Preconditioned Gradient Descent Algorithm for Inverse Filtering on Spatially Distributed Networks

Cheng Cheng, Nazar Emirov, and Qiyu Sun, *Member, IEEE*

***Abstract*—Graph filters and their inverses have been widely used in denoising, smoothing, sampling, interpolating and learning. Implementation of an inverse filtering procedure on spatially distributed networks (SDNs) is a remarkable challenge, as each agent on an SDN is equipped with a data processing subsystem with limited capacity and a communication subsystem with confined range due to engineering limitations. In this letter, we introduce a preconditioned gradient descent algorithm to implement the inverse filtering procedure associated with a graph filter having small geodesic-width. The proposed algorithm converges exponentially, and it can be implemented at vertex level and applied to time-varying inverse filtering on SDNs.**

***Index Terms*—Graph signal processing, inverse filtering, spatially distributed network, gradient descent method, preconditioning, quasi-Newton method.**

I. Introduction

Spatially distributed networks (SDNs) have been widely used in (wireless) sensor networks, drone fleets, smart grids and many real world applications [1]–[4]. An SDN has a large amount of agents and each agent equipped with a data processing subsystem having limited data storage and computation power and a communication subsystem for data exchanging to its "neighboring" agents within communication range. The topology of an SDN can be described by a connected, undirected and unweighted finite graph $\mathcal{G} := (V, E)$ with a vertex in V representing an agent and an edge in E between vertices indicating that the corresponding agents are within some range in the spatial space. In this letter, we consider SDNs equipped with a communication subsystem at each agent to directly communicate between two agents if the geodesic distance between their corresponding vertices $i, j \in V$ is at most L, i.e., $\rho(i, j) \le L$, where $\rho(i, j)$ is the number of edges in a shortest path connecting $i, j \in V$, and we call the minimal integer $L \ge 1$ as the *communication range* of the SDN. Therefore the implementation of data processing on our SDNs is a distributed task and it should be designed at agent/vertex level with confined communication range. In this letter, we consider the implementation of graph filtering and inverse filtering on SDNs, which are required to be fulfilled at agent level with communication range no more than L.

Manuscript received July 22, 2020; revised September 11, 2020; accepted September 20, 2020. Date of publication October 8, 2020; date of current version October 28, 2020. This work was supported by the National Science Foundation under Grant DMS-1816313. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ran Tao. *(Corresponding author: Qiyu Sun.)*

Cheng Cheng is with the School of Mathematics, Sun Yat-sen University, Guangzhou 510275, China (e-mail: cheng_cheng87@126.com).

Nazar Emirov and Qiyu Sun are with the Department of Mathematics, University of Central Florida, Orlando, FL 32816 USA (e-mail: nazaremirov@knights.ucf.edu; qiyu.sun@ucf.edu).

Digital Object Identifier 10.1109/LSP.2020.3029699

A signal on a graph $\mathcal{G} = (V, E)$ is a vector $\mathbf{x} = (x(i))_{i\in V}$ indexed by the vertex set, and a graph filter $\mathbf{H}$ maps a graph signal $\mathbf{x}$ linearly to another graph signal $\mathbf{y} = \mathbf{H}\mathbf{x}$, which is usually represented by a matrix $\mathbf{H} = (H(i, j))_{i,j\in V}$ indexed by vertices in V. Graph filtering $\mathbf{x} \mapsto \mathbf{H}\mathbf{x}$ and its inverse filtering $\mathbf{y} \mapsto \mathbf{H}^{-1}\mathbf{y}$ play important roles in graph signal processing and they have been used in smoothing, sampling, interpolating and many real-world applications [2], [5]–[9]. A graph filter $\mathbf{H} = (H(i, j))_{i,j\in V}$ is said to have *geodesic-width* $\omega(\mathbf{H})$ if $H(i, j) = 0$ for all $i, j \in V$ with $\rho(i, j) > \omega(\mathbf{H})$ [4], [10], [11]. For a filter $\mathbf{H} = (H(i, j))_{i,j\in V}$ with geodesic-width $\omega(\mathbf{H})$, the corresponding filtering process

