boldsymbol{y}^H\right\}$$

$$N \times N \qquad\qquad K \times K$$

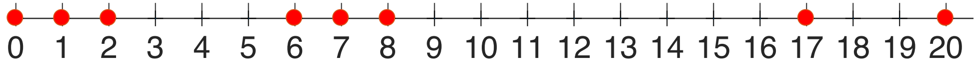
Sparse covariance sensing (Toeplitz structure)

$$\boldsymbol{R}_{\boldsymbol{x}}(\boldsymbol{\theta}) = \sum_{i=1}^{Q} \theta_i \boldsymbol{Q}_i \longrightarrow \boxed{\begin{matrix}\text{compression} \\ \boldsymbol{\Phi}\end{matrix}} \longrightarrow \boldsymbol{R}_{\boldsymbol{y}}(\boldsymbol{\theta}) = \sum_{i=1}^{Q} \theta_i \boldsymbol{\Phi} \boldsymbol{Q}_i \boldsymbol{\Phi}^T$$

- ➢ Minimal sparse rulers ensure identifiability and best compression rate (Toeplitz)

  - ✓ Difference set: $\Delta\mathcal{I} = \{|i_1 - i_2|, \forall i_1, i_2 \in I\}$
  - ✓ Length-$(N-1)$ sparse ruler has $\Delta\mathcal{I} = \{0, 1, \ldots, N-1\}$

$N = 21$:

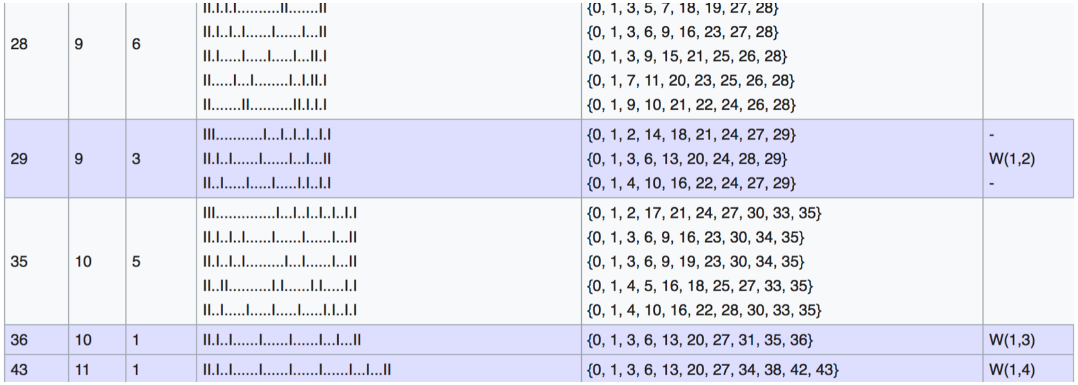


[Redei-Renyi-1949], [Romero-Ariananda-Tian-Leus-2016]

- L. Redei and A. Renyi, “On the representation of the numbers 1,2,. . . ,n by means of differences (Russian),” Matematicheskii sbornik, vol. 66, no. 3, pp. 385–389, 1949.
- D. Romero, D.D. Ariananda, Z. Tian, and G. Leus. “Compressive covariance sensing: Structure-based compressive sensing beyond sparsity,” IEEE Signal Processing Magazine, vol. 33, no. 1, pp.78-93, Jan. 2016.

Sparse covariance sensing (Toeplitz structure)

- Minimal sparse rulers are precomputed



28	9	6	II.I.I.I..........II.......II II.I..I..I......I......I...II II.I.....I.....I.....I...II.I II.....I...I........I..I.II.I II.......II..........II.I.I.I	{0, 1, 3, 5, 7, 18, 19, 27, 28} {0, 1, 3, 6, 9, 16, 23, 27, 28} {0, 1, 3, 9, 15, 21, 25, 26, 28} {0, 1, 7, 11, 20, 23, 25, 26, 28} {0, 1, 9, 10, 21, 22, 24, 26, 28}	
29	9	3	III...........I...I..I..I..I.I II.I..I......I......I...I...II II..I.....I.....I.....I.I..I.I	{0, 1, 2, 14, 18, 21, 24, 27, 29} {0, 1, 3, 6, 13, 20, 24, 28, 29} {0, 1, 4, 10, 16, 22, 24, 27, 29}	- W(1,2) -
35	10	5	III..............I...I..I..I..I..I.I II.I..I..I......I......I......I...II II.I..I..I.........I...I......I...II II..II..........I.I......I.I.....I.I II..I.....I.....I.....I.....I.I..I.I	{0, 1, 2, 17, 21, 24, 27, 30, 33, 35} {0, 1, 3, 6, 9, 16, 23, 30, 34, 35} {0, 1, 3, 6, 9, 19, 23, 30, 34, 35} {0, 1, 4, 5, 16, 18, 25, 27, 33, 35} {0, 1, 4, 10, 16, 22, 28, 30, 33, 35}	
36	10	1	II.I..I......I......I......I...I...II	{0, 1, 3, 6, 13, 20, 27, 31, 35, 36}	W(1,3)
43	11	1	II.I..I......I......I......I......I...I...II	{0, 1, 3, 6, 13, 20, 27, 34, 38, 42, 43}	W(1,4)

https://en.wikipedia.org/wiki/Sparse_ruler

- Suboptimal designs for DOA estimation: co-prime, nested samplers

[Vaidyanathan-Pal-2011]

- P.P. Vaidyanathan and P. Pal. "Sparse sensing with co-prime samplers and arrays." *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 573-586, Feb. 2011.

Submodular optimization

Requires $f(\cdot)$ to be submodular function of its arguments

- Define the sampling set:

$$\mathcal{X} := \mathcal{S} = \{n|w_n = 1, n = 1, 2, \ldots, N\}$$

or

$$\mathcal{X} := \mathcal{N} \setminus \mathcal{S} = \{n|w_n = 0, n = 1, 2, \ldots, N\}$$

- Set function $f(\mathcal{X})$ is submodular, if $\forall \mathcal{X} \subseteq \mathcal{Y} \subset N$, $s \in \mathcal{N} \setminus \mathcal{Y}$

$$f(\mathcal{X} \cup \{s\}) - f(\mathcal{X}) \geq f(\mathcal{Y} \cup \{s\}) - f(\mathcal{Y})$$

- Set function $f(\mathcal{X})$ is monotone non-decreasing, if

$$f(\mathcal{X} \cup \{s\}) \geq f(\mathcal{X})$$

Design problem

Select the “best” subset of vertices out of the candidate vertices that guarantee a certain desired reconstruction accuracy.

$$\underset{\mathcal{X}}{\text{maximize}} \; f(\mathcal{X})$$

$$\text{s.to} \quad |\mathcal{X}| = L$$

$$L = K \text{ or } L = N - K$$

Nonconvex Boolean problem

Submodular optimization

If $f(\cdot)$ is **submodular** and **monotonic**

Linear sweep time

Algorithm 1 Greedy algorithm

```
1. Require $\mathcal{X} = \emptyset, L$.
2. for $k = 1$ to $L$
3.     $s^* = \arg\max_{s \notin \mathcal{X}} f(\mathcal{X} \cup \{s\})$
4.     $\mathcal{X} \leftarrow \mathcal{X} \cup \{s^*\}$
5. end
6. Return $\mathcal{X}$
```

$$L = K \text{ or } L = N - K$$

Then, greedy algorithm is near-optimal

$$f(\mathcal{X}) \geq \underbrace{(1 - 1/e)}_{63\%} \max_{|\mathcal{Y}|=L} f(\mathcal{Y})$$

[Nemhauser-Wolsey-Fisher-1978]

- G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, “An analysis of approximations for maximizing submodular set functions— I,” Mathematical Programming, vol. 14, no. 1, pp. 265–294, 1978.

Design problem

Select the “best” subset of vertices out of the candidate vertices that guarantee a certain desired reconstruction accuracy.

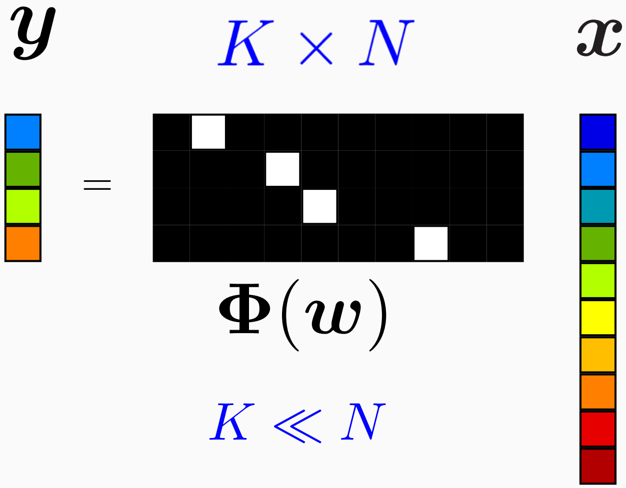
$$\underset{\mathcal{X}}{\text{maximize}} \;\; f(\mathcal{X})$$
$$\text{s.to} \quad |\mathcal{X}| = L$$

$L = K$ or $L = N - K$

What is a suitable submodular function $f(\mathcal{X})$ for sparse sampling?

