Iterative polynomial approximation algorithms for inverse graph filters

Cheng Cheng, Qiyu Sun, Cong Zheng

***Abstract*—Chebyshev interpolation polynomials exhibit the exponential approximation property to analytic functions on a cube. Based on the Chebyshev interpolation polynomial approximation, we propose iterative polynomial approximation algorithms to implement the inverse filter with a polynomial graph filter of commutative graph shifts in a distributed manner. The proposed algorithms exhibit exponential convergence properties, and they can be implemented on distributed networks in which agents are equipped with a data processing subsystem for limited data storage and computation power, and with a one-hop communication subsystem for direct data exchange only with their adjacent agents. Our simulations show that the proposed polynomial approximation algorithms may converge faster than the Chebyshev polynomial approximation algorithm and the conventional gradient descent algorithm do.**

Keywords: Graph inverse filter, polynomial approximation algorithm, distributed implementation, graph signal processing.

I. Introduction

Graphs are widely used to model the complicated topological structure of networks, such as (wireless) sensor networks, smart grids and social networks [1], [10], [13], [25], [26], [27], [30], [32]. Many data sets on a network can be represented by signals $\mathbf{x} = (x_i)_{i\in V}$ residing on the graph $\mathcal{G} = (V, E)$, where x_i represents the real/complex/vector-valued data at the vertex/agent $i \in V$, a vertex in V may represent an agent of the network, and an edge in E between vertices could indicate that the corresponding agents have a peer-to-peer communication link between them. Graph signal processing paves an innovative way to extract valuable insights from datasets residing on the complicated networks, [7], [8], [22], [23], [36].

The filtering procedure for signals on a network is a linear transformation

$$\mathbf{x} \longmapsto \mathbf{y} = \mathbf{H}\mathbf{x}, \tag{I.1}$$

which maps a graph signal $\mathbf{x}$ to another graph signal $\mathbf{y} = \mathbf{H}\mathbf{x}$, and $\mathbf{H} = (H(i,j))_{i,j\in V}$ is known as a *graph filter*. An *graph shift* is an elementary graph filter, and we say that a matrix $\mathbf{S} = (S(i,j))_{i,j\in V}$ on the graph $\mathcal{G} = (V,E)$ if $S(i,j) \neq 0$ only if either $j = i$ or $(i,j) \in E$. In [13], the notion of multiple commutative graph shifts $\mathbf{S}_1, \ldots, \mathbf{S}_d$ are introduced,

$$\mathbf{S}_k\mathbf{S}_{k'} = \mathbf{S}_{k'}\mathbf{S}_k,\ 1 \le k, k' \le d, \tag{I.2}$$

and some multiple commutative graph shifts on circulant/Cayley graphs and on Cartesian product graphs are constructed with physical interpretation. An important property for commutative graph shifts $\mathbf{S}_1, \ldots, \mathbf{S}_d$ is that they can be upper-triangularized simultaneously,

$$\widehat{\mathbf{S}}_k = \mathbf{U}^{\mathrm{H}}\mathbf{S}_k\mathbf{U},\ 1 \le k \le d, \tag{I.3}$$

where $\mathbf{U}$ is a unitary matrix and $\widehat{\mathbf{S}}_k = (\widehat{S}_k(i,j))_{1\le i,j\le N}, 1 \le k \le d$, are upper triangular matrices. As $\widehat{S}_k(i,i), 1 \le i \le N$, are eigenvalues of $\mathbf{S}_k, 1 \le k \le d$, we call the set

$$\Lambda = \left\{\boldsymbol{\lambda}_i = \left(\widehat{S}_1(i,i), ..., \widehat{S}_d(i,i)\right), 1 \le i \le N\right\} \tag{I.4}$$

as the *joint spectrum* of $\mathbf{S}_1, \ldots, \mathbf{S}_d$ [13]. For the case that graph shifts $\mathbf{S}_1, \ldots, \mathbf{S}_d$ are symmetric, all $\widehat{S}_k(i,i), 1 \le i \le N, 1 \le k \le d$ are real-valued and the joint spectrum of graph shifts $\mathbf{S}_1, \ldots, \mathbf{S}_d$ is contained in some cube,

$$\Lambda \subset [\boldsymbol{\mu}, \boldsymbol{\nu}] := [\mu_1, \nu_1] \times \cdots \times [\mu_d, \nu_d] \subset \mathbb{R}^d, \tag{I.5}$$

where for each $1 \le k \le d$, $[\mu_k, \nu_k]$ is the (minimal) interval to contain the spectrum of the graph shift $\mathbf{S}_k$.