$$(x(i))_{i\in V} =: \mathbf{x} \longmapsto \mathbf{H}\mathbf{x} = \mathbf{y} := (y(i))_{i\in V} \tag{1}$$

can be implemented at vertex level, and the output at a vertex $i \in V$ is a "weighted" sum of the input in its $\omega(\mathbf{H})$-neighborhood,

$$y(i) = \sum_{\rho(j,i)\le\omega(\mathbf{H})} H(i, j)x(j). \tag{2}$$

For SDNs with communication range $L \ge \omega(\mathbf{H})$, the above implementation at vertex level provides an essential tool for the filtering procedure (1), in which each agent $i \in V$ has equipped with subsystems to store $H(i, j)$ and $x(j)$ with $\rho(j, i) \le \omega(\mathbf{H})$, to compute addition and multiplication in (2), and to exchange data to its neighboring agents $j \in V$ satisfying $\rho(j, i) \le \omega(\mathbf{H})$.

For an invertible filter $\mathbf{H}$, the implementation of the inverse filtering procedure

$$\mathbf{y} \longmapsto \mathbf{H}^{-1}\mathbf{y} =: \mathbf{x} \tag{3}$$

cannot be directly applied for our SDNs, since the inverse filter $\mathbf{H}^{-1}$ may have geodesic-width *larger* than the communication range L. For the consideration of implementing inverse filtering on our SDN, we construct a diagonal preconditioning matrix $\mathbf{P}_\mathbf{H}$ in (5) at vertex level, and propose the preconditioned gradient descent algorithm (PGDA) (11) to implement inverse filtering on the SDN, see Algorithms II.1 and II.2.

A conventional approach to implement the inverse filtering procedure (3) is via the iterative quasi-Newton method

$$\mathbf{e}^{(m)} = \mathbf{H}\mathbf{x}^{(m-1)} - \mathbf{y} \text{ and } \mathbf{x}^{(m)} = \mathbf{x}^{(m-1)} - \mathbf{G}\mathbf{e}^{(m)},\ m \ge 1, \tag{4}$$

with arbitrary initial $\mathbf{x}^{(0)}$, where the filter $\mathbf{G}$ is an approximation to the inverse $\mathbf{H}^{-1}$. A challenge in the quasi-Newton method is how to select the approximation filter $\mathbf{G}$ appropriately. For the widely used polynomial graph filters $\mathbf{H} = h(\mathbf{S}) = \sum_{k=0}^{K} h_k\mathbf{S}^k$ of a graph shift $\mathbf{S}$ where $h(t) = \sum_{k=0}^{K} h_k t^k$ [11]–[19], several methods have been proposed to construct polynomial approximation filters $\mathbf{G}$ [13], [14], [16], [17], [19]. However, for the convergence of the corresponding quasi-Newton method, some prior knowledge is required for the polynomial h and the graph shift $\mathbf{S}$, such as the whole spectrum of the shift $\mathbf{S}$ in the optimal polynomial approximation method [19], the interval containing the spectrum of the shift $\mathbf{S}$ in the Chebyshev approximation method [16], [17], [19], and the spectral radius of the shift $\mathbf{S}$ and the zero set of the polynomial h in the autoregressive moving average filtering algorithm [13], [14]. For a non-polynomial graph filter $\mathbf{H}$, the approximation filter in the gradient descent method is of the form $\mathbf{G} = \beta\mathbf{H}^T$ with selection of the optimal step length β depending on maximal and minimal singular values of the filter $\mathbf{H}$ [12], [20], and the approximation filter in the iterative matrix inverse approximation algorithm (IMIA) could be selected under a strong assumption on $\mathbf{H}$ [10, Theorem 3.2]. The proposed PGDA (11) is the quasi-Newton method (4) with $\mathbf{P}_{\mathbf{H}}^{-2}\mathbf{H}^T$ being selected as the approximation filter $\mathbf{G}$, see (7). Comparing with the quasi-Newton methods in [10], [12]–[14], [16], [17], [19], [20], one significance of the proposed PGDA is that the sequence $\mathbf{x}^{(m)}, m \ge 0$, in (11) converges exponentially to the output $\mathbf{x}$ of the inverse filtering procedure (3) whenever the filter $\mathbf{H}$ is invertible, see Theorems II.3 and III.1.

