Polynomial filters of multiple commutative shifts and their distributed implementation

Qiyu Sun

Department of Mathematics
University of Central Florida

Inverse Problems and Analysis Seminar, University of Delaware, October 5, 2021

Thank Mahya Ghandehari for the invitation.

1. Graph signal processing and data science on networks
2. Graph Laplacian and graph Fourier transform
3. Commutative graph shifts
4. Polynomial filters and distributed implementation
5. Inverse of polynomial filters and distributed implementation
6. Distance between non-polynomial filters and polynomial filters
7. Numerical demonstrations

Abstract

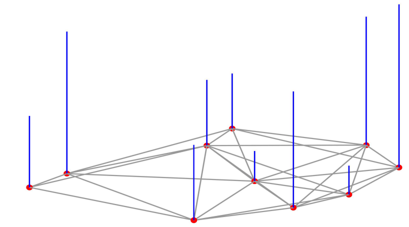
- Graph signal processing provides an innovative framework to handle data residing on networks.
- Polynomial graph filters and their inverses play important roles in graph signal processing vs. FIR (finite impulse response) and IIR filters in classical signal processing.
- The concept of commutative graph shifts plays a similar role in graph signal processing as the one-order delay in classical multi-dimensional signal processing.
- Consider the filtering and inverse filtering procedure associated polynomial filters of multiple commutative shifts and also iterative approximation algorithms and the associated distributed optimization problems.
- Mainly based on the paper “Polynomial graph filters of multiple shifts and distributed implementation of inverse filtering" with N. Emirov, C. Cheng and J. Jiang, submitted to *Sampling Theory, Signal Processing, and Data Analysis*

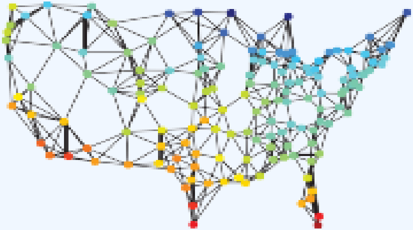
GRAPH SIGNAL PROCESSING AND DATA SCIENCE ON NETWORKS

- **Networks** have been widely used in many real world applications, including (wireless) sensor networks, smart grids, social network and epidemic spreading [1]
- The topological structures of networks could be described by some **graphs** $\mathcal{G} = (V, E)$ with vertices in V representing agents and edges in E between two vertices indicating the availability of a peer-to-peer communication between agents, or the functional connectivity between neural regions in brain, or the correlation between temperature records of neighboring weather stations.

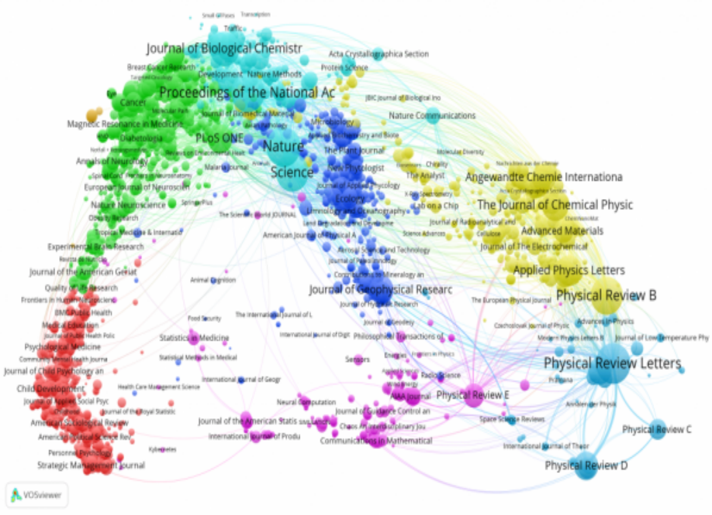
[1] R. Hebner, The power grid in 2030, IEEE Spectrum, 54 (2017), pp. 50–55. A. Ortega, P. Frossard, J. Kovacevic, J. M. F. Moura, and P. Vandergheynst, Proceedings of the IEEE, 106 (2018), pp. 808–828. D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, IEEE Signal Processing Magazine, 30 (2013), pp. 83–98. C. Cheng, Y. Jiang, and Q. Sun, *Appl. Comput. Harmon. Anal.*, vol. 47, pp. 109-148, 2019

- **A data set** on the network can be described by a **signal** on the graph $\mathcal{G} = (V, E)$, i.e., a vector $\mathbf{x} = (x_v)_{v \in V}$ residing on the nodes, where x_v represent the real/complex/vector-valued data at the knot/agent $v \in V$.