Sparse sensing models

Sparsely sensed signals



$$\boldsymbol{y} \quad K \times N \quad \boldsymbol{x}$$

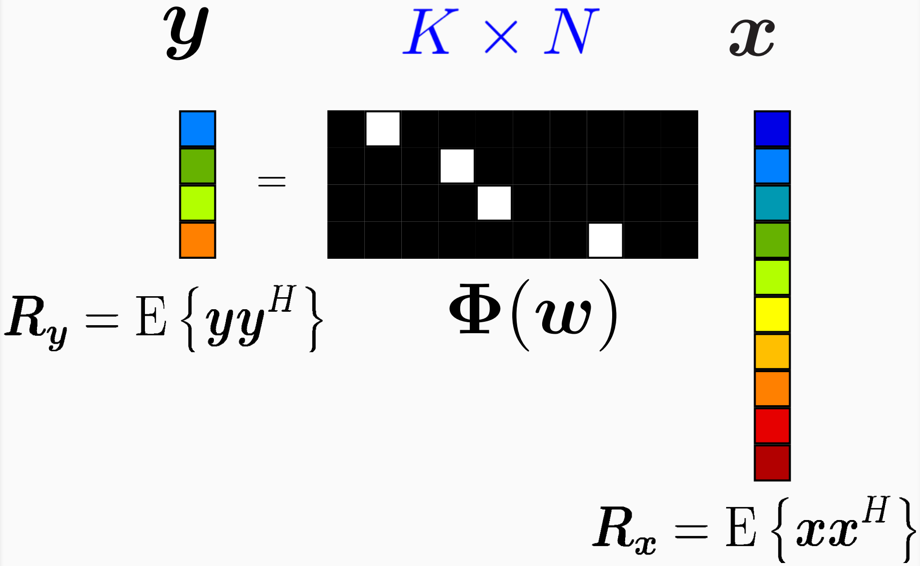
$$=$$

$$\boldsymbol{\Phi}(\boldsymbol{w})$$

$$K \ll N$$

Least squares solution: $[\boldsymbol{\Phi} \boldsymbol{U}_{\text{BL}}]^{\dagger} \boldsymbol{y}$

Sparsely sensed statistics



$$\boldsymbol{y} \quad K \times N \quad \boldsymbol{x}$$

$$=$$

$$\boldsymbol{R}_{\boldsymbol{y}} = \mathrm{E}\left\{\boldsymbol{y}\boldsymbol{y}^{H}\right\} \quad \boldsymbol{\Phi}(\boldsymbol{w})$$

$$\boldsymbol{R}_{\boldsymbol{x}} = \mathrm{E}\left\{\boldsymbol{x}\boldsymbol{x}^{H}\right\}$$

Least squares solution: $[(\boldsymbol{\Phi} \otimes \boldsymbol{\Phi})\boldsymbol{\Psi}]^{\dagger} \boldsymbol{r}_{\boldsymbol{y}}$

How do design the subsampler?

- Quality of the least squares solution

$$[\boldsymbol{\Phi}\boldsymbol{U}_{\text{BL}}]^{\dagger}\boldsymbol{y} \quad \text{or} \quad [(\boldsymbol{\Phi}\otimes\boldsymbol{\Phi})\boldsymbol{\Psi}]^{\dagger}\boldsymbol{r}_b$$

depends on the spectrum (eigenvalues) of

$$\boldsymbol{T}(\boldsymbol{w}) = [\boldsymbol{\Phi}\boldsymbol{U}_{\text{BL}}]^{H}[\boldsymbol{\Phi}\boldsymbol{U}_{\text{BL}}] = \boldsymbol{U}_{\text{BL}}^{H}\text{diag}(\boldsymbol{w})\boldsymbol{U}_{\text{BL}}$$

or

$$\boldsymbol{T}(\boldsymbol{w}) = [(\boldsymbol{\Phi}\otimes\boldsymbol{\Phi})\boldsymbol{\Psi}]^{H}[(\boldsymbol{\Phi}\otimes\boldsymbol{\Phi})\boldsymbol{\Psi}] = \boldsymbol{\Psi}^{H}[\text{diag}(\boldsymbol{w})\otimes\text{diag}(\boldsymbol{w})]\boldsymbol{\Psi}$$

- We try to balance the spectrum:

$$\arg \max_{\boldsymbol{w}\in\{0,1\}^N} \quad \log\det\{\boldsymbol{T}(\boldsymbol{w})\} \quad \text{s.to} \quad \|\boldsymbol{w}\|_0 = K$$

Scalar measure of the error covariance matrix

How to design the subsampler?

$$\arg\max_{\boldsymbol{w}\in\{0,1\}^N} \quad \log\det\{\boldsymbol{T}(\boldsymbol{w})\} \quad \text{s.to} \quad \|\boldsymbol{w}\|_0 = K$$

- Using set notation

$$\mathcal{X} = \{m | w_m = 1, m = 1, 2, \ldots, M\}$$

- Set function

$$f(\mathcal{X}) = \log\det\left\{\sum_{i\in\mathcal{X}} \boldsymbol{u}_{\mathrm{BL},i}\boldsymbol{u}_{\mathrm{BL},i}^H\right\} \quad \text{or} \quad f(\mathcal{X}) = \log\det\left\{\sum_{(i,j)\in\mathcal{X}\times\mathcal{X}} \psi_{i,j}\psi_{i,j}^H\right\}$$

$$\boldsymbol{U}_{\mathsf{BL}} = [\boldsymbol{u}_{\mathsf{BL},1}, \cdots, \boldsymbol{u}_{\mathsf{BL},N}]^T \qquad \boldsymbol{\Psi} = [\psi_{1,1}, \psi_{1,2}, \cdots, \psi_{N,N}]^H$$

Set function is submodular and monotone non-decreasing

How to design the subsampler?

$$\arg\max_{\boldsymbol{w}\in\{0,1\}^N} \log\det\{T(\boldsymbol{w})\} \quad \text{s.to} \quad \|\boldsymbol{w}\|_0 = K$$

- This combinatorial optimization can be near optimally solved using a low-complexity greedy algorithm

$$f(\mathcal{X}) \geq \underbrace{(1 - 1/e)}_{63\%} \max_{|\mathcal{Y}|=K} f(\mathcal{Y})$$

[Nemhauser-Wolsey-Fisher-1978]

```
Require $\mathcal{X} = \emptyset, K$.
for $k = 1$ to $K$
    $s^* = \arg\max_{s \notin \mathcal{X}} f(\mathcal{X} \cup \{s\})$
    $\mathcal{X} \leftarrow \mathcal{X} \cup \{s^*\}$
end
Return $\mathcal{X}$
```

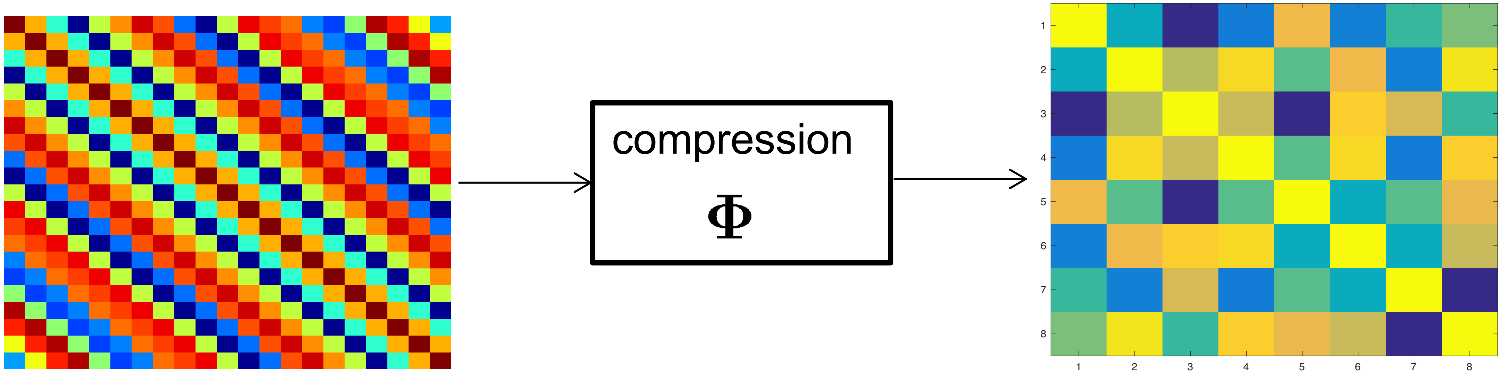
✓ Leverages submodularity

✓ Linear sweep time

- G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, “An analysis of approximations for maximizing submodular set functions— I,” Mathematical Programming, vol. 14, no. 1, pp. 265–294, 1978.