Data processing of time-varying signals, such as data collected by an SDN of sensors over a period of time, has been received a lot of attentions recently [6], [7], [9], [19], [22]–[24]. For a *time-varying* filter $\mathbf{H}_t = (H_t(i,j))_{i,j\in V}, t \ge 0$, with geodesic width $\omega(\mathbf{H}_t) \le L$ bounded by the communication range L of the SDN, the quasi-Newton method (4) to implement the inverse filtering procedure $\mathbf{y}_t \longmapsto \mathbf{H}_t^{-1}\mathbf{y}_t,\ t \ge 0$, on the SDN should be designed to be *self-adaptive*, since each agent $i \in V$ of the SDN does not have the whole updated filter $\mathbf{H}_t$ and it only receives the entries $H_t(i,j)$ and $H_t(j,i), \rho(j,i) \le L$, on the i-th row and column of $\mathbf{H}_t$ within the range L at every time instant t [4]. Clearly, the quasi-Newton method (4) is self-adaptive if the approximation filters $\mathbf{G}_t = (G_t(i,j))_{i,j\in V}, t \ge 0$ are locally selected without the involvement of any global information of the time-varying filter $\mathbf{H}_t$. The IMIA algorithm is self-adaptive [10, Eq. (3.4)] but the gradient descent method [12], [20] is not self-adaptive in general except that the step length β can be chosen to be *time-independent*. The second significance of the proposed PGDA is its *self-adaptivity* and *compatibility* to implement the time-varying inverse filtering procedure on our SDNs, as hence the approximation filter $\mathbf{P}_{\mathbf{H}}^{-2}\mathbf{H}^T$ in the PGDA is constructed at the vertex level with confined communication range, see Algorithm II.1.

II. Preconditioned Gradient Descent Algorithm for Inverse Filtering

Let $\mathcal{G} := (V, E)$ be a connected, undirected and unweighted graph and $\mathbf{H} = (H(i,j))_{i,j\in V}$ be a filter on the graph $\mathcal{G}$ with geodesic-width $\omega(\mathbf{H})$. Denote the set of all s-hop neighbors of a vertex $i \in V$ by $B(i,s) = \{j \in V,\ \rho(j,i) \le s\},\ s \ge 0$. In this section, we induce a diagonal matrix $\mathbf{P}_{\mathbf{H}}$ with diagonal elements $P_{\mathbf{H}}(i,i), i \in V$, given by

$$P_{\mathbf{H}}(i,i) := \max_{k\in B(i,\omega(\mathbf{H}))}\left\{\max\left(\sum_{j\in B(k,\omega(\mathbf{H}))}|H(j,k)|,\ \sum_{j\in B(k,\omega(\mathbf{H}))}|H(k,j)|\right)\right\}. \quad (5)$$

Algorithm II.1: Realization of the Preconditioner $\mathbf{P}_{\mathbf{H}}$ at a vertex $i \in V$.

Inputs: Geodesic width $\omega(\mathbf{H})$ of the filter $\mathbf{H}$ and nonzero entries $H(i,j)$ and $H(j,i)$ for $j \in B(i,\omega(\mathbf{H}))$ in the i-th row and column of the filter $\mathbf{H}$.

1) Calculate $d(i) = \max\{\sum_{j\in B(i,\omega(\mathbf{H}))}|H(i,j)|, \sum_{j\in B(i,\omega(\mathbf{H}))}|H(j,i)|\}$.

2) Send $d(i)$ to all neighbors $k \in B(i,\omega(\mathbf{H}))\backslash\{i\}$ and receive $d(k)$ from neighbors $k \in B(i,\omega(\mathbf{H}))\backslash\{i\}$.

3) Calculate $P_{\mathbf{H}}(i,i) = \max_{k\in B(i,\omega(\mathbf{H}))} d(k)$.

Output: $P_{\mathbf{H}}(i,i)$.

The above diagonal matrix $\mathbf{P}_{\mathbf{H}}$ can be evaluated at vertex level and constructed on SDNs with communication range $L \ge \omega(\mathbf{H})$, see Algorithm II.1.