A popular family of graph filters is *polynomial filters* of commutative graph shifts $\mathbf{S}_1, \ldots, \mathbf{S}_d$,

$$\mathbf{H} = h(\mathbf{S}_1, \ldots, \mathbf{S}_d) = \sum_{l_1=0}^{L_1} \cdots \sum_{l_d=0}^{L_d} h_{l_1,\ldots,l_d}\mathbf{S}_1^{l_1} \cdots \mathbf{S}_d^{l_d}, \tag{I.6}$$

where h is a multivariate polynomial in variables $t_1, \cdots, t_d$,

$$h(t_1, \ldots, t_d) = \sum_{l_1=0}^{L_1} \cdots \sum_{l_d=0}^{L_d} h_{l_1,\ldots,l_d} t_1^{l_1} \ldots t_d^{l_d}$$

[6], [9], [13], [16], [18], [21], [28], [31], [34]. The consideration of polynomial filters of multiple graph shifts is mainly motivated by signal processing of time-varying data sets on a sensor network over a period of time, which carry different correlation characteristics for spatial-temporal directions.

Inverse filtering procedure associated with a polynomial filter of graph shifts is a versatile tool, offering a wide applications across denoising, signal reconstruction, graph semi-supervised learning and many other applications [2], [3], [5], [6], [9], [16], [17], [21], [24], [29], [31]. Its importance lies in its ability to recover original signals from observed data, enabling a deeper understanding of underlying network structures and dynamics. In this paper, we consider distributed implementation of inverse filtering procedure on simple graphs (i.e., unweighted undirected graphs containing no loops or multiple edges) of large order $N \ge 1$.

Given a polynomial filter $\mathbf{H}$ of graph shifts, one of the main challenges in the corresponding inverse filtering procedure

$$\mathbf{y} \longmapsto \mathbf{x} = \mathbf{H}^{-1}\mathbf{y} \tag{I.7}$$

is on its distributed implementation, as the inverse filter $\mathbf{H}^{-1}$ is usually not a polynomial filter of small degree even if $\mathbf{H}$ is. The first two authors of this paper proposed the following exponentially convergent quasi-Newton method with arbitrary initial $\mathbf{x}^{(0)}$,

$$\mathbf{e}^{(m)}=\mathbf{H}\mathbf{x}^{(m-1)}-\mathbf{y} \ \text{ and } \ \mathbf{x}^{(m)}=\mathbf{x}^{(m-1)}-\mathbf{G}\mathbf{e}^{(m)},\ m\ge 1 \tag{I.8}$$

to fulfill the inverse filtering procedure, where the polynomial approximation filter $\mathbf{G}$ to the inverse $\mathbf{H}^{-1}$ is so chosen that the spectral radius of $\mathbf{I}-\mathbf{GH}$ is strictly less than 1 [5], [6], [13]. An important problem is how to select the polynomial approximation filter $\mathbf{G}$ appropriately for the fast convergence of the quasi-Newton method (I.8). The above problem has been well studied when $\mathbf{H}$ is a polynomial filter of the graph Laplacian (and a single graph shift in general) [3], [6], [15], [16], [29], [31]. For a polynomial filter $\mathbf{H}$ of multiple graph shifts, optimal/Chebyshev polynomial approximation filters are introduced in [13]. In Section II, we show the exponential approximation property of Chebyshev interpolation polynomials to analytic functions on a cube, and then we introduce the Chebyshev interpolation polynomial filter to approximate the inverse filter $\mathbf{H}^{-1}$ and the corresponding quasi-Newton method algorithm (III.2) to implement the inverse filtering procedure (I.7). In Section III, we show the corresponding Chebyshev interpolation polynomial algorithm is of exponential convergence and can be applied to implement the inverse filtering procedure (I.7) associated with a polynomial filter on distributed networks, see Theorem III.1 and Algorithms III.1. Numerical experiments in Section IV indicate that the proposed Chebyshev interpolation polynomial approximation algorithm have better performance than Chebyshev polynomial approximation algorithm (CPA), the gradient descent algorithm with optimal step size (OGDA) and the autoregressive moving-average model (ARMA) do [3], [13], [15], [16], [17], [29], [31], [34].

Notation: Bold lower cases and capitals are used to represent the column vectors and matrices respectively. Define $\|\mathbf{x}\|_2=(\sum_{i\in V}|x_i|^2)^{1/2}$ and $\|\mathbf{x}\|_\infty=\sup_{i\in V}\{|x_i|\}$ for a graph signal $\mathbf{x}=(x_i)_{i\in V}$ and $\|\mathbf{A}\|_2=\sup_{\|\mathbf{x}\|_2=1}\|\mathbf{A}\mathbf{x}\|_2$ for a graph filter $\mathbf{A}$. Denote the transpose, the Hermitian and Frobenius norm of a matrix $\mathbf{A}$ by $\mathbf{A}^T$, $\mathbf{A}^H$ and $\|\mathbf{A}\|_{\mathrm{F}}$ respectively.