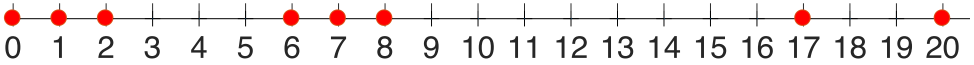
Toeplitz matrix – array processing

$$\boldsymbol{x} = \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{s} + \boldsymbol{n} \Rightarrow \boldsymbol{R_x} = \boldsymbol{A}(\boldsymbol{\theta})\text{diag}(\boldsymbol{\sigma}_s^2)\boldsymbol{A}^H(\boldsymbol{\theta}) + \sigma^2\boldsymbol{I}$$

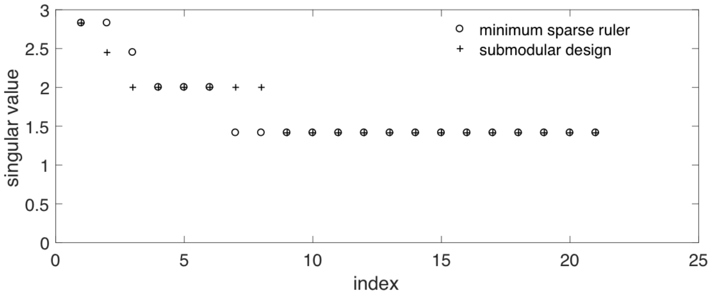


$N = 21$ $K = 8$

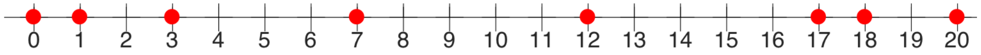
sparse ruler (best compression rate, but not easy to compute)



$$[(\boldsymbol{\Phi} \otimes \boldsymbol{\Phi})\boldsymbol{\Psi}]^H[(\boldsymbol{\Phi} \otimes \boldsymbol{\Phi})\boldsymbol{\Psi}]$$

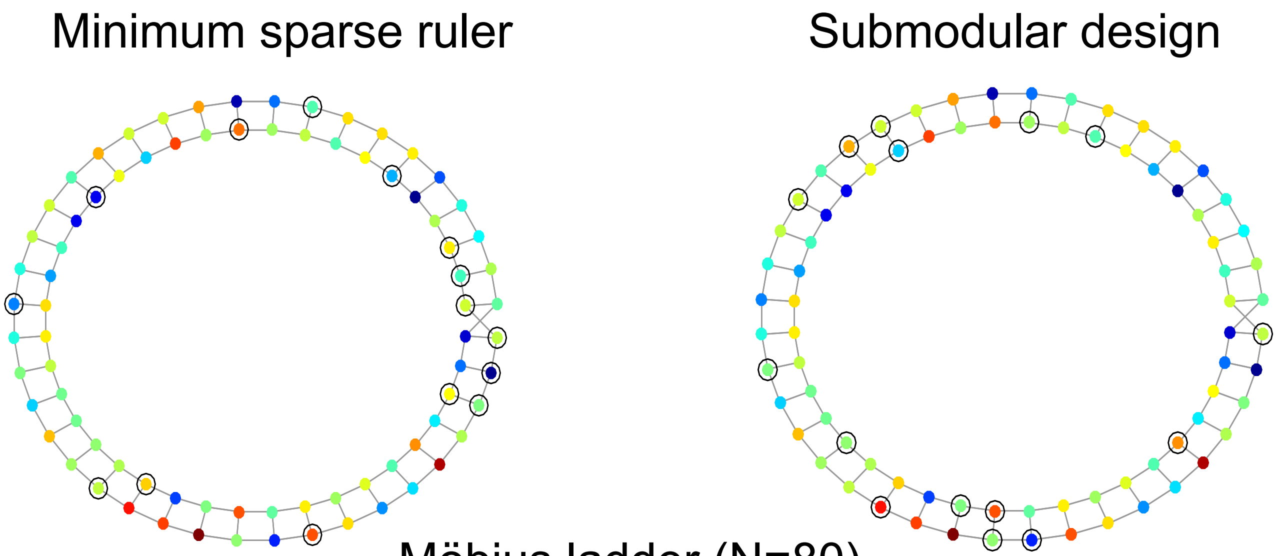


submodular design



Localize more sources than sensors!

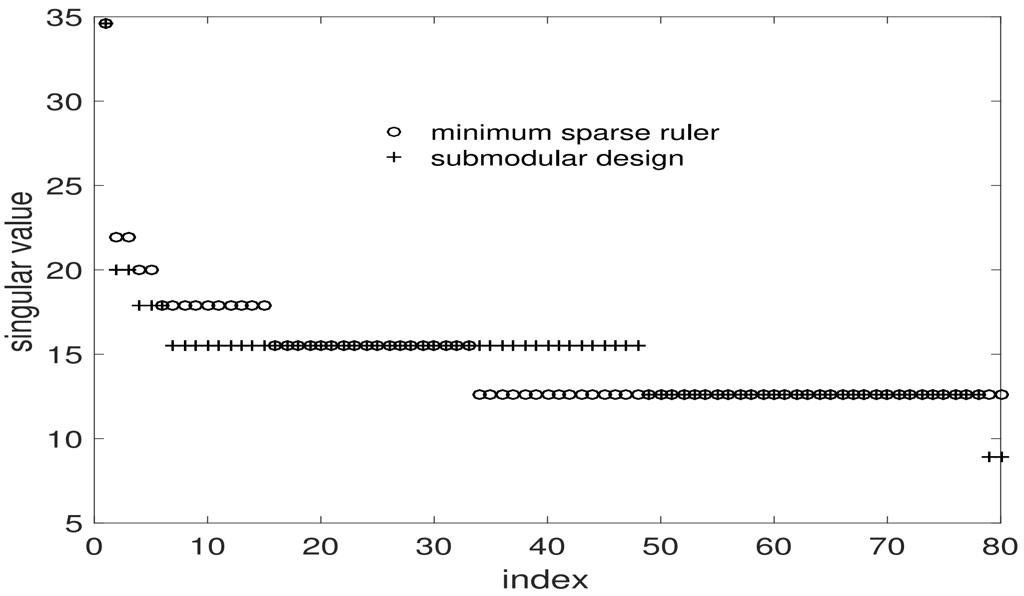
Circulant matrix



Minimum sparse ruler

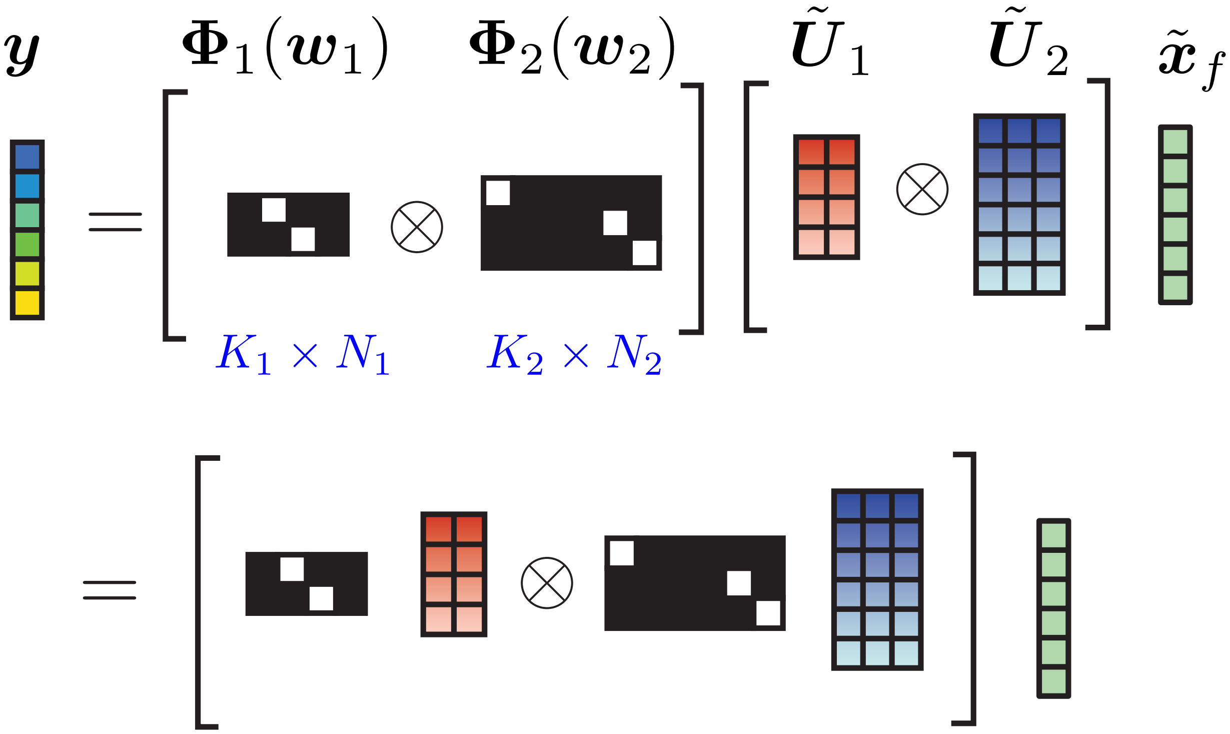
Submodular design

Möbius ladder (N=80)



Sparse sensing models

Sparsely sensed multidomain signals



Least squares solution: $[(\boldsymbol{\Phi}_1\boldsymbol{U}_1)^\dagger \otimes (\boldsymbol{\Phi}_2\boldsymbol{U}_2)^\dagger]\boldsymbol{y}$

Design of $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$ is crucial for the least-squares solution to be unique

How to design the subsampler?

- Quality of the least squares solution

$$[(\mathbf{\Phi}_1 \boldsymbol{U}_1)^\dagger \otimes (\mathbf{\Phi}_2 \boldsymbol{U}_2)^\dagger]\boldsymbol{y}$$

depends on the error covariance matrix

$$\begin{aligned}\boldsymbol{T}(\mathcal{X}) &= \left(\mathbf{\Phi}_1 \tilde{\boldsymbol{U}}_1 \otimes \mathbf{\Phi}_2 \tilde{\boldsymbol{U}}_2\right)^H \left(\mathbf{\Phi}_1 \tilde{\boldsymbol{U}}_1 \otimes \mathbf{\Phi}_2 \tilde{\boldsymbol{U}}_2\right) \\ &= (\mathbf{\Phi}_1 \tilde{\boldsymbol{U}}_1)^H (\mathbf{\Phi}_1 \tilde{\boldsymbol{U}}_1) \otimes (\mathbf{\Phi}_2 \tilde{\boldsymbol{U}}_2)^H (\mathbf{\Phi}_2 \tilde{\boldsymbol{U}}_2) \\ &= \boldsymbol{T}_1(\mathcal{X}_1) \otimes \boldsymbol{T}_2(\mathcal{X}_2)\end{aligned}$$

$\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$

- Since rank$(\boldsymbol{A} \otimes \boldsymbol{B}) =$ rank$(\boldsymbol{A})$rank$(\boldsymbol{B})$, we require (additional constraints)

$$|\mathcal{X}_1| \geq L_1 \text{ and } |\mathcal{X}_2| \geq L_2$$

How to design the subsampler?