For symmetric matrices $\mathbf{A}$ and $\mathbf{B}$, we use $\mathbf{B} \preceq \mathbf{A}$ and $\mathbf{B} \prec \mathbf{A}$ to denote the positive semidefiniteness and positive definiteness of their difference $\mathbf{A} - \mathbf{B}$ respectively. A crucial observation about the diagonal matrix $\mathbf{P}_{\mathbf{H}}$ is as follows.

Theorem II.1: Let $\mathbf{H}$ be a graph filter with geodesic-width $\omega(\mathbf{H})$ and $\mathbf{P}_{\mathbf{H}}$ be as in (5). Then

$$\mathbf{H}^T\mathbf{H} \preceq \mathbf{P}_{\mathbf{H}}^2. \quad (6)$$

Proof: Write $\mathbf{H} = (H(i,j))_{i,j\in V}$. For $\mathbf{x} = (x(i))_{i\in V}$,

$$\begin{aligned}
0 \le \mathbf{x}^T\mathbf{H}^T\mathbf{H}\mathbf{x} &= \sum_{j\in V}\Big|\sum_{i\in V}H(j,i)x(i)\Big|^2 \\
&\le \sum_{j\in V}\left(\sum_{i\in V}|H(j,i)||x(i)|^2\right)\times\left(\sum_{i'\in V}|H(j,i')|\right) \\
&= \sum_{i\in V}|x(i)|^2\sum_{j\in B(i,\omega(\mathbf{H}))}|H(j,i)|\times\left(\sum_{i'\in V}|H(j,i')|\right) \\
&\le \sum_{i\in V}|x(i)|^2P_{\mathbf{H}}(i,i)\sum_{j\in B(i,\omega(\mathbf{H}))}|H(j,i)| \\
&\le \sum_{i\in V}(P_{\mathbf{H}}(i,i))^2|x(i)|^2 = \mathbf{x}^T\mathbf{P}_{\mathbf{H}}^2\mathbf{x}.
\end{aligned}$$

This proves (6) and completes the proof. ■

Denote the spectral radius and operator norm of a matrix $\mathbf{A}$ by $r(\mathbf{A})$ and $\|\mathbf{A}\|_2 = \sup_{\|\mathbf{x}\|_2=1}\|\mathbf{A}\mathbf{x}\|_2$ respectively, where $\|\mathbf{x}\|_2 = (\sum_{j\in V}|x(j)|^2)^{1/2}$ for $\mathbf{x} = (x_j)_{j\in V}$. By Theorem II.1, $\mathbf{P}_{\mathbf{H}}^{-2}\mathbf{H}^T$ is an approximation filter to the inverse filter $\mathbf{H}^{-1}$ in the sense that

$$r(\mathbf{I} - \mathbf{P}_{\mathbf{H}}^{-2}\mathbf{H}^T\mathbf{H}) = \|\mathbf{I} - \mathbf{P}_{\mathbf{H}}^{-1}\mathbf{H}^T\mathbf{H}\mathbf{P}_{\mathbf{H}}^{-1}\|_2 < 1. \quad (7)$$

Remark II.2: Define the Schur norm of a matrix $\mathbf{H} = (H(i,j))_{i,j\in V}$ by

$$\|\mathbf{H}\|_{\mathcal{S}} = \max\left\{\max_{i\in V}\sum_{j\in V}|H(i,j)|,\ \max_{j\in V}\sum_{i\in V}|H(i,j)|\right\},$$

Algorithm II.2: Implementation of the PGDA (11) at a vertex $i \in V$.

Inputs: Iteration number M, geodesic-width $\omega(\mathbf{H})$, preconditioning constant $P_{\mathbf{H}}(i,i)$, observation $y(i)$ at vertex i, and filter coefficients $H(i,j)$ and $H(j,i),\ j \in B(i,\omega(\mathbf{H}))$.

1) Calculate $\widetilde{H}(j,i) = H(j,i)/(P_{\mathbf{H}}(i,i))^2$.

Initialization: Initial $x^{(0)}(j), j \in B(i,\omega(\mathbf{H}))$, and $m=1$.