II. Chebyshev interpolating polynomials

Let $[\boldsymbol{\mu},\boldsymbol{\nu}]=[\mu_1,\nu_1]\times\cdots\times[\mu_d,\nu_d]$ be a cube in $\mathbb{R}^d$ and h be a multivariate polynomial that does not vanish on the cube $[\boldsymbol{\mu},\boldsymbol{\nu}]$, i.e.,

$$h(\mathbf{t})\neq 0 \ \text{ for all } \ \mathbf{t}\in[\boldsymbol{\mu},\boldsymbol{\nu}]. \tag{II.1}$$

Write $\mathbf{t}_{\mathbf{j};\boldsymbol{\mu},\boldsymbol{\nu}}=(t_{j_1;\mu_1,\nu_1},\ldots,t_{j_d;\mu_d,\nu_d}), 0\le j_k\le M, 1\le k\le d$, be rescaled Chebyshev points in the cube $[\boldsymbol{\mu},\boldsymbol{\nu}]$, and the Lagrange basis at rescaled Chebyshev points be defined by

$$\ell_M(\mathbf{t},\mathbf{t}_{\mathbf{j};\boldsymbol{\mu},\boldsymbol{\nu}})=\prod_{k=1}^{d}\prod_{0\le i_k\le M, i_k\neq j_k}\frac{t_k-t_{i_k;\mu_k,\nu_k}}{t_{j_k;\mu_k,\nu_k}-t_{i_k;\mu_k,\nu_k}},$$

where

$$t_{j_k;\mu_k,\nu_k}=\frac{\nu_k+\mu_k}{2}+\frac{\nu_k-\mu_k}{2}\cos\frac{(j_k+1/2)\pi}{M+1}.$$

TABLE I: Shown are the maximal approximation errors of Jacobi polynomial approximations, Chebyshev polynomial approximation and Chebyshev interpolation polynomial approximation to $1/h_1$ on $[0,2]$ with the polynomial degree $0\le M\le 4$.

M / (α,β)	0	1	2	3	4
Cheby. Poly.	1.0463	0.5837	0.2924	0.1467	0.0728
(1/2, 1/2)	0.7014	0.5904	0.3897	0.2505	0.1517
ChebyInt	0.7500	0.4497	0.2342	0.1186	0.0595

For a polynomial h satisfying (II.1), an excellent method of approximating the reciprocal $1/h$ on the cube $[\boldsymbol{\mu},\boldsymbol{\nu}]$ is the Chebyshev interpolation polynomial

$$C_M(\mathbf{t})=\sum_{\|\mathbf{j}\|_\infty\le M}\frac{1}{h(\mathbf{t}_{\mathbf{j};\boldsymbol{\mu},\boldsymbol{\nu}})}\ell_M(\mathbf{t},\mathbf{t}_{\mathbf{j};\boldsymbol{\mu},\boldsymbol{\nu}}), \tag{II.2}$$

which is the unique polynomial of the form $\sum_{\|\mathbf{n}\|_\infty\le M} d_{\mathbf{n}}\mathbf{t}^{\mathbf{n}}$ for some $\mathbf{d}_{\mathbf{n}}, \|\mathbf{n}\|_\infty\le M$, satisfying the interpolation property

$$C_M(\mathbf{t}_{\mathbf{j};\boldsymbol{\mu},\boldsymbol{\nu}})=\frac{1}{h(\mathbf{t}_{\mathbf{j};\boldsymbol{\mu},\boldsymbol{\nu}})} \ \text{ for all } \|\mathbf{j}\|_\infty\le M. \tag{II.3}$$

Recall that the Lebesgue constant for the above polynomial interpolation at rescaled Chebyshev points is of the order $(\ln(M+2))^d$ [4]. This together with the exponential convergence of Chebyshev polynomial approximation, see [33, Theorem 8.2] and [35, Theorem 2.2], implies that

$$\tilde b_M:=\sup_{\mathbf{t}\in[\boldsymbol{\mu},\boldsymbol{\nu}]}|1-h(\mathbf{t})C_M(\mathbf{t})|\le D_1 r_1^M,\ M\ge 0, \tag{II.4}$$

for some positive constants $D_1\in(0,\infty)$ and $r_1\in(0,1)$.

Presented at the bottom row of Table I is the maximal approximation error $\tilde b_M, 0\le M\le 4$, of the Chebyshev interpolation polynomial C_M on $[0,2]$, ChebyInt for abbreviation, to the reciprocal of the univariate function

$$h_1(t)=(9/4-t)(3+t),\ t\in[0,2] \tag{II.5}$$

in [13, Eqn. (5.4)]. We observe that the Chebyshev interpolation polynomial approximation outperforms the Jacobi polynomial approximations with $\alpha=\beta=1/2$ in [37] and the Chebyshev polynomial approximation.