- As before, we optimize a scalar function of the error covariance matrix

$$\begin{aligned} &\underset{\mathcal{X}}{\text{maximize}}\ f(\boldsymbol{T}(\mathcal{X})) \\ &\text{s.to}\quad |\mathcal{X}| = K,\ \mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \\ &\qquad\quad |\mathcal{X}| \geq L_1 \quad |\mathcal{X}_2| \geq L_2 \end{aligned}$$

- In particular, we minimize the so-called *frame potential* (related to the mean squared error)

$$F(\mathcal{X}) := \text{trace}\{\boldsymbol{T}^H\boldsymbol{T}\} = \text{trace}\{\boldsymbol{T}_1^H\boldsymbol{T}_1 \otimes \boldsymbol{T}_2^H\boldsymbol{T}_2\} := F_1(\mathcal{X}_1)F_2(\mathcal{X}_2)$$

- Or, maximize the set function with change of variable $\mathcal{S} = \mathcal{N} \setminus \mathcal{X}$

$$G(\mathcal{S}) = F(\mathcal{N}) - F(\mathcal{N} \setminus \mathcal{S}) \qquad\qquad \mathcal{N} = \mathcal{N}_1 \cup \mathcal{N}_2$$

Set function is submodular and monotone non-decreasing

How to design the subsampler?

- Therefore, we have to solve

Truncated partition matroid

$$\underset{\mathcal{S}\subseteq\mathcal{N}}{\text{maximize}}\ G(\mathcal{S})$$

$$\text{s.to}\quad \mathcal{S} \in \mathcal{I}_u \cap \mathcal{I}_p,$$
$$\mathcal{I}_u = \{\mathcal{S} \subseteq \mathcal{N} : \mathcal{S} \leq N - K\}$$
$$\mathcal{I}_p = \{\mathcal{S} \subseteq \mathcal{N} : |\mathcal{S} \cap \mathcal{N}_i| \leq N_i - L_i, i = 1, 2\}$$

[Ortiz-Jiménez et al.-2018]

```
1. Require $\mathcal{X} = \emptyset, K, \mathcal{I}_u, \mathcal{I}_p$.
2. for $k = 1$ to $N - K$
3.     $s^* = \underset{s \notin \mathcal{X}}{\arg\max}\, \{f(\mathcal{X} \cup \{s\}) : \mathcal{X} \in \mathcal{I}_u \cap \mathcal{I}_p\}$
4.     $\mathcal{X} \leftarrow \mathcal{X} \cup \{s^*\}$
5. end
6. Return $\mathcal{X}$
```

- Near optimality guarantees

$$G(\mathcal{S}_{\text{greedy}}) \geq \tfrac{1}{2} G(\mathcal{S}^\star)$$

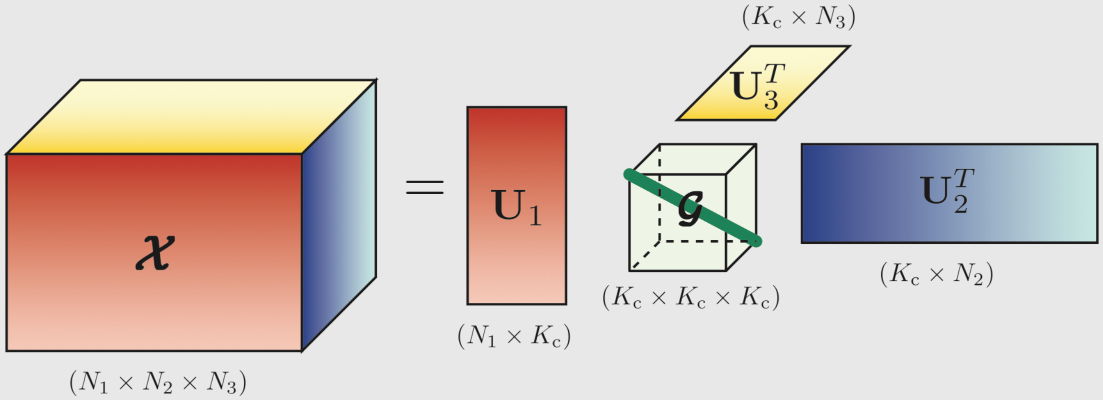
[Nemhauser-Wolsey-Fisher-1978]

- Linear sweep time

- G. Ortiz-Jiménez, M. Coutino, S.P. Chepuri, and G. Leus. Sparse Sampling for Inverse Problems with Tensors. *IEEE TSP (under review)*, June 2018. (available as arXiv:1806.10976).
- G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, “An analysis of approximations for maximizing submodular set functions— I,” Mathematical Programming, vol. 14, no. 1, pp. 265–294, 1978.

Sparse tensor sampling – diagonal core

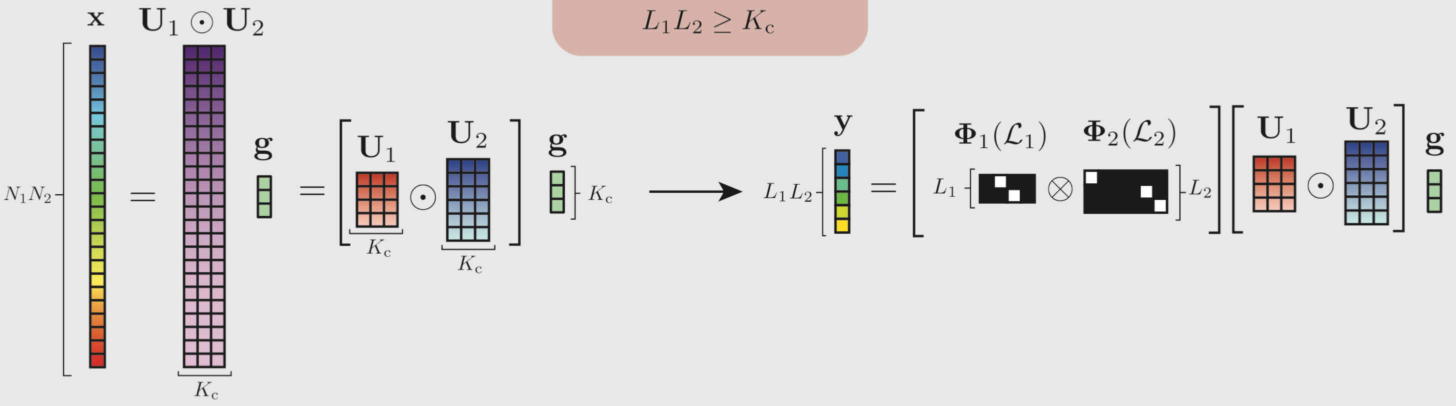
Diagonal core



Sparse sampling

Identifiability

$L_1L_2 \geq K_c$



Sampler design

No all-0 column

Khatri-Rao product always improves rank: $\mathrm{rank}(\mathbf{A} \odot \mathbf{B}) \geq \max\{\mathrm{rank}(\mathbf{A}), \mathrm{rank}(\mathbf{B})\}$

If $L_i > z_i$ and there is one $\mathrm{rank}(\mathbf{\Psi}_j) = K_c$ then $\mathrm{rank}(\mathbf{\Psi}_1 \odot \cdots \odot \mathbf{\Psi}_R) = K_c$

max #zeros any col. $\mathbf{U}_i$

Frame potential: $F(\mathcal{L}) := ||\mathbf{T}_1 \circ \cdots \circ \mathbf{T}_R||_F^2$ ⟶ Submodular surrogate

Sparse tensor sampling

$G(\mathcal{S}_{\mathrm{greedy}}) \geq 0.5G(\mathcal{S}^\star)$

$$\underset{\mathcal{L}_1,\ldots,\mathcal{L}_R}{\text{maximize}} \; f\left\{\mathbf{T}(\mathcal{L})\right\} \;\text{ s.t. }\; \sum_{i=1}^{R} |\mathcal{L}_i| = L, \; \mathcal{L} = \bigcup_{i=1}^{R} \mathcal{L}_i,$$

$$L_j \geq \max\{K_{\mathrm{c}}, z_j + 1\}$$

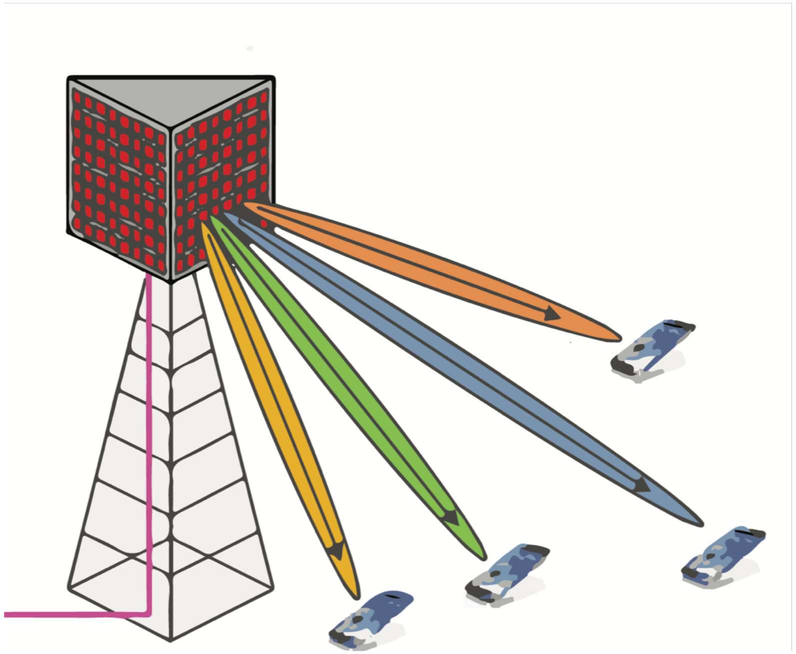
$$L_i \geq \max\{1, z_i + 1\} \quad i \neq j$$