Visualize a citation network of 5000 journals with the largest number of citation links with other journals from all fields of science in the period 1980-2016 in VOSviewer. [2]



[2]https://www.cwts.nl/blog?article=n-r2r294

- **Data processing** on the network can be described by a **graph signal processing**, usually represented by a (non)linear function $\mathbf{x} \longmapsto A(\mathbf{x})$, or in graph signal processing, non(linear) filtering procedure on the signal space.
- Graph signal processing provides an *innovative framework* to handle data residing on various networks and many irregular domains.
- By leveraging graph spectral theory and applied harmonic analysis, many concepts in classical Euclidean setting have been extended to graph setting, such as **graph Fourier transform**, graph wavelet transform and nonsubsampled filter banks, in recent years. [3]
- Objective of this talk is on polynomial filters of multiple commutative shifts: distributed implementation and inverses.

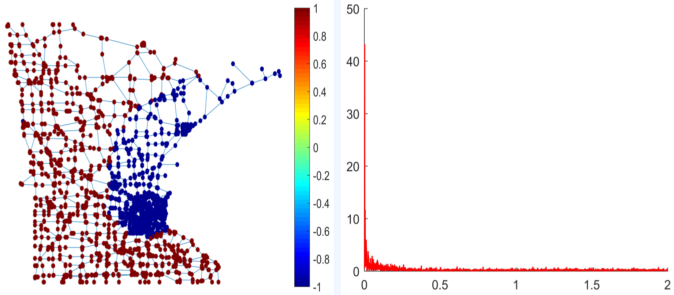
[3] Book chapter: Introduction to Graph Signal Processing by L. Stankovic, M. Dakovic and E. Sejdic, Spring 2018; Special Issue: Sampling Signals on Graphs: From Theory to Applications, IEEE Signal Processing Magazine, November 2020; and Special Issue on Harmonic Analysis on Graphs, JFAA, 2021.

Graph Laplacian and graph Fourier transform

- Graph $\mathcal{G} := (V; E)$ provides a flexible model to represent complicated relationships between data on networks, where $V = \{1, \ldots, N\}$ and $E \subset V \times V$.
- **Adjacency matrix** $\mathbf{A} = (a(i; j))_{i,j \in V}$ of an undirected graph $\mathcal{G} = (V; E)$, where $a(i,j) > 0$ if and only if $(i,j) \in E$. Unweighted graph: $a(i,j) = 1$ if $(i,j) \in E$.
- **Degree matrix** $\mathbf{D} = \mathrm{diag}(d_i)_{i \in V}$ with $d_i = \sum_{j \in V} a(i,j)$.
- **Laplacian matrix** $\mathbf{L} = \mathbf{D} - \mathbf{A}$ (all eigenvalues are nonnegative).
- **Symmetric normalized Laplacian** $\mathbf{L}^{\mathrm{sym}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$ (all eigenvalues contained in $[0, 2]$); Random walk normalized Laplacian $\mathbf{L}^{\mathrm{rw}} = \mathbf{D}^{-1}\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$.

Graph Fourier transform

- Write $\mathbf{L} = \mathbf{U}^T \Lambda \mathbf{U} = \sum_{i=1}^{N} \lambda_i u_i \mathbf{u}_i^T$ where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_N]$ is an orthogonal matrix, $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ is a diagonal matrix (eigendecompostion)
- **Graph Fourier transform** of a graph signal x is by $\hat{x} = \mathbf{U}\mathbf{x}$ and the **inverse graph Fourier transform** is $\mathbf{x} = \mathbf{U}^T \hat{\mathbf{x}}$. (Finite Fourier transform for the cycle graph)



Piecewise constant signal on Minnesota traffic graph and its Fourier transform in magnitudes.

- **Phase retrieval**: How to find real/complex/vector-valued graph signals x in some linear space so that they can determined, up to a trivial ambiguity, from magnitude $|\hat{x}|$ of their Fourier measurements or $|\langle \psi, x\rangle|, \psi \in \Psi$ of their frame measurements?
- Chen, Cheng and S. made some contribution on the recovery of a velocity field on graphs from from absolute speed (distance) at each vertex and relative speed (distance) of neighboring vertices. It is closely related to the classical **Euclidean distance geometry** (EDG) used in molecular conformation in computational chemistry, localization of wireless sensor networks, dimensionality reduction in machine learning, statistics of multidimensional scaling etc.[4]
- Phase retrieval for real/complex/vector-valued graph signals is an **inverse problem widely open** for further study.

[4] Y. CHEN, C. CHENG AND S., PHASE RETRIEVAL OF COMPLEX AND VECTOR-VALUED FUNCTIONS, submitted

Commutative graph shifts

- On a connected undirected graph $\mathcal{G}$, **geodesic distance** $\rho(i,j)$ between vertices i and $j \in V$ is the number of edges in the shortest path to connect i and j. Using the geodesic distance ρ, we denote the set of all R-**neighbors** of a vertex $i \in V$ by

$$B(i,R) = \{j \in V,\ \rho(j,i) \leq R\}.$$

- **Graph shifts S** $= (s(i,j))_{i,j \in V}$ if $s(i,j) = 0$ if $\rho(i,j) \geq 2$.
- Examples: Adjacent matrix, Laplacian, symmetric (random walk) normalized Laplacian matrix, and more
- **Commutative graph shifts** $\mathbf{S}_1, \ldots, \mathbf{S}_d$ if

$$\mathbf{S}_i\mathbf{S}_j = \mathbf{S}_j\mathbf{S}_i,\ 1 \leq i,j \leq d. \tag{1}$$