III. Chebyshev interpolation approximation algorithm

Let h be a multivariate polynomial satisfying (II.1), and $C_M, M\ge 0$, be the Chebyshev interpolation polynomial approximation to $1/h$ in (II.2). Set $\mathbf{H}=h(\mathbf{S}_1,\ldots,\mathbf{S}_d)$ and $\mathbf{C}_M=C_M(\mathbf{S}_1,\ldots,\mathbf{S}_d), M\ge 0$. By the spectral assumption (I.5), the spectral radii of $\mathbf{I}-\mathbf{C}_M\mathbf{H}$ are bounded by $\tilde b_M$ in (II.4) respectively, i.e.,

$$\rho(\mathbf{I}-\mathbf{C}_M\mathbf{H})\le\tilde b_M,\ M\ge 0. \tag{III.1}$$

Therefore with appropriate selection of the polynomial degree M, we obtain the exponential convergence of the following iterative polynomial approximation algorithm for inverse filtering,

$$\begin{cases}\mathbf{e}^{(m)}=\mathbf{H}\mathbf{x}^{(m-1)}-\mathbf{y}\\ \mathbf{x}^{(m)}=\mathbf{x}^{(m-1)}-\mathbf{C}_M\mathbf{e}^{(m)},\ m\ge 1\end{cases} \tag{III.2}$$

with arbitrary initials $\mathbf{x}^{(0)}$, where the input $\mathbf{y}$ is obtained via the filtering procedure (I.1).

Theorem III.1. *Let $\mathbf{S}_1, \ldots, \mathbf{S}_d$ be commutative graph shifts satisfying* (I.5), *h be a multivariate polynomial satisfying* (II.1), *and let $\tilde{b}_M$ be given in* (II.4). *If $\tilde{b}_M < 1$, then for any input $\mathbf{y}$, the sequence $\mathbf{x}^{(m)}, m \geq 0$, in the iterative algorithm* (III.2) *converges to the output $\mathbf{H}^{-1}\mathbf{y}$ of the inverse filtering procedure* (I.7) *exponentially. In particular, for any $r \in (\rho(\mathbf{I} - \mathbf{C}_M\mathbf{H}), 1)$ there exists a positive constant C such that*

$$\|\mathbf{x}^{(m)} - \mathbf{H}^{-1}\mathbf{y}\|_2 \leq C\|\mathbf{x}^{(0)} - \mathbf{H}^{-1}\mathbf{y}\|_2 r^m, \ m \geq 0. \quad \text{(III.3)}$$

Proof. As $\mathbf{S}_1, \ldots, \mathbf{S}_d$ are commutative graph shifts, $1/h$ is analytic on the joint spectrum $[\boldsymbol{\mu}, \boldsymbol{\nu}]$, we have

$$\begin{aligned} \mathbf{x}^{(m)} - \mathbf{H}^{-1}\mathbf{y} &= (\mathbf{I} - \mathbf{C}_M\mathbf{H})^m(\mathbf{x}^{(0)} - \mathbf{H}^{-1}\mathbf{y}), \\ &- \sum_{n=m+1}^{\infty} (\mathbf{I} - \mathbf{C}_M\mathbf{H})^n \mathbf{C}_M\mathbf{y}) \quad \text{(III.4)} \end{aligned}$$

from the iterative algorithm (III.2). By the spectral assumption (I.5) on commutative graph shifts $\mathbf{S}_1, \ldots, \mathbf{S}_d$, the spectral radii of $\mathbf{I} - \mathbf{C}_M\mathbf{H}$ is bounded by $\tilde{b}_M$ in (II.4), i.e.,

$$\rho(\mathbf{I} - \mathbf{C}_M\mathbf{H}) \leq \tilde{b}_M. \quad \text{(III.5)}$$

By Gelfand's formula on spectral radius, there exists a positive constant C for any $r \in (\rho(\mathbf{I} - \mathbf{C}_M\mathbf{H}), 1)$ such that

$$\|(\mathbf{I} - \mathbf{C}_M\mathbf{H})^m\|_2 \leq Cr^m, \ n \geq 1. \quad \text{(III.6)}$$

From (III.4) and (III.6), it follows that

$$\begin{aligned} \|\mathbf{x}^{(m)} - \mathbf{H}^{-1}\mathbf{y}\|_2 &\leq \|(\mathbf{I} - \mathbf{C}_M\mathbf{H})^m\|_2\|\mathbf{x}^{(0)} - \mathbf{H}^{-1}\mathbf{y}\|_2 \\ &\leq Cr^m\|\mathbf{x}^{(0)} - \mathbf{H}^{-1}\mathbf{y}\|_2, m \geq 0. \end{aligned}$$