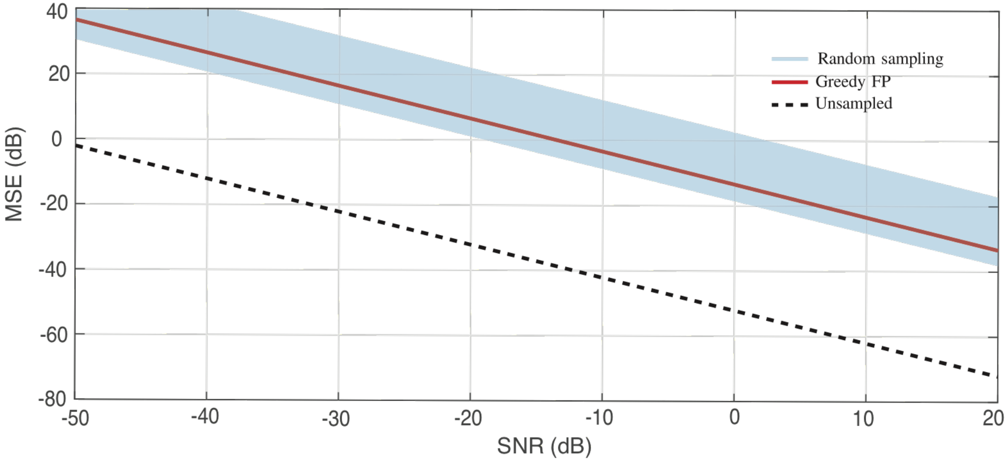
Illustration – multiuser source separation

- Resource allocation for source separation $x(r,l,m,n) = \sum_{k=0}^{K_c} s_k(r)c_k(l)e^{j2\pi n\Delta_x \sin\theta_k} e^{j2\pi n\Delta_y \sin\phi_k}$
- 50 x 60 uniform rectangular array
- 10 users with 100 samples spreading sequence
- Select L=15 ⟶ 0.048% samples out of 300,000

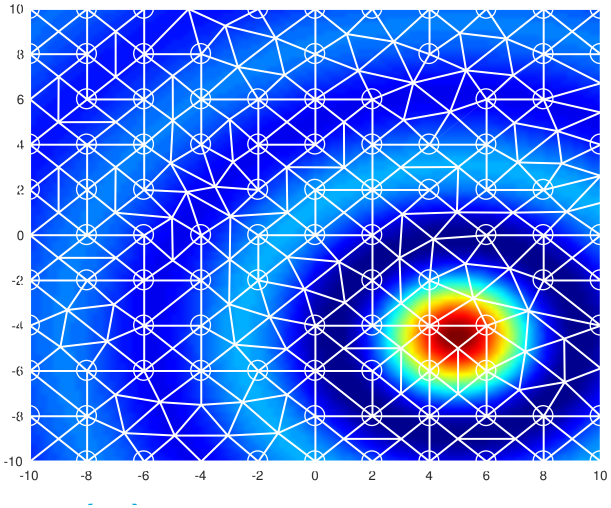
$$\downarrow$$

$$\boldsymbol{\mathcal{X}}(r) = \boldsymbol{\mathcal{S}}(r) \bullet_1 \mathbf{U}_1 \bullet_2 \mathbf{U}_2 \bullet_3 \mathbf{U}_3 + \boldsymbol{\mathcal{W}}(r)$$

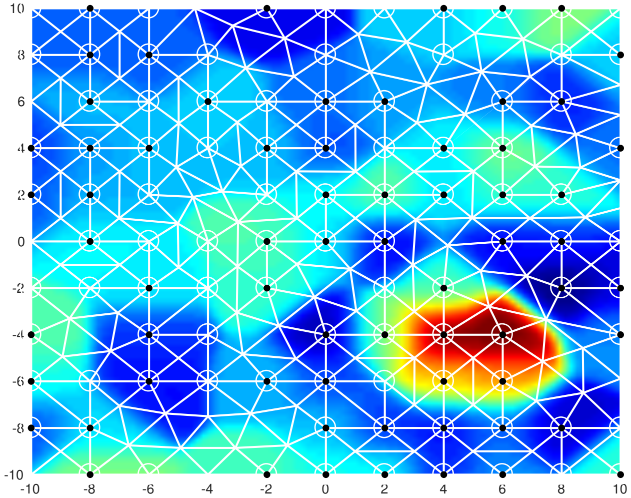




Sampler design for kernel-based method



Ground truth



Measured 67 out of 97 mesh points

Design of sampling sets for kernel methods

- Submodular optimization
- Convex optimization

[Coutino-Chepuri-Leus-2018]

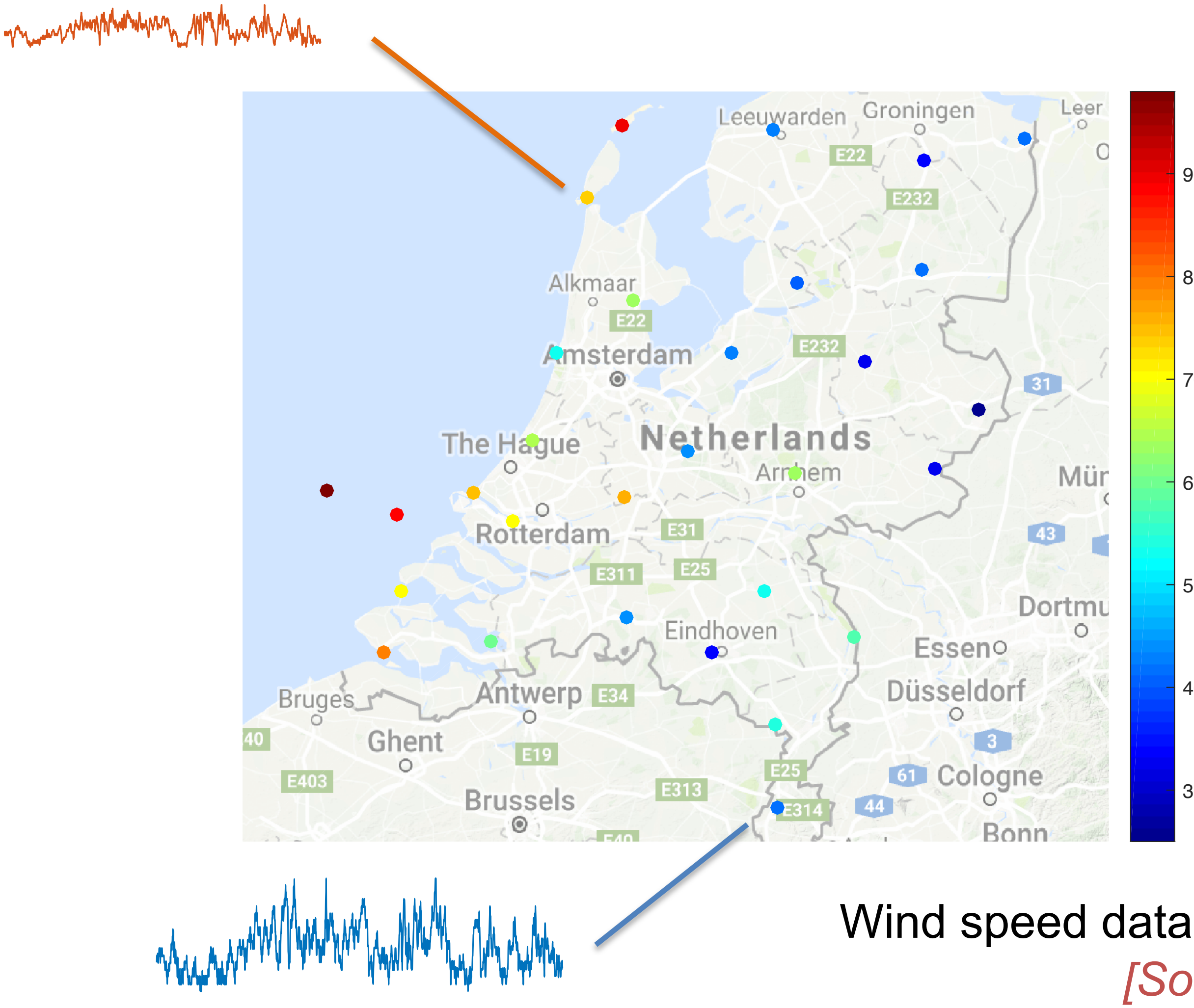
- M. Coutino, S.P. Chepuri and G. Leus. Subset Selection for Kernel-based Reconstruction. In Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2018), Calgary, Canada, April 2018.