- **Commutative graph shifts $\mathbf{S}_1, \ldots, \mathbf{S}_d$** if $\mathbf{S}_i\mathbf{S}_j = \mathbf{S}_j\mathbf{S}_i$ for all $1 \le i, j \le d$.
- **Simultaneous upper-triangularization** [5]:

Proposition

There is a unitary matrix $\mathbf{U}$ such that $\hat{\mathbf{S}}_k = \mathbf{U}^{\mathrm{H}}\mathbf{S}_k\mathbf{U}, 1 \le k \le d$ are upper-triangular matrices with diagonal entries $\hat{S}_k(i,i), i \in V$ (eigenvalues of $\mathbf{S}_k$).

- **Joint spectrum**

$$\Lambda = \left\{\lambda_i = \left(\widehat{S}_1(i,i), ..., \widehat{S}_d(i,i)\right), 1 \le i \le N\right\}. \tag{2}$$

- If $\mathbf{S}_1, \ldots, \mathbf{S}_d$ are **simultaneously triangularizable**, i.e., $\mathbf{S}_k = \mathbf{V}^{-1}\tilde{\mathbf{S}}_k\mathbf{V}, 1 \le k \le d$, where $\tilde{\mathbf{S}}_k$ are diagonal matrices. Then

$$\mathbf{S}_i\mathbf{S}_j = \mathbf{V}^{-1}\tilde{\mathbf{S}}_i\tilde{\mathbf{S}}_j\mathbf{V} = \mathbf{V}^{-1}\tilde{\mathbf{S}}_j\tilde{\mathbf{S}}_i\mathbf{V} = \mathbf{S}_j\mathbf{S}_i, \ 1 \le i, j \le d,$$

$\Longrightarrow \mathbf{S}_k, 1 \le k \le d$, are commutative.

[5] R. A. Horn and C. R. Johnson. *Matrix Analysis*, Cambridge University Press, 2012

UCF

Commutative graph shifts: Example 1

- **Circurant graph** $\mathcal{C}(N, Q) = (V_N, E_N(Q))$ generated by $Q = \{q_1, \ldots, q_M\}$, where $1 \le q_i < N/2$, $V_N = \{0, 1, \ldots, N-1\}$ and $E_N(Q) = \cup_{1\le k\le d}\big\{(i, i \pm q_k \mod N), i \in V_N\big\}$.
- The circulant graph $\mathcal{C}(N, Q)$ can be decomposed into a family of **cycle graphs** $\mathcal{C}(N, Q_k)$ generated by $Q_k = \{q_k\}, 1 \le k \le d$, (cycle graph) and the symmetric normalized Laplacian matrix $\mathbf{L}^{\rm sym}_{\mathcal{C}(N,Q)}$ on $\mathcal{C}(N, Q)$ is the average of symmetric normalized Laplacian matrices $\mathbf{L}^{\rm sym}_{\mathcal{C}(N,Q_k)}$ on $\mathcal{C}(N, Q_k), 1 \le k \le d$, i.e.,

$$\mathbf{L}^{\rm sym}_{\mathcal{C}(N,Q)} = \frac{1}{d}\sum_{k=1}^{d} \mathbf{L}^{\rm sym}_{\mathcal{C}(N,Q_k)}.$$

Proposition

The symmetric normalized Laplacian matrices $\mathbf{L}^{\rm sym}_{\mathcal{C}(N,Q_k)}$ of the circulant graphs $\mathcal{C}(N, Q_k), 1 \le k \le d$, are **commutative graph shifts** on the circulant graph $\mathcal{C}(N, Q)$.

UCF

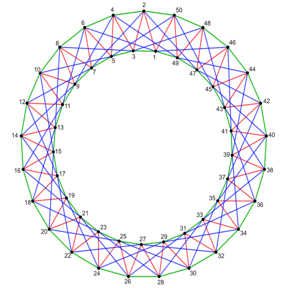


Figure: The circulant graph with 50 nodes and generating set $Q_0 = \{1, 2, 5\}$, where edges in red/green/blue are also edges of the circulant graphs $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_5$ generated by $\{1\}, \{2\}, \{5\}$ respectively.

- $\mathbf{L}^{\mathrm{sym}}_{\mathcal{C}(N,Q_k)}, 1 \leq k \leq d$ are **commutative graph shifts** on circulant graphs. Similar conclusions for **Cayley graphs**.