2) Calculate $v^{(m)}(i) = y(i) - \sum_{j\in B(i,\omega(\mathbf{H}))} H(i,j)x^{(m-1)}(j)$.

3) Send $v^{(m)}(i)$ to neighbors $j \in B(i,\omega(\mathbf{H}))$ and receive $v^{(m)}(j)$ from neighbors $j \in B(i,\omega(\mathbf{H}))$.

4) Update $x^{(m)}(i) = x^{(m-1)}(i) + \sum_{j\in B(i,\omega(\mathbf{H}))} \widetilde{H}(j,i)v^{(m)}(j)$.

5) Send $x^{(m)}(i)$ to neighbors $j \in B(i,\omega(\mathbf{H}))$ and receive $x^{(m)}(j)$ from neighbors $j \in B(i,\omega(\mathbf{H}))$.

6) Set $m = m+1$ and return to Step **2)** if $m \le M$.

Outputs: $x(j) := x^{(M)}(j),\ j \in B(i,\omega(\mathbf{H}))$.

and denote the zero and identity matrices of appropriate size by $\mathbf{O}$ and $\mathbf{I}$ respectively. One may verify that

$$\mathbf{O} \prec \mathbf{H}^T\mathbf{H} \preceq \|\mathbf{H}\|_{\mathcal{S}}^2\mathbf{I}. \tag{8}$$

By (5), we have $\mathbf{P_H} \preceq \|\mathbf{H}\|_{\mathcal{S}}\mathbf{I}$. Then we may consider the conclusion (6) for the preconditioner $\mathbf{P_H}$ as a *distributed* version of the well-known matrix dominance (8) for the graph filter $\mathbf{H}$.

Preconditioning technique has been widely used in numerical analysis to solve a linear system, where the difficulty is how to select the preconditioner appropriately. In this letter, we use $\mathbf{P_H}$ as a right preconditioner to the linear system $\mathbf{Hx} = \mathbf{y}$ associated with the inverse filtering procedure (3), and we solve the following right preconditioned linear system

$$\mathbf{HP_H^{-1}z} = \mathbf{y} \ \text{ and } \ \mathbf{x} = \mathbf{P_H^{-1}z}, \tag{9}$$

via the gradient descent algorithm

$$\begin{cases} \mathbf{z^{(m)}} = \mathbf{z^{(m-1)}} - \mathbf{P_H^{-1}H^T}\left(\mathbf{HP_H^{-1}z^{(m-1)}} - \mathbf{y}\right) \\ \mathbf{x}^{(m)} = \mathbf{P_H^{-1}z}^{(m)},\ \ m \ge 1, \end{cases}$$

with initial $\mathbf{z}^{(0)}$. The above iterative algorithm can be reformulated as a quasi-Newton method (4) with $\mathbf{G}$ replaced by $\mathbf{P_H^{-2}H}^T$,

$$\begin{cases} \mathbf{e^{(m)}} = \mathbf{Hx^{(m-1)}} - \mathbf{y} \\ \mathbf{x}^{(m)} = \mathbf{x}^{(m-1)} - \mathbf{P_H^{-2}H}^T\mathbf{e}^{(m)},\ \ m \ge 1 \end{cases} \tag{10}$$

with initial $\mathbf{x}^{(0)}$. We call the above approach to implement the inverse filtering procedure (3) by the *preconditioned gradient descent algorithm*, or PGDA for abbreviation.

Define $\mathbf{w}_m := \mathbf{P_H}(\mathbf{x}^{(m)} - \mathbf{H}^{-1}\mathbf{y}),\ m \ge 0$. Then

$$\mathbf{w}_m = \left(\mathbf{I} - \mathbf{P_H^{-1}H}^T\mathbf{HP_H^{-1}}\right)\mathbf{w}_{m-1},\ m \ge 1 \tag{11}$$

by (10). Therefore the iterative algorithm (10) converges exponentially by (7) and (11).