This proves the exponential convergence of $\mathbf{x}^{(m)}, m \geq 0$. □

We call the iterative polynomial approximation algorithm (III.2) as *Chebyshev interpolation polynomial approximation algorithm*, CIPA for abbreviation. We remark that in each iteration in CIPA contains essentially two filtering procedures associated with polynomial filters $\mathbf{C}_M$ and $\mathbf{H}$, and hence it can be implemented at the vertex level with one-hop communication, see Algorithm III.1. Therefore the CIPA algorithms can be implemented on a distributed network with each agent equipped with limited storage and data processing ability, and one-hop communication subsystem. More importantly, the memory, computational cost and communication expense for each agent of the network are **independent** on the size of the whole network.

Algorithm III.1 The CIPA algorithm to implement the inverse filtering procedure $\mathbf{y} \longmapsto \mathbf{H}^{-1}\mathbf{y}$ at a vertex $i \in V$.

Inputs: Polynomial coefficients of polynomial filters $\mathbf{H}$ and $\mathbf{C}_M$, entries $S_k(i,j), j \in \mathcal{N}_i$ in the i-th row of the shifts $\mathbf{S}_k, 1 \leq k \leq d$, the value $y(i)$ of the input signal $\mathbf{y} = (y(i))_{i \in V}$ at the vertex i, and number m of iteration.
Initialization: Initial $e^{(0)}(i) = y(i)$, $x^{(0)}(i) = 0$ and $n = 0$.
Iteration: Use the iteration in [13, Algorithm 4] except replacing $\widetilde{\mathbf{G}}_L$ by $\mathbf{C}_M$, and the output is $x^{(n)}(i)$.
Output: The approximated value $x(i) \approx x^{(m)}(i)$ is the output signal $\mathbf{H}^{-1}\mathbf{y} = (x(i))_{i \in V}$ at the vertex i.

IV. Numerical Experiments

Circulant graphs are widely used in image processing [11], [12], [13], [19], [20]. Our numerical results show that the CIPA have impressive performances to implement the inverse filtering procedure than the Chebyshev polynomial approximation algorithm in [13] and the gradient descent method in [29] do. Some Tikhonov regularization problem can be converted to an inverse filtering procedure [13], [14], we also demonstrate the denoising performance of the polynomial approximation algorithms to the walking dog dataset.

A. Polynomial approximation algorithms

Let $N \geq 1$ and we say that $a = b \bmod N$ if $(a-b)/N$ is an integer. The *circulant graph* $\mathcal{C}(N, Q)$ generated by $Q = \{q_1, \ldots, q_L\}$ is a simple graph with the vertex set $V_N = \{0, 1, \ldots, N-1\}$ and the edge set $E_N(Q) = \{(i, i \pm q \bmod N), \ i \in V_N, q \in Q\}$, where $q_l, 1 \leq l \leq L$, are integers contained in $[1, N/2)$. Let $Q_0 = \{1, 2, 5\}$ and the polynomial filters be $\mathbf{H}_1 = h_1(\mathbf{L}^{\rm sym}_{\mathcal{C}(N,Q_0)})$, the input signal $\mathbf{x}$ have i.i.d. entries randomly selected in $[-1, 1]$, and the input signal $\mathbf{y} = \mathbf{H}_1\mathbf{x}$, where $h_1(t) = (9/4 - t)(3 + t)$ in (II.5), and $\mathbf{L}^{\rm sym}_{\mathcal{C}(N,Q_0)}$ is the symmetric normalized Laplacian on the circulant graph $\mathcal{C}(N, Q_0)$. Shown in Table II are averages of the relative iteration error

$$\mathrm{E}(m) = \frac{\|\mathbf{x}^{(m)} - \mathbf{x}\|_2}{\|\mathbf{x}\|_2}, \ m \geq 1,$$

over 1000 trials to implement the inverse filtering procedure $\mathbf{y} \longmapsto \mathbf{H}_1^{-1}\mathbf{y}$ via CPA (the Chebyshev polynomial approximation algorithm in [13]), the JPA$(1/2, 1/2)$ (Jacobi polynomial approximation with appropriate selection of parameters $\alpha = 1/2$ and $\beta = 1/2$ in [37]), CIPA with zero initial $\mathbf{x}^{(0)} = \mathbf{0}$, the gradient descent method with optimal step size in [29] and autoregressive moving average method in [16], OGDA and ARMA for abbreviation, where $\mathbf{x}^{(m)}, m \geq 1$, are the output of the polynomial approximation algorithm at m-th iteration and M is the degree of polynomials the polynomial approximation. We observe that CIPA have the **best** performances on the implementation of inverse filtering procedure than the JPA$(1/2, 1/2)$ in [37], CPA in [13] does, and CIPA has much better performance than than the gradient descent method does.