- Given two finite graphs $\mathcal{G}_1 = (V_1, E_1)$ and $\mathcal{G}_2 = (V_2, E_2)$ with adjacency matrices $\mathbf{A}_1$ and $\mathbf{A}_2$, define their **Cartesian product graph** $\mathcal{G}_1 \times \mathcal{G}_2$ has vertex set $V_1 \times V_2$ and adjacency matrix given by $\mathbf{A} = \mathbf{A}_1 \otimes \mathbf{I}_{\#V_2} + \mathbf{I}_{\#V_1} \otimes \mathbf{A}_2$. (Kronecker product)
- $\mathbf{L}_1^{\mathrm{sym}} \otimes \mathbf{I}_{\#V_2}$ and $\mathbf{I}_{\#V_1} \otimes \mathbf{L}_2^{\mathrm{sym}}$ are graph filters of the Cartesian product graph $\mathcal{G}_1 \times \mathcal{G}_2$, where $\mathbf{L}_i^{\mathrm{sym}}$ are symmetric normalized Laplacian matrices of the graph $\mathcal{G}_i, i = 1, 2$.

Proposition

$\mathbf{L}_1^{\mathrm{sym}} \otimes \mathbf{I}_{\#V_2}$ and $\mathbf{I}_{\#V_1} \otimes \mathbf{L}_2^{\mathrm{sym}}$ are commutative graph shifts of the Cartesian product graph $\mathcal{G}_1 \times \mathcal{G}_2$.

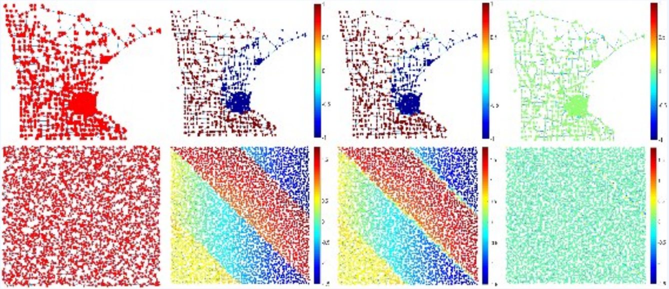
- $\mathbf{L}^{\mathrm{sym}}_{\mathcal{C}(N,Q_k)}$, $1 \le k \le d$ are **commutative graph shifts** on circulant graphs. Similar conclusions for Cayley graphs.
- $\mathbf{L}^{\mathrm{sym}}_{1} \otimes \mathbf{I}_{\#V_2}$ and $\mathbf{I}_{\#V_1} \otimes \mathbf{L}^{\mathrm{sym}}_{2}$ are commutative graph filters.
- An illustrative example of Cartesian product graph is for **time-varying data processing** on networks, where $\mathbf{L}^{\mathrm{sym}}_{1} \otimes \mathbf{I}_{\#V_2}$ and $\mathbf{I}_{\#V_1} \otimes \mathbf{L}^{\mathrm{sym}}_{2}$ have different features (regularity in the time/spatial domain, for instance, weather data including time and location, smart grids, social networks.
- The concept of commutative graph shifts $\mathbf{S}_1, \ldots, \mathbf{S}_d$ plays a similar role in graph signal processing as the **one-order delay** $z_1^{-1}, \ldots, \mathbf{z}_d^{-1}$ in classical multi-dimensional signal processing, and in practice graph shifts may have specific features and physical interpretation.

Polynomial filters and distributed implementation

- For a polynomial $p(t) = \sum_{m=0}^{M} p_m t^m$, define **polynomial filters**

$$p(\mathbf{L}^{\rm sym}) = p_0 \mathbf{I} + \sum_{m=1}^{M} p_m (\mathbf{L}^{\rm sym})^m.$$

- Spline filter banks $\mathbf{H}^{\rm spln}_{0,M} = (\mathbf{I} - \mathbf{L}^{\rm sym}/2)^M$ and $\mathbf{H}^{\rm spln}_{0,M} = (\mathbf{L}^{\rm sym}/2)^M,\ M \geq 1.$ [6]



[6] M. S. Kotzagiannidis and P. L. Dragotti, Appl. Comput. Harmon. Anal, 47 (2019), 539-565; Junzheng Jiang, Cheng Cheng and Qiyu Sun, IEEE Transactions on Signal Processing, 67(2019), 3938 - 3953.

- For a polynomial $p(t)=\sum_{m=0}^{M} p_m t^m$, define polynomial filters $p(\mathbf{L}^{\mathrm{sym}})=p_0\mathbf{I}+\sum_{m=1}^{M} p_m(\mathbf{L}^{\mathrm{sym}})^m$.
- Given a commutative graph shifts $\mathbf{S}_1,\ldots,\mathbf{S}_d$ and a multivariate polynomial $h(t_1,\ldots,t_d)=\sum c_{m_1,\ldots,m_d} t_1^{m_1}\ldots t_d^{m_d}$, define **polynomial filter of multiple graph shifts**

$$h(\mathbf{S}_1,\ldots,\mathbf{S}_d)=\sum c_{m_1,\ldots,m_d}\mathbf{S}_1^{m_1}\ldots\mathbf{S}_d^{m_d}.$$

  (The polynomial filter is well-defined due to the commutativity, for instance t_1t_2 and t_2t_1 are the same polynomial) [7]
- **Geodesic-width** $\omega(\mathbf{H})$ of a graph filter $\mathbf{H}=(H(i,j))_{i,j\in V}$ is the smallest nonnegative integer $\omega(\mathbf{H})$ such that $H(i,j)=0$ hold for all $i,j\in V$ with $\rho(i,j)>\omega(\mathbf{H})$. cf. Finite response filter (FIR) in classical signal processing.
- **Conclusion**: For a polynomial filter **H**, its geodesic-width is no more than its degree.