Sparse Graph Learning

- S.P. Chepuri, S. Liu, G. Leus, and A. Hero. Learning Sparse Graphs Under Smoothness Prior. *ICASSP 2017*, New Orleans, USA.

- V. Kalofolias, "How to learn a graph from smooth signals," in Proc. of the 19th International Conference on Artificial Intelligence and Statistics, 2016.

- X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning laplacian matrix in smooth graph signal representations," *IEEE TSP, vol. 64, no. 23, Dec. 2016*.



Wind speed data from 30 stations

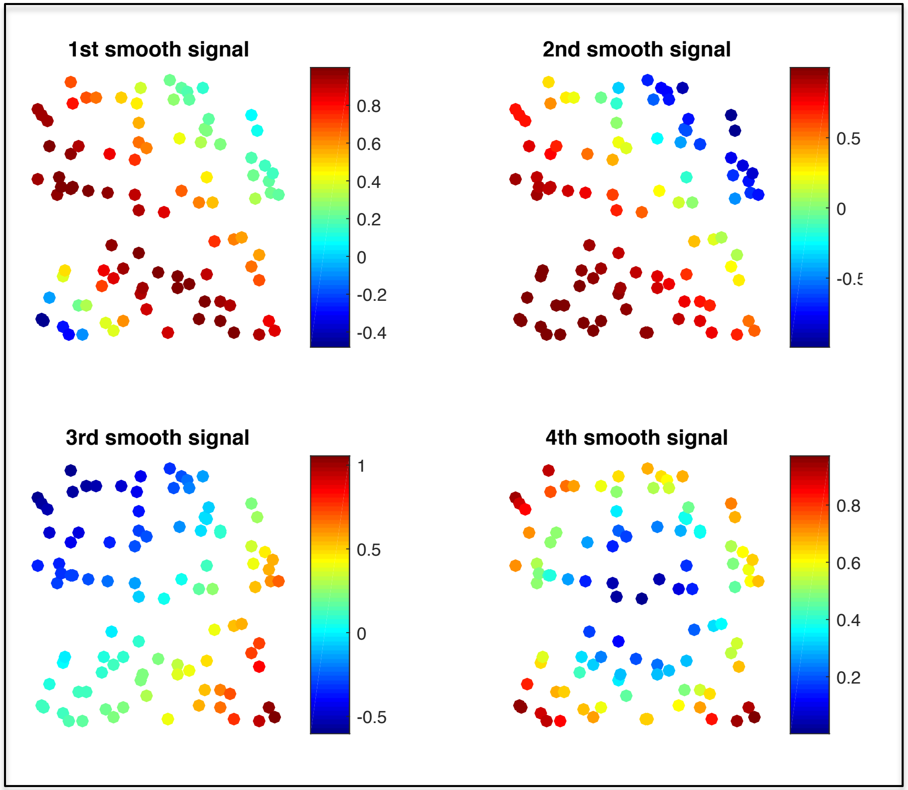
[Source: KNMI, Netherlands]

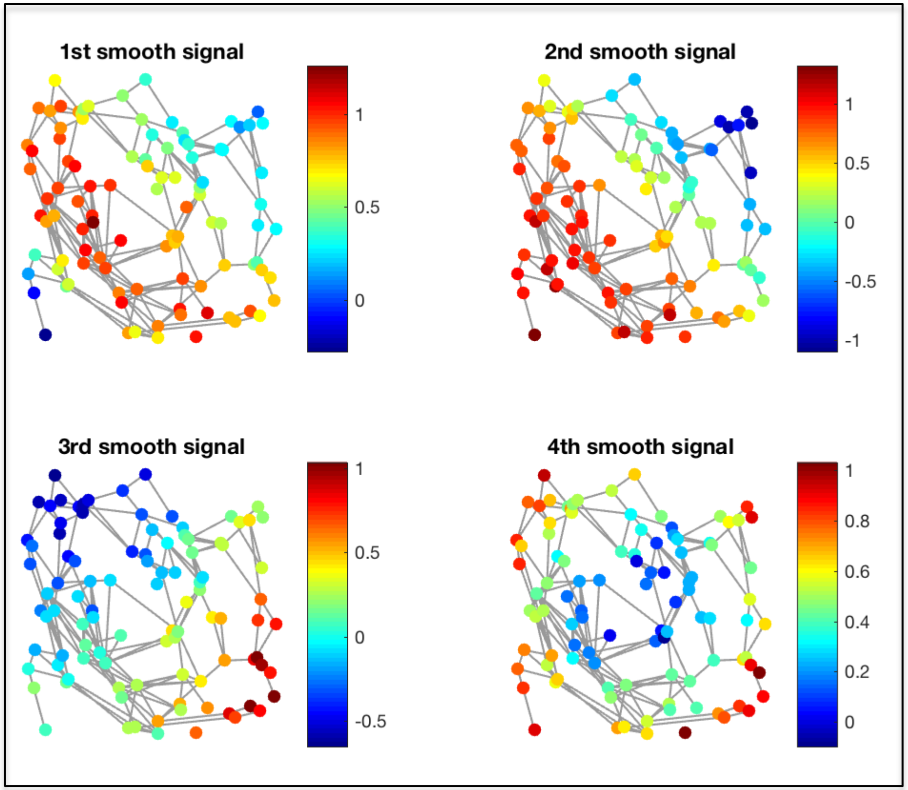
“Learn a sparse graph that sufficiently explains the data”

Sparse graph learning problem

Learn a “**sparse graph**” (or the graph Laplacian) from data:

- ✓ with “ K ” edges
- ✓ data varies “smoothly” on the resulting graph

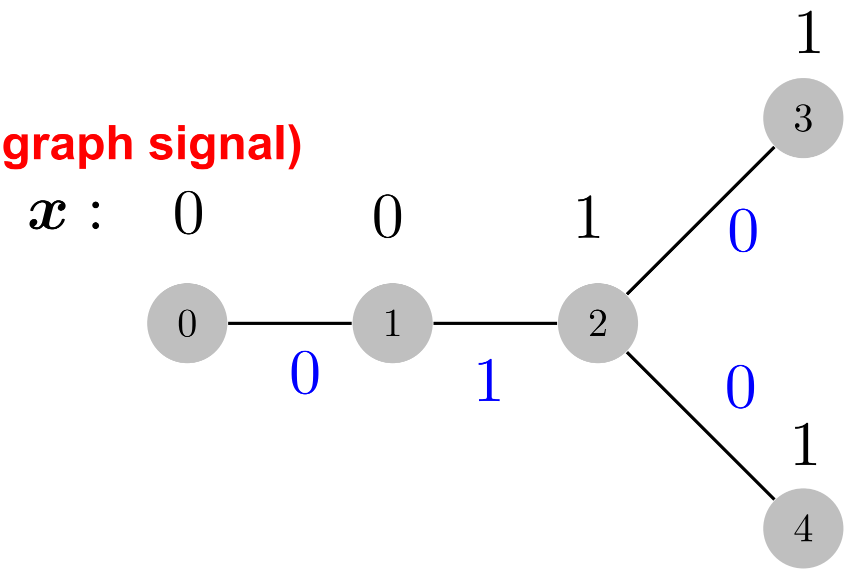




Learnt graph with K = 175 edges using 4 snapshots

Graph Laplacian – quadratic form

(graph signal)



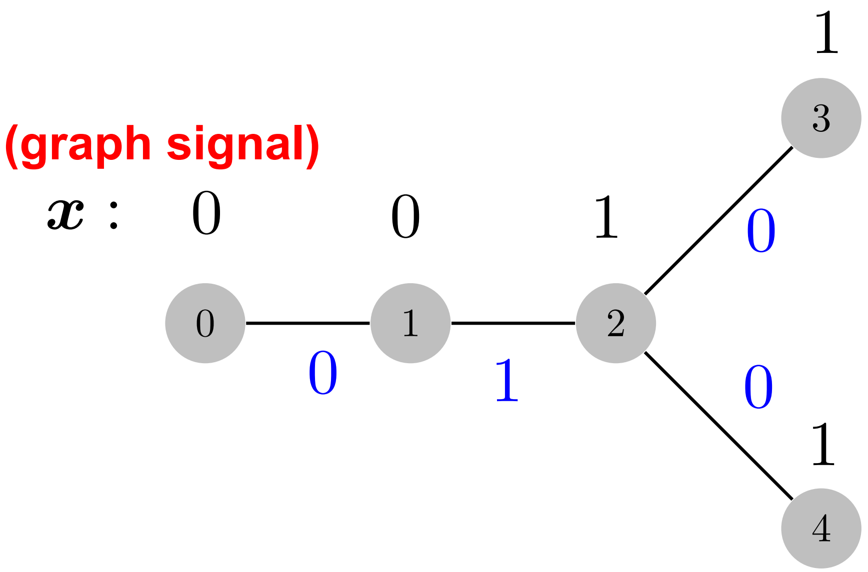
$$\boldsymbol{x}^T \boldsymbol{L} \boldsymbol{x} = \sum_{(i,j) \in \mathcal{E}} (x_i - x_j)^2$$

$$= 1$$

Sum of squares of differences across edges

- Quantifies **smoothness** of $\boldsymbol{x}$ with respect to the underlying graph