B. Denoising dancing dog dataset

In the second simulation, we consider applying polynomial approximation algorithms to denoise the walking dog dataset $\mathbf{W}$ of size $442854 = 2502 \times 59 \times 3$ [14]. Let $\mathcal{T}$ be the line graph with 59 vertices and $\mathcal{W} = (V, E)$ be the undirected graph with 2502 vertices and edges constructed by the 5 nearest neighboring algorithm. The walking dog data is modelled

TABLE II: Average relative iteration errors $E(m)$ to implement the inverse filtering $\mathbf{y} \longmapsto \mathbf{H}_1^{-1}\mathbf{y}$ on the circulant graph $\mathcal{C}(1000, Q_0)$ via polynomial approximation algorithms with polynomial degree $M = 1$, the gradient descent algorithm with optimal step size and ARMA, where we take zero as the initial.

Alg. \ Iter. m	1	2	3	4	5
CPA	0.4494	0.2191	0.1103	0.0566	0.0295
JPA(1/2, 1/2)	0.2056	0.0769	0.0390	0.0213	0.0119
CIPA	0.2994	0.1010	0.0349	0.0122	0.0043
OGDA	0.2350	0.0856	0.0349	0.0147	0.0063
ARMA	0.3259	0.2583	0.1423	0.1098	0.0718

as a time-varying signal $\mathbf{W}(t,i) \in \mathbb{R}^3, t \in \{1,\ldots,59\}, i \in V$ on the product graph $\mathcal{T} \times \mathcal{W}$. Consider the scenario that the known dataset is the noisy walking dog dataset

$$\widetilde{\mathbf{W}} = \mathbf{W} + \lambda \boldsymbol{\eta} \tag{IV.1}$$

corrupted by some random noises $\lambda\boldsymbol{\eta}$, where $\boldsymbol{\eta}$ has its components $\boldsymbol{\eta}(t,i), t \in \{1,\ldots,59\}, i \in V$ being independently and randomly selected with a normal Gaussian distribution, and the normalization factor $\lambda = 0.2\|\mathbf{W}\|_F(\mathbb{E}\|\boldsymbol{\eta}\|_F^2)^{-1/2} = 29.1398$ is so chosen that the Frobenius norm $\lambda\|\boldsymbol{\eta}\|_F$ of the additive noise is about 20% of the norm $\|\mathbf{W}\|_F$ of the walking dog dataset.

A conventional denoising procedure is the Tikhonov denoising approach in each coordinate dimension ([13], [14]). Then the denoised walking dog dataset is given by

$$\widehat{\mathbf{w}} := \arg\min_{\mathbf{z}} \|\mathbf{z} - \widetilde{\mathbf{w}}\|_2^2 + \gamma_1 \mathbf{z}^T\mathbf{S}_1\mathbf{z} + \gamma_2\mathbf{z}^T\mathbf{S}_2\mathbf{z}, \tag{IV.2}$$

where $\widetilde{\mathbf{w}}$ is the vectorization of the noisy dog dataset $\widetilde{\mathbf{W}}$, $\mathbf{S}_1 = \mathbf{I} \otimes \mathbf{L}_{\mathcal{W}}^{\mathrm{sym}}$, $\mathbf{S}_2 = \mathbf{L}_{\mathcal{T}}^{\mathrm{sym}} \otimes \mathbf{I}$, $\mathbf{L}_{\mathcal{W}}^{\mathrm{sym}}$ and $\mathbf{L}_{\mathcal{T}}^{\mathrm{sym}}$ are symmetrically normalized Laplacian matrices on the graph $\mathcal{W}$ and $\mathcal{T}$ respectively, and penalty constants $\gamma_1, \gamma_2 \geq 0$ are used to balance the fidelity and regularization in the vertex and temporal domains. Shown in the top row of Figure 1 are the snapshots of the original walking dog dataset, the noisy walking dog dataset and the denoised walking dog dataset at time $t = 1$, where the penalty constants $\gamma_1 = \gamma_2 = 1$ are used. We observe that the proposed iterative polynomial approximation algorithms can effectively denoise the walking dog dataset. From the top right and top middle plots of Figure 1, we see that the denoised dog dataset reveals the shape of the dog, while the noisy walking dog dataset obscures the gesture of the two front legs.