[7] Nazar Emirov, Cheng Cheng, Junzheng Jiang and S. “Polynomial graph filters of multiple shifts and distributed implementation of inverse filtering" submitted to *Sampling Theory, Signal Processing, and Data Analysis*

- Given a filter $\mathbf{H} = (H(i,j))_{i,j\in V}$ with geodesic-width ω, the filtering procedure $(x_i)_{i\in V} =: \mathbf{x} \longmapsto \mathbf{Hx} = \mathbf{y} = (y_i)_{i\in V}$ can be **implemented at the vertex level,**

$$y_i = \sum_{j\in V} H(i,j)x_j = \sum_{\rho(j,i)\leq\omega} H(i,j)x_j.$$

  (For $i \in V$, receive data $x_j, j \in B(i,\omega)$ and then evaluate).
- For a polynomial filter $p(\mathbf{L}^{\mathrm{sym}}) = p_0\mathbf{I} + \sum_{m=1}^{M} p_m(\mathbf{L}^{\mathrm{sym}})^m$,

$$\mathbf{y} \ = \ p_0\mathbf{x} + \mathbf{L}^{\mathrm{sym}}(p_1\mathbf{x} + \cdots + \mathbf{L}^{\mathrm{sym}}(p_{M-2}\mathbf{x} + (p_{M-1}\mathbf{x} + p_M\mathbf{L}^{\mathrm{sym}}\mathbf{x})).$$

- Iterative one-hop implementation (each vertex communicate with neighboring vertex only)

$$x_1 = p_{M-1}\mathbf{x}+p_M\mathbf{L}^{\mathrm{sym}}\mathbf{x},\ \mathbf{x}_{i+1} = p_{M-i-1}\mathbf{x}+\mathbf{L}^{\mathrm{sym}}\mathbf{x}_i, 1 \leq i \leq M-1, \mathbf{x}_M = \mathbf{y}.$$

- Emirov, Cheng, Jiang and S. proposed an one-hop implementation for the filtering procedure associated with polynomial filter of multiple graph shifts $h(\mathbf{S}_1,\ldots,\mathbf{S}_d) = \sum c_{m_1,\ldots,m_d}\mathbf{S}_1^{m_1}\ldots\mathbf{S}_d^{m_d}$.

UCF

INVERSE OF POLYNOMIAL FILTERS AND DISTRIBUTED IMPLEMENTATION

- Consider the inverse $\mathbf{H}^{-1}$ of polynomial filters of multiple graph shifts $\mathbf{H} = h(\mathbf{S}_1, \ldots, \mathbf{S}_d) = \sum c_{m_1,\ldots,m_d} \mathbf{S}_1^{m_1} \ldots \mathbf{S}_d^{m_d}$.
- **Inverse filtering** associated with the graph filter having small geodesic-width plays an important role in graph signal processing, such as denoising, graph semi-supervised learning, non-subsampled filter banks and signal reconstruction.
- $\mathbf{G}_0 = \mathbf{H}^{-1}\mathbf{H}_0^T$ and $\mathbf{G}_1 = \mathbf{H}^{-1}\mathbf{H}_1^T$, where $\mathbf{H} = \mathbf{H}_0^T\mathbf{H}_0 + \mathbf{H}_1^T\mathbf{H}_1$ in **nonsubsampled filter banks**.

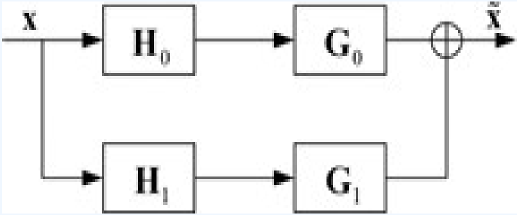


Figure: Block diagram of an NSGFB with analysis filter bank (H_0, H_1) and synthesis filter bank (G_0, G_1), where x is the input of the NSGFB and $\tilde{x}$ is its output.