Graph Laplacian – quadratic form



$$\boldsymbol{x}^T \boldsymbol{L} \boldsymbol{x} = \sum_{(i,j)\in\mathcal{E}} (x_i - x_j)^2$$

$$= 1$$

Sum of squares of differences across edges

- Laplacian matrix can be written as a outer product of “incidence” vectors

$$\boldsymbol{L} = \boldsymbol{A}\boldsymbol{A}^T = \sum_{m=1}^{M} \boldsymbol{a}_m \boldsymbol{a}_m^T \quad \text{(quadratic form)}$$

$$\left.\begin{array}{r} [\boldsymbol{a}_m]_i = 1 \\ [\boldsymbol{a}_m]_j = -1 \\ \text{zeros elsewhere} \end{array}\right\} \text{For an edge “m” connecting node “i ” and “j ”}$$

Graph learning as a sampling problem

- Denote the subgraph of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ or K-sparse graph

$$\mathcal{G}_s(\mathcal{V}, \mathcal{E}_s) \text{ with the edge set } \mathcal{E}_s \subset \mathcal{E} \text{ such that } |\mathcal{E}_s| = K \ll M$$

- Introduce an "edge sampling" vector

$$\boldsymbol{w} = [w_1, w_2, \cdots, w_M]^T \in \{0, 1\}^M$$

$w_m = 1$ if an edge belongs to the edge subset $\mathcal{E}_s$

No. of edges of:
- Complete graph
- Given graph

- Graph Laplacian of the K-sparse graph

$$\boldsymbol{L}_s(\boldsymbol{w}) = \sum_{m=1}^{M} w_m \boldsymbol{a}_m \boldsymbol{a}_m^T$$

(Recall the outer product decomposition of the Laplacian)

Sparse edge selection

- Given L “noiseless” graph signals $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_L]$

- *K-sparse* graph learning will be

$$\arg\min_{\boldsymbol{w}\in\mathcal{W}} \quad \frac{1}{L}\sum_{k=1}^{L} \boldsymbol{x}_k^T \boldsymbol{L}_s(\boldsymbol{w})\boldsymbol{x}_k = \frac{1}{L}\mathrm{tr}\{\boldsymbol{X}^T \boldsymbol{L}_s(\boldsymbol{w})\boldsymbol{X}\}$$

$$\mathcal{W} = \{\boldsymbol{w} \in \{0,1\}^M \mid \|\boldsymbol{w}\|_0 = K\}$$

Non-convex (Boolean optimization problem)

Sparse edge selection

- Given L “noiseless” graph signals $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_L]$
- *K-sparse* graph learning will be

$$\arg\min_{\boldsymbol{w}\in\mathcal{W}} \quad \frac{1}{L}\sum_{k=1}^{L}\boldsymbol{x}_k^T\boldsymbol{L}_s(\boldsymbol{w})\boldsymbol{x}_k = \frac{1}{L}\mathrm{tr}\{\boldsymbol{X}^T\boldsymbol{L}_s(\boldsymbol{w})\boldsymbol{X}\}$$

$$\mathcal{W} = \{\boldsymbol{w} \in \{0,1\}^M \mid \|\boldsymbol{w}\|_0 = K\}$$

- Cost function (modular):

$$\frac{1}{L}\mathrm{tr}\left\{\boldsymbol{X}^T\boldsymbol{L}_s(\boldsymbol{w})\boldsymbol{X}\right\} = \sum_{m=1}^{M} w_m \mathrm{tr}\left\{\boldsymbol{X}^T(\boldsymbol{a}_m\boldsymbol{a}_m{}^T)\boldsymbol{X}\right\}$$

- **Solution: rank ordering!**

  - ✓ Computational complexity O(K log K), or O(K) with parallel implementation

Sparse edge selection

- Given L “noiseless” graph signals, K-sparse graph learning

$$\arg\min_{\boldsymbol{w}\in\mathcal{W}} \quad \frac{1}{L}\sum_{k=1}^{L}\boldsymbol{x}_k^T\boldsymbol{L}_s(\boldsymbol{w})\boldsymbol{x}_k = \frac{1}{L}\mathrm{tr}\{\boldsymbol{X}^T\boldsymbol{L}_s(\boldsymbol{w})\boldsymbol{X}\}$$

$$\mathcal{W} = \{\boldsymbol{w} \in \{0,1\}^M \mid \|\boldsymbol{w}\|_0 = K\}$$

Example: Suppose covariance matrix of $\boldsymbol{x}$ is $\boldsymbol{R}_{\boldsymbol{x}}$, then

$$L^{-1}\mathrm{tr}\{\boldsymbol{X}^T\boldsymbol{L}_s(\boldsymbol{w})\boldsymbol{X}\} = \sum_{m=1}^{M} w_m(\boldsymbol{a}_m{}^T\widehat{\boldsymbol{R}}_{\boldsymbol{x}}\boldsymbol{a}_m)$$

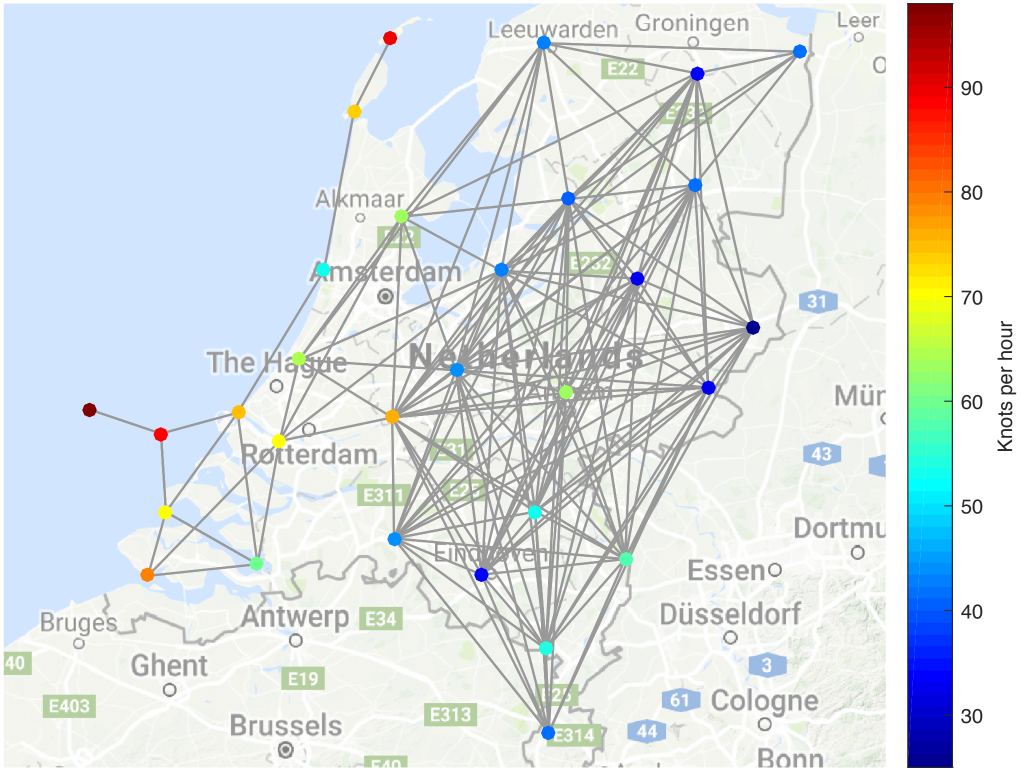
Solution: select K edges between those nodes having highest cross-correlation as

$$\boldsymbol{a}_m{}^T\widehat{\boldsymbol{R}}_{\boldsymbol{x}}\boldsymbol{a}_m = [\widehat{\boldsymbol{R}}_{\boldsymbol{x}}]_{i,i} + [\widehat{\boldsymbol{R}}_{\boldsymbol{x}}]_{j,j} - 2[\widehat{\boldsymbol{R}}_{\boldsymbol{x}}]_{i,j}$$

(Special case: GMRF model with $\boldsymbol{R}_{\boldsymbol{x}} := \boldsymbol{L}^\dagger + \sigma^2\mathbf{I}$)

Numerical experiments – windspeed data

K=125

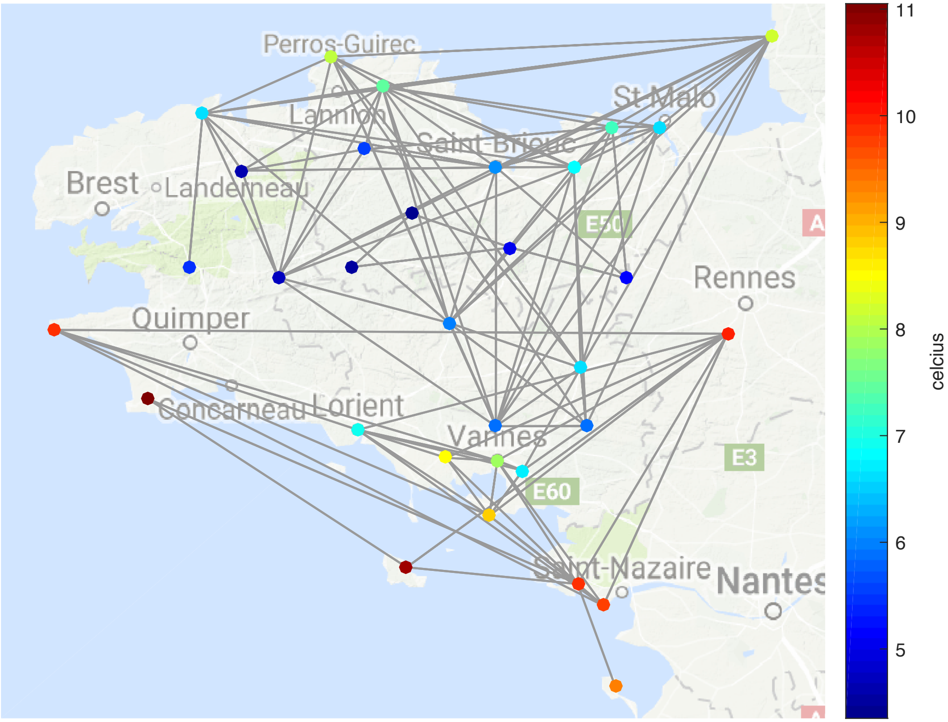


Wind speed data of year 2002 from 30 stations

[Source: KNMI, Netherlands]