Theorem II.3: Let $\mathbf{H}$ be an invertible graph filter and $\mathbf{x}^{(m)}, m \ge 0$, be as in (11). Then

$$\|\mathbf{P_H}(\mathbf{x}^{(m)} - \mathbf{H}^{-1}\mathbf{y})\|_2 \le \left\|\mathbf{I} - \mathbf{P_H^{-1}H}^T\mathbf{HP_H^{-1}}\right\|_2^m \times \|\mathbf{P_H}(\mathbf{x}^{(0)} - \mathbf{H}^{-1}\mathbf{y})\|_2,\ m \ge 0.$$

Algorithm III.1: Implementation of the SPGDA (15) at a vertex $i \in V$.

Inputs: Iteration number M, geodesic-width $\omega(\mathbf{H})$, observation $y(i)$ at vertex i, and filter coefficients $H(i,j)$ and $H(j,i), j \in B(i,\omega(\mathbf{H}))$.

1) Calculate $P_{\mathbf{H}}^{\rm sym}(i,i) = \sum_{j\in B(i,\omega(\mathbf{H}))} |H(i,j)|$, $\widetilde{H}(i,j) = H(i,j)/P_{\mathbf{H}}^{\rm sym}(i,i)$ and $\tilde{y}(i) = y(i)/P_{\mathbf{H}}^{\rm sym}(i,i), j \in B(i,\omega(\mathbf{H}))$.

Initialization: Initial $x^{(0)}(j), j \in B(i,\omega(\mathbf{H}))$ and $m=1$.

2) Compute $x^{(m)}(i) = x^{(m-1)}(i) + \tilde{y}(i) - \sum_{j\in B(i,\omega(\mathbf{H}))} \widetilde{H}(i,j)x^{(m-1)}(j)$.

3) Send $x^{(m)}(i)$ to neighbors $j \in B(i,\omega(\mathbf{H}))$ and receive $x^{(m)}(j)$ from neighbors $j \in B(i,\omega(\mathbf{H}))$.

4) Set $m = m+1$ and return to Step **2)** if $m \le M$.

Outputs: $x(j) := x^{(M)}(j), j \in B(i,\omega(\mathbf{H}))$.

In addition to the exponential convergence in Theorem II.3, each iteration in the PGDA can be implemented at vertex level, see Algorithm II.2. Therefore for an invertible filter $\mathbf{H}$ with $\omega(\mathbf{H}) \le L$, the PGDA (11) can implement the inverse filtering procedure (3) on SDNs with each agent only storing, computing and exchanging the information in a L-hop neighborhood.

III. Symmetric Preconditioned Gradient Descent Algorithm for Inverse Filtering

In this section, we consider implementing the inverse filtering procedure (3) associated with a **positive definite** filter $\mathbf{H} = (H(i,j))_{i,j\in V}$ on a connected, undirected and unweighted graph $\mathcal{G}$. Define the diagonal matrix $\mathbf{P}_{\mathbf{H}}^{\rm sym}$ with diagonal entries

$$P_{\mathbf{H}}^{\rm sym}(i,i) = \sum_{j\in B(i,\omega(\mathbf{H}))} |H(i,j)|,\ i \in V, \tag{12}$$

and set

$$\widehat{\mathbf{H}} = (\mathbf{P}_{\mathbf{H}}^{\rm sym})^{-1/2}\mathbf{H}(\mathbf{P}_{\mathbf{H}}^{\rm sym})^{-1/2}. \tag{13}$$

We remark that the normalized matrix in (14) associated with a diffusion matrix has been used to understand diffusion process [25], and the one corresponding to the Laplacian $\mathbf{L}_{\mathcal{G}}$ on the graph $\mathcal{G}$ is half of its normalized Laplacian $\mathbf{L}_{\mathcal{G}}^{\rm sym} := (\mathbf{D}_{\mathcal{G}})^{-1/2}\mathbf{L}_{\mathcal{G}}(\mathbf{D}_{\mathcal{G}})^{-1/2}$, where $\mathbf{D}_{\mathcal{G}}$ is degree matrix of $\mathcal{G}$ [11]. Similar to the PGDA (11), we propose the following *symmetric preconditioned gradient descent algorithm*, or SPGDA for abbreviation,