- **Minimization problem** $\min_{\mathbf{x}} \|\mathbf{Hx}-\mathbf{b}\|_2^2+\lambda\|\mathbf{L}^{\rm sym}x\|_2^2$ or in general

$$\min_{\mathbf{x}} F(\mathbf{x}) = \sum_{j\in V} f_j(\mathbf{x})$$

where local objective functions f_j depend only on neighboring vertices $\mathbf{x}_j, j\in B(i,m)$. [8]
- The **challenge** arisen in the inverse filtering is on its implementation, as the inverse filter $\mathbf{H}^{-1}$ usually has full geodesic-width even if the original filter $\mathbf{H}$ has small geodesic-width.
- For the case that the filter $\mathbf{H}$ is strictly positive definite, the inverse filtering procedure $b\longmapsto \mathbf{H}^{-1}\mathbf{b}$ can be implemented by applying the iterative **gradient descent method** in a distributed network,

$$x_{n+1}=x_n-\beta(\mathbf{H}\mathbf{x}_n-\mathbf{b}), n\geq 1$$

when the step size β is appropriately selected.

[8] N. Emirov, G. Song and S., A Divide-and-Conquer Algorithm for Distributed Optimization on Networks, in preparation

- If $\mathbf{H}$ is strictly positive definite, the inverse filtering procedure $b \longmapsto \mathbf{H}^{-1}\mathbf{b}$ cab be solved by the iterative **gradient descent method**, $x_{n+1} = x_n - \beta(\mathbf{H}\mathbf{x}_n - \mathbf{b}), n \geq 1$.
- To consider implementation of inverse filtering of an arbitrary invertible filter H, we select a graph filter G with small geodesic-width to approximate $\mathbf{H}^{-1}$, $\rho(I - GH) < 1$, and propose the following iterative algorithm to implement the inverse filtering procedure:

$$\begin{cases} z^{(m)} = \mathbf{G}\mathbf{e}^{(m-1)} \\ e^{(m)} = e^{(m-1)} - Hz^{(m)} \\ x^{(m)} = x^{(m-1)} + z^{(m)}, m \geq 1 \end{cases} \tag{3}$$

with initial $e^{(0)} = \mathbf{b}$ and $x^{(0)} = 0$.
- **Conclusion**: $x^{(m)}$ converges to $\mathbf{H}^{-1}\mathbf{b}$ exponentially.

UCF

- If $\rho(I - GH) < 1$, then $x^{(m)}$ in the iterative algorithm

$$\begin{cases} z^{(m)} = \mathbf{G}e^{(m-1)} \\ e^{(m)} = e^{(m-1)} - Hz^{(m)} \\ x^{(m)} = x^{(m-1)} + z^{(m)}, m \geq 1 \end{cases} \quad (4)$$

  with initial $e^{(0)} = \mathbf{b}$ and $x^{(0)} = 0$, converges to $\mathbf{H}^{-1}\mathbf{b}$ exponentially.
- **Problem**: How to choose **G** with small geodesic-width or polynomial filters of small degree so that $\rho(I - GH) < 1$?
- Recall **Joint spectrum** of Commutative graph shifts $\mathbf{S}_1, \ldots, \mathbf{S}_d$:

$$\Lambda = \left\{\boldsymbol{\lambda}_i = \left(\widehat{S}_1(i,i), ..., \widehat{S}_d(i,i)\right), 1 \leq i \leq N\right\}.$$

- **Observation**:

$$\rho(\mathbf{I} - \mathbf{GH}) = \max_{\lambda \in \Lambda} |1 - g(\lambda)h(\lambda)|$$

  for polynomial filters $\mathbf{G} = g(\mathbf{S}_1, \ldots, \mathbf{S}_d)$ and $\mathbf{H} = h(\mathbf{S}_1, \ldots, \mathbf{S}_d)$

UCF

- **Problem**: How to choose **G** with geodesic-width so that $\rho(I - GH) < 1$?
- Recall **Joint spectrum** of Commutative graph shifts $\mathbf{S}_1, \ldots, \mathbf{S}_d$:

$$\Lambda = \left\{\lambda_i = \left(\widehat{S}_1(i,i), ..., \widehat{S}_d(i,i)\right), 1 \le i \le N\right\}. \qquad (5)$$

Observation: $\rho(\mathbf{I} - \mathbf{GH}) = \max_{\lambda \in \Lambda} |1 - g(\lambda)h(\lambda)|$ for polynomial filters $\mathbf{G} = g(\mathbf{S}_1, \ldots, \mathbf{S}_d)$ and $\mathbf{H} = h(\mathbf{S}_1, \ldots, \mathbf{S}_d)$

- If $\mathbf{H} = h(\mathbf{S}_1, \ldots, \mathbf{S}_d)$ is a polynomial filter and the joint spectrum is known, we may select the optimal approximation filter $\mathbf{G}_{O,n} = g_{o,n}(\mathbf{S}_1, \ldots, \mathbf{S}_d)$ as follows:

$$g_{O,n} = \arg\min_{g \in \mathcal{P}_n} \max_{\lambda \in \Lambda} |1 - g(\lambda)h(\lambda)|$$

where $\mathcal{P}_n$ is the space of all polynomial of degree at most n.