Denote the output of the m-th iteration of the proposed algorithms by $\widehat{\mathbf{w}}^{(m)}$ and define the output signal-to-noise ratio of the proposed algorithms at m-th iteration by

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

Comparing with CIPA, OGDA in [29] has a slower convergence rate, see the bottom left plot of Figure 1. Our experiments also indicate that the OGDA may achieve a similar denoising performance after 20 iterations to that CIPA do after 3 iterations. Plotted in the bottom middle of Figure 1 are the average output signal-to-noise ratios $\max(\mathrm{SNR}(3), -5)$ of the

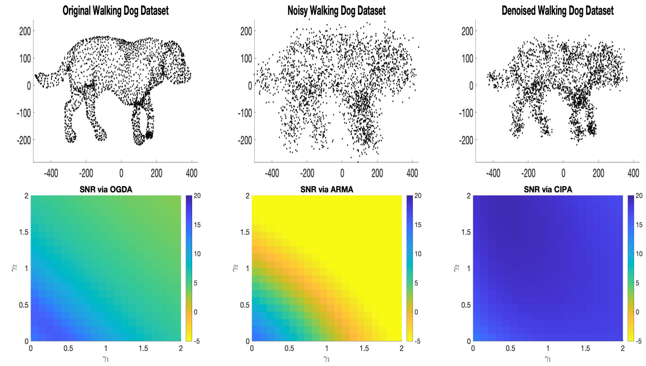

Fig. 1: Plotted on the top left, top middle and top right are the snapshots of the original walking dog dataset, the noisy dataset and the denoised dataset at the time $t = 1$, where the input SNR is 13.9734 and the output SNR is 18.9654. Plotted on the bottom from the left to the right are average output signal-to-noise ratios $\max(\mathrm{SNR}(m), -5)$ at m-th iteration of OGDA, ARMA and CIPA on denoising the walking dog dataset through Tikhonov denoising approach in (IV.2) with respect to different penalty constants $\gamma_1, \gamma_2 \in [0, 2]$, where $m = 3$.

ARMA model proposed in [16] with respect to the penalty constants $\gamma_1, \gamma_2 \in [0, 2]$ over 20 trials on the random noise $\boldsymbol{\eta}$ in (IV.1). It shows that the ARMA model may fail to denoise the walking dog dataset for penalty constants γ_1, γ_2 not close to zero. Recall that the requirement for convergence of the ARMA model in [16] is that the spectrum of $\mathbf{T} := \gamma_1\mathbf{S}_1 + \gamma_2\mathbf{S}_2$ is contained in $(-1, 1)$. Observe that the spectrum of $\mathbf{T}$ is contained in $[0, 2(\gamma_1 + \gamma_2)]$ from the spectral properties of $\mathbf{S}_1$ and $\mathbf{S}_2$. Then a possible explanation for why the ARMA model did not perform well is that it does not meet the requirement for the convergence of the ARMA method when γ_1, γ_2 are not close to zero.

Acknowledgement: This work is partially supported by National Key RD Program of China (No. 2024YFA1013703), National Nature Science Foundation of China (12171490), Guangdong Basic and Applied Basic Research Foundation (2022A1515011060), and Fundamental Research Funds for the Central Universities, Sun Yat-sen University (24lgqb019).

References

[1] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18-42, 2017.

[2] S. Chen, A. Sandryhaila, and J. Kovačević, "Distributed algorithm for graph signal inpainting," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, QLD, Apr. 2015, pp. 3731-3735.

[3] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević, "Signal recovery on graphs: variation minimization," *IEEE Trans. Signal Process.*, vol. 63, no. 17, pp. 4609-4624, Sept. 2015.

[4] W. Cheney and W. Light, *A Course in Approximation Theory*, American Mathematical Society, 2009.

[5] C. Cheng, N. Emirov, and Q. Sun, "Preconditioned gradient descent algorithm for inverse filtering on spatially distributed networks," *IEEE Signal Process. Lett.*, vol. 27, pp. 1834-1838, Oct. 2020.

[6] C. Cheng, J. Jiang, N. Emirov, and Q. Sun, "Iterative Chebyshev polynomial algorithm for signal denoising on graphs," in *Proceeding 13th Int. Conf. on SampTA*, Bordeaux, France, Jul. 2019, pp. 1-5.

[7] C. Cheng, Y. Jiang, and Q. Sun, "Spatially distributed sampling and reconstruction," *Appl. Comput. Harmon. Anal.*, vol. 47, no. 1, pp. 109-148, July 2019.

[8] C. Chong and S. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proc. IEEE*, vol. 91, pp. 1247-1256, Aug. 2003.

[9] M. Coutino, E. Isufi, and G. Leus, "Advances in distributed graph filtering," *IEEE Trans. Signal Process.*, vol. 67, no. 9, pp. 2320-2333, May 2019.

[10] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard, "Graph signal processing for machine learning: A review and new perspectives," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 117-127, 2020.