Numerical experiments – French temp. data

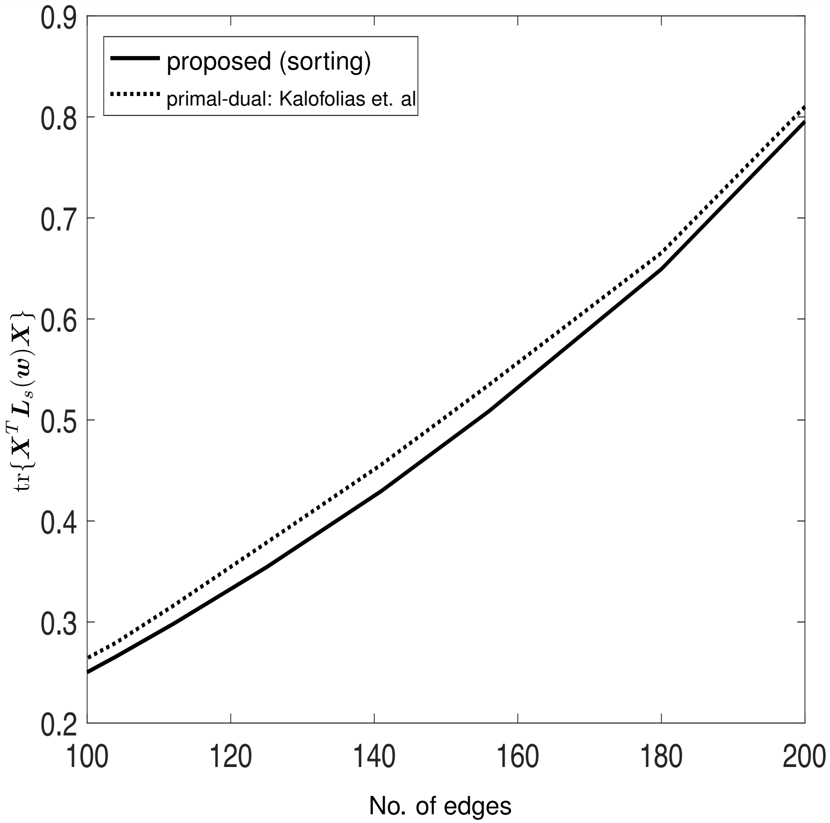
K=110



Temperature data of Brittany, France from 32 stations

Thanks to N. Perraudin and P. Vandergheynst for the dataset.

Numerical experiments - performance



Kalofolias: $\text{minimize}_{\boldsymbol{L}\in\mathcal{L}} \sum_{k=1}^{L} \boldsymbol{x}_k^T \boldsymbol{L} \boldsymbol{x}_k + \lambda \text{card}(\mathbf{L})$

$$\mathcal{L} = \{\boldsymbol{L} \succeq 0, L_{i,j} = L_{j,i} \leq 0, \boldsymbol{L1} = \mathbf{0}\}$$

- V. Kalofolias, “How to learn a graph from smooth signals,” in Proc. of the 19th International Conference on Artificial Intelligence and Statistics, 2016, pp. 920–929.

Sparse edge selection with “denoising”

- Given “L” noisy signals: $\boldsymbol{y}_k = \boldsymbol{x}_k + \boldsymbol{n}_k$,

$$\arg\min_{\{\boldsymbol{x}_k\}_{k=1}^{L}, \boldsymbol{w}\in\mathcal{W}} \frac{1}{L}\sum_{k=1}^{L}(\|\boldsymbol{y}_k - \boldsymbol{x}_k\|_2^2 + \gamma\, \boldsymbol{x}_k^T \boldsymbol{L}_s(\boldsymbol{w})\boldsymbol{x}_k)$$

- Solution 1: (alternating minimization)

Fixed $\boldsymbol{w}$: $\boldsymbol{X}_{\min}(\boldsymbol{w}) = [\mathbf{I} + \gamma \boldsymbol{L}_s(\boldsymbol{w})]^{-1}\boldsymbol{Y}$ (denoising)

Fixed $\boldsymbol{X}$: $\boldsymbol{w}_{\min}(\boldsymbol{X})$ sorting, as before (edge selection)

- ✓ Converges to a stationary point
- ✓ Suffers from the choice of the initial estimate

Sparse edge selection and “denoising”

- Given “*L*” noisy signals: $\boldsymbol{y}_k = \boldsymbol{x}_k + \boldsymbol{n}_k$,

$$\arg\min_{\{\boldsymbol{x}_k\}_{k=1}^{L}, \boldsymbol{w}\in\mathcal{W}} \frac{1}{L}\sum_{k=1}^{L}(\|\boldsymbol{y}_k - \boldsymbol{x}_k\|_2^2 + \gamma\, \boldsymbol{x}_k^T \boldsymbol{L}_s(\boldsymbol{w})\boldsymbol{x}_k)$$

- ***Solution 2: (convex optimization – one step)***

$$\widehat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}\in\mathcal{W}} \quad r(\boldsymbol{w}); \qquad \widehat{\boldsymbol{X}} = \boldsymbol{X}_{\min}(\widehat{\boldsymbol{w}})$$

with $r(\boldsymbol{w}) = \|\boldsymbol{Y} - \boldsymbol{X}_{\min}(\boldsymbol{w})\|_F^2 + \gamma\,\mathrm{tr}\{\boldsymbol{X}_{\min}^T(\boldsymbol{w})\boldsymbol{L}_s(\boldsymbol{w})\boldsymbol{X}_{\min}(\boldsymbol{w})\}$

Hint: Solution to optimal “X” as a function of “w” can be computed in closed form

- Convex program:

$$\begin{aligned} &\arg\min_{\boldsymbol{Z},\boldsymbol{w}} \quad \mathrm{tr}\{\boldsymbol{Z}\} \\ &\text{s.to} \quad \begin{bmatrix} \boldsymbol{Z} - \gamma\boldsymbol{Y}^T\boldsymbol{L}_s(\boldsymbol{w})\boldsymbol{Y} & \boldsymbol{Y}^T \\ \boldsymbol{Y} & \boldsymbol{I} + \gamma\boldsymbol{L}_s(\boldsymbol{w}) \end{bmatrix} \succeq \boldsymbol{0}_{L+N}, \\ &\qquad \boldsymbol{1}^T\boldsymbol{w} = K,\ 0 \le w_m \le 1, m = 1, 2, \ldots, M, \end{aligned}$$

Summary

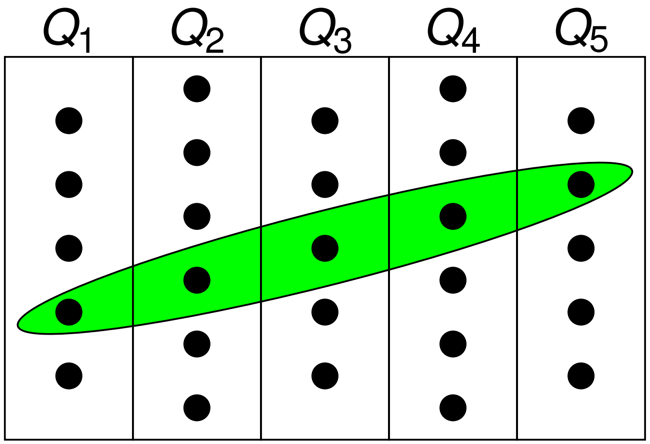
- Reconstructing bandlimited/smooth graph signals via sparse sampling
- Relation to kernel-based signal reconstruction
- Reconstructing product graph signals via sparse tensor sampling
- Reconstructing second-order statistics by subsampling without priors
- Sparse graph learning as a sampling problem

Thank You!
Questions?

Matroids

A finite matroid $\mathcal{M}$ is a pair $(\mathcal{N}, \mathcal{I})$, where $\mathcal{N}$ is a finite set (also called the ground set) and $\mathcal{I}$ is a family of subsets of $\mathcal{N}$ (called the independent sets) that satisfies the following properties:

1. The empty set is independent, i.e., $\varnothing \in \mathcal{I}$.
2. For every $\mathcal{X} \subseteq \mathcal{Y} \subseteq \mathcal{N}$, if $\mathcal{Y} \in \mathcal{I}$, then $\mathcal{X} \in \mathcal{I}$.
3. For every $\mathcal{X}, \mathcal{Y} \subseteq \mathcal{N}$ such that $|\mathcal{Y}| > |\mathcal{X}|$ and $\mathcal{X}, \mathcal{Y} \in \mathcal{I}$ there exists one $x \in \mathcal{Y} \setminus \mathcal{X}$ such that $\mathcal{X} \cup \{x\} \in \mathcal{I}$.



Example: *partition matroid*

S is independent, if
$|S \cap Q_i| \leq 1$ for each Q_i.