- **Conclusion**: If H is invertible, then $r_n := \rho(I - \mathbf{G}_{O,n}\mathbf{H}) = \max_{\lambda \in \Lambda} |1 - g_n(\lambda)h(\lambda)|$ is a decreasing sequence with $r_N = 0$, and hence the proposed iterative algorithm converges exponentially for all $n \ge n_0$.
- **Conclusion**: Every iteration can be one-hop implemented at the vertex level.

- **Problem**: How to choose **G** with geodesic-width so that $\rho(I - GH) < 1$?
- If $\mathbf{H} = h(\mathbf{S}_1, \dots, \mathbf{S}_d)$ is a polynomial filter and the joint spectrum is known, we may select the approximation filter $\mathbf{G}_n = g_n(S_1, \dots, S_d)$, where $g_n = \arg\min_{g \in \mathcal{P}_n} \max_{\lambda \in \Lambda} |1 - g(\lambda)h(\lambda)|$.
- For a graph G of large order, it is often computationally expensive to find the joint spectrum Λ exactly. However, the graph shifts $S_k, 1 \leq k \leq d,$, in some engineering applications are symmetric and their spectrum sets are known being contained in some intervals. For instance, the normalized Laplacian matrix on a simple graph is symmetric and its spectrum is contained in $[0, 2]$.

- Assume that commutative shifts $S_1, \ldots, S_d$ is contained in a cube. Let $g_K, K \geq 0$ be the multivariate Chebyshev polynomial approximation to $(h(t_1, \ldots, t_d))^{-1}$ and denote $G_K = g_K(S_1, \ldots, S_d)$.
- Observation:

$$\sup_{t \in Q} |1 - g_K h(t)| \leq C r^K, \; K \geq 1$$

for some $r \in (0, 1)$ and $C \in (0, \infty)$.

- For large K, the sequence $x^{(m)}, m \geq 1$ in the iterative algorithm

$$\begin{cases} z^{(m)} = \mathbf{G}_K \mathbf{e}^{(m-1)} \\ e^{(m)} = e^{(m-1)} - H z^{(m)} \\ x^{(m)} = x^{(m-1)} + z^{(m)}, m \geq 1 \end{cases} \qquad (6)$$

converges exponentially to $\mathbf{H}^{-1}\mathbf{b}$ and it can be implemented in one-hop communication in each iteration.

Distance between non-polynomial filters and polynomial filters

- For a polynomial filter $\mathbf{H}$ of commutative graph shifts $\mathbf{S}_1, \ldots, \mathbf{S}_d$, we have

$$[\mathbf{H}, \mathbf{S}_k] = \mathbf{H}\mathbf{S}_k - \mathbf{S}_k\mathbf{H} = 0, 1 \leq k \leq d$$

- How to estimate the distance

$$\mathrm{dist}(\mathbf{H}, \mathcal{P}) = \inf_{\mathbf{P} \in \mathcal{P}} \|\mathbf{H} - \mathbf{P}\|_F$$

between a graph filter $\mathbf{H}$ and the set $\mathcal{P}$ of all polynomial filters of commutative graph shifts $\mathcal{S}_1, \ldots, \mathbf{S}_d$.

Theorem

If the commutative graph shifts $\mathbf{S}_1, \ldots, \mathbf{S}_d$ can be diagonalized simultaneously by a unitary matrix and elements in their joint spectrum Λ are distinct, then

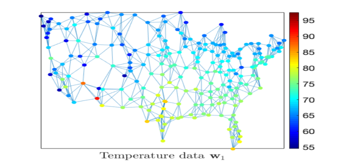
$$C_0 \Big(\sum_{k=1}^{d} \|[\mathbf{H}, \mathbf{S}_k]\|_F^2 \Big)^{1/2} \leq \mathrm{dist}(\mathbf{H}, \mathcal{P}) \leq C_1 \Big(\sum_{k=1}^{d} \|[\mathbf{H}, \mathbf{S}_k]\|_F^2 \Big)^{1/2}.$$

UCF

Numerical demonstrations

Denoising an hourly temperature dataset

- Denoising the hourly temperature dataset collected at 218 locations in the United States on August 1st, 2010, measured in Fahrenheit. The above real-world dataset is of size 218×24, and it can be modelled as a time-varying signal $w(i), 1 \le i \le 24$, on the product graph $\mathcal{C} \times \mathcal{W}$, where $\mathcal{C}$ is the circulant graph with 24 vertices and generator $\{1\}$, and $\mathcal{W}$ is the undirected graph with 218 locations as vertices and edges constructed by the 5 nearest neighboring algorithm.



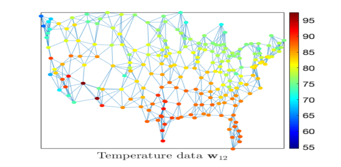


Figure: Presented on the left and right sides are the temperature data $\mathbf{w}_1$ and $\mathbf{w}_{12}$.