[11] V. N. Ekambaram, G. C. Fanti, B. Ayazifar, and K. Ramchandran, "Circulant structures and graph signal processing," in *Proc. IEEE Int. Conf. Image Process.*, 2013, pp. 834-838.

[12] V. N. Ekambaram, G. C. Fanti, B. Ayazifar, and K. Ramchandran, "Multiresolution graph signal processing via circulant structures," in *Proc. IEEE Digital Signal Process./Signal Process. Educ. Meeting (DSP/SPE)*, 2013, pp. 112-117.

[13] N. Emirov, C. Cheng, J. Jiang, and Q. Sun, "Polynomial graph filter of multiple shifts and distributed implementation of inverse filtering," *Sampl. Theory Signal Process. Data Anal.*, vol. 20, Article No. 2, 2022.

[14] F. Grassi, A. Loukas, N. Perraudin and B. Ricaud, "A time-vertex signal processing framework: scalable processing and meaningful representations for time-series on graphs," *IEEE Trans. Signal Process.*, vol. 66, no. 3, pp. 817-829, Feb. 2018.

[15] E. Isufi, A. Loukas, N. Perraudin, and G. Leus, "Forecasting time series with VARMA recursions on graphs," *IEEE Trans. Signal Process.*, vol. 67, no. 18, pp. 4870-4885, Sept. 2019.

[16] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Autoregressive moving average graph filtering," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 274-288, Jan. 2017.

[17] J. Jiang, C. Cheng, and Q. Sun, "Nonsubsampled graph filter banks: Theory and distributed algorithms," *IEEE Trans. Signal Process.*, vol. 67, no. 15, pp. 3938-3953, Aug. 2019.

[18] J. Jiang, D. B. Tay, Q. Sun, and S. Ouyang, "Design of nonsubsampled graph filter banks via lifting schemes," *IEEE Signal Process. Lett.*, vol. 27, pp. 441-445, Feb. 2020.

[19] M. S. Kotzagiannidis and P. L. Dragotti, "Sampling and reconstruction of sparse signals on circulant graphs – an introduction to graph-FRI," *Appl. Comput. Harmon. Anal.*, vol. 47, no. 3, pp. 539-565, Nov. 2019.

[20] M. S. Kotzagiannidis and P. L. Dragotti, "Splines and wavelets on circulant graphs," *Appl. Comput. Harmon. Anal.*, vol. 47, no. 2, pp. 481-515, Sept. 2019.

[21] K. Lu, A. Ortega, D. Mukherjee, and Y. Chen, "Efficient rate-distortion approximation and transform type selection using Laplacian operators," in *2018 Picture Coding Symposium (PCS)*, San Francisco, CA, June 2018, pp. 76-80.

[22] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Comput. Netw.*, vol. 51, no. 10, pp. 2529-2553, July 2007.

[23] N. Motee and Q. Sun, "Sparsity and spatial localization measures for spatially distributed systems," *SIAM J. Control Optim.*, vol. 55, no. 1, pp. 200-235, Jan. 2017.

[24] M. Onuki, S. Ono, M. Yamagishi, and Y. Tanaka, "Graph signal denoising via trilateral filter on graph spectral domain," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 137-148, June 2016.

[25] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808-828, May 2018.

[26] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042-3054, June 2014.

[27] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644-1656, Apr. 2013.

[28] S. Segarra, A. G. Marques, and A. Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 4117-4131, Aug. 2017.

[29] X. Shi, H. Feng, M. Zhai, T. Yang, and B. Hu, "Infinite impulse response graph filters in wireless sensor networks," *IEEE Signal Process. Lett.*, vol. 22, no. 8, pp. 1113-1117, Aug. 2015.

[30] 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, May 2013.

[31] D. I. Shuman, P. Vandergheynst, D. Kressner, and P. Frossard, "Distributed signal processing via Chebyshev polynomial approximation," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 4, pp. 736-751, Dec. 2018.

[32] L. Stanković, M. Daković, and E. Sejdić, "Introduction to graph signal processing," In *Vertex-Frequency Analysis of Graph Signals*, Springer, pp. 3-108, 2019.

[33] L. N. Trefethen, *Approximation Theory and Approximation Practice*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2013.

[34] W. Waheed and D. B. H. Tay, "Graph polynomial filter for signal denoising," *IET Signal Process.*, vol. 12, no. 3, pp. 301-309, Apr. 2018.

[35] S. Wang, "On Error bounds for orthogonal polynomial expansions and Gauss-type quadrature," *SIAM J. Numer. Anal.*, vol. 50, no. 3, pp. 1240-1263, 2012.

[36] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292-2330, Aug. 2008.

[37] C. Zheng, C. Cheng, and Q. Sun, "Wiener filters on graphs and distributed implementations", *Digit. Signal Process.*, vol. 162, Article No. 105156, 2025.