- Noisy temperature data

$$\widetilde{\mathbf{w}}_i = \mathbf{w}_i + \boldsymbol{\eta}_i, \ \ i = 1, \ldots, 24.$$

- We propose the following denoising approach,

$$\widehat{\mathbf{W}} := \arg\min_{\mathbf{Z}} \|\mathbf{Z} - \widetilde{\mathbf{W}}\|_2^2 + \tilde{\alpha}\mathbf{Z}^T(\mathbf{I} \otimes \mathbf{L}_{\mathcal{W}}^{\mathrm{sym}})\mathbf{Z} + \tilde{\beta}\mathbf{Z}^T(\mathbf{L}_{\mathcal{C}}^{\mathrm{sym}} \otimes \mathbf{I})\mathbf{Z}, \quad (7)$$

  where $\widetilde{\mathbf{W}}$ is the vectorization of the noisy temperature data $\widetilde{\mathbf{w}}_1, \ldots, \widetilde{\mathbf{w}}_{24}$ with noises $\boldsymbol{\eta}_i, 1 \leq i \leq 24$ having their components randomly selected in $[-\eta, \eta]$ in a uniform distribution, $\mathbf{L}_{\mathcal{W}}^{\mathrm{sym}}$ and $\mathbf{L}_{\mathcal{C}}^{\mathrm{sym}}$ are normalized Laplacian matrices on the graph $\mathcal{W}$ and $\mathcal{C}$ respectively, and $\tilde{\alpha}, \tilde{\beta} \geq 0$ are penalty constants in the vertex and temporal domains to be appropriately selected.

Presented in Table 1 are the average over 1000 trials of the input signal-to-noise ratio ISNR and the output signal-to-noise ratio

$$\mathrm{SNR}(m) = -20 \log_{10} \frac{\|\widehat{\mathbf{W}}^{(m)} - \mathbf{W}\|_2}{\|\mathbf{W}\|_2},\ m \geq 1,$$

which are used to measure the denoising performance of the IOPA1($\tilde{\alpha}, \tilde{\beta}$), ICPA1($\tilde{\alpha}, \tilde{\beta}$) and GD0($\tilde{\alpha}, \tilde{\beta}$) at the mth iteration, where $\widehat{\mathbf{W}}^{(\infty)} := \widehat{\mathbf{W}}$ and $\widehat{\mathbf{W}}^{(m)}, m \geq 1$, are outputs of the IOPA1($\tilde{\alpha}, \tilde{\beta}$) algorithm, or the ICPA1($\tilde{\alpha}, \tilde{\beta}$), or the GD0($\tilde{\alpha}, \tilde{\beta}$) at m-th iteration.

From Table 1, we see that the Tikhonov regularization on the temporal-vertex domain has **better performance** on denoising the hourly temperature dataset than the Tikhonov regularization **only** either on the vertex domain (i.e. $\tilde{\beta} = 0$) or on the temporal domain (i.e. $\tilde{\alpha} = 0$) do.

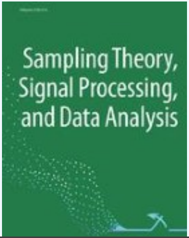
UCF

Table: The average over 1000 trials of the signal-to-noise ratio $\mathrm{SNR}(m), m = 1, 2, 4, 6, \infty$ denoise the US hourly temperature dataset collected at 218 locations on August 1st, 2010, where $\eta = 35, 20, 10$.

SNR \ m / Alg.	1	2	4	6	∞
η=10, ISNR=22.4320					
IOPA1($\tilde{\alpha}$, 0)	23.3572	24.5564	24.5565	24.5565	24.5565
IOPA1(0, $\tilde{\beta}$)	16.9511	25.9123	26.4291	26.4284	26.4284
IOPA1($\tilde{\alpha}$, $\tilde{\beta}$)	14.2863	24.9125	26.9961	26.9990	26.9990
ICPA1($\tilde{\alpha}$, 0)	22.5720	24.5572	24.5565	24.5565	24.5565
ICPA1(0, $\tilde{\beta}$)	18.6319	26.2493	26.4294	26.4285	26.4284
ICPA1($\tilde{\alpha}$, $\tilde{\beta}$)	12.7428	23.3488	26.9816	26.9989	26.9990
GD0($\tilde{\alpha}$, 0)	11.7089	21.2276	24.5387	24.5566	24.5565
GD0(0, $\tilde{\beta}$)	6.2342	12.3916	22.7545	26.1414	26.4284
GD0($\tilde{\alpha}$, $\tilde{\beta}$)	4.9806	9.9239	19.2003	25.2121	26.9990

Take home message

- GSP: an innovative framework to handle data residing on distributed networks.
- **Polynomial filters of (multiple) graph shifts**: important roles in graph signal processing vs. finite impulse response filter (FIR)
- **Distributed implementation** for the filtering and inverse filtering procedure.
- Welcome all to submit your work to the new journal *Sampling Theory, Signal Processing, and Data Analysis* edited by Akram Aldroubi, Zuhair Nashed, Götz Pfander.




THANK YOU
thanks
THANKS
danke
謝謝
شكرا
Gratias
THANKS
merci
Thanks
gracias