$$\mathbf{x}^{(m)} = \mathbf{x}^{(m-1)} - (\mathbf{P}_{\mathbf{H}}^{\rm sym})^{-1}(\mathbf{Hx}^{(m-1)} - \mathbf{y}),\ m \ge 1, \tag{14}$$

with initial $\mathbf{x}^{(0)}$, to solve the following preconditioned linear system $\widehat{\mathbf{H}}\mathbf{z} = (\mathbf{P}_{\mathbf{H}}^{\rm sym})^{-1/2}\mathbf{y}$ and $\mathbf{x} = (\mathbf{P}_{\mathbf{H}}^{\rm sym})^{-1/2}\mathbf{z}$. Comparing with the PGDA (11), the SPGDA for a positive definite graph filter has less computation and communication cost in each iteration and it also can be implemented at vertex level, see Algorithm III.1.

For $\mathbf{x} = (x(i))_{i\in V}$, we obtain from (12) and the symmetry of the matrix $\mathbf{H}$ that

$$\mathbf{x}^T\mathbf{Hx} \le \sum_{i,j\in V} |H(i,j)| \frac{(x(i))^2 + (x(j))^2}{2} = \mathbf{x}^T\mathbf{P}_{\mathbf{H}}^{\rm sym}\mathbf{x}.$$

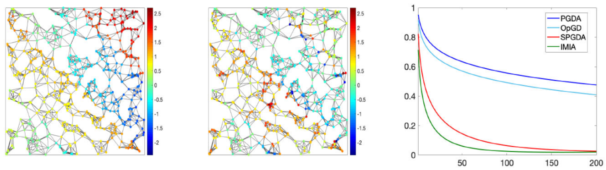

Fig. 1. Plotted on the left is a corrupted blockwise polynomial signal $\mathbf{x}$ and in the middle is the output $\mathbf{y}=\mathbf{H}\mathbf{x}$ of the filtering procedure, where $\|\mathbf{x}\|_2=24.8194$, $\|\mathbf{y}\|_2=21.5317$ and the condition number of the filter $\mathbf{H}$ is 107.40. Shown on the right is average of the relative inverse filtering error $E_2(m)=\|\mathbf{x}^{(m)}-\mathbf{x}\|_2/\|\mathbf{x}\|_2, 1\le m\le 200$ over 1000 trials, where $N=K=512, \eta=0.2, \gamma=0.05$ and $\mathbf{x}^{(m)}$, $m\ge 1$, are the outputs of SPGDA, PGDA, OpGD and IMIA.

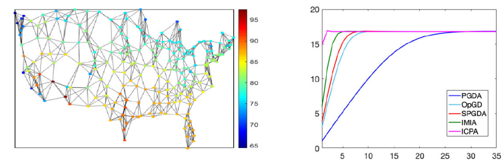

Fig. 2. Plotted on the left is the original temperature data $\mathbf{x}_{12}$. Shown on the right is average of the signal-to-noise ratio $\mathrm{SNR}(m)=-20\log_{10}\|\mathbf{x}^{(m)}-\mathbf{x}_{12}\|_2/\|\mathbf{x}_{12}\|_2, 1\le m\le 35$, over 1000 trials, where $\mathbf{x}^{(m)}$, $m\ge 1$, are the outputs of PGDA, SPGDA, OpGD, IMIA and ICPA, and average of the limit SNR is 16.7869.

Combining (5) and (12) proves that $\mathbf{H}\preceq \mathbf{P}_{\mathbf{H}}^{\mathrm{sym}}\preceq \mathbf{P}_{\mathbf{H}}$, cf. (6). This together with (13) implies that $r(\mathbf{I}-(\mathbf{P}_{\mathbf{H}}^{\mathrm{sym}})^{-1}\mathbf{H})=r(\mathbf{I}-\widehat{\mathbf{H}})=\|\mathbf{I}-\widehat{\mathbf{H}}\|_2<1$. Similar to the proof of Theorem II.3, we have

Theorem III.1: Let $\mathbf{H}$ be a positive definite graph filter. Then $\mathbf{x}^{(m)}, m\ge 0$, in (14) converges exponentially,

$$\|(\mathbf{P}_{\mathbf{H}}^{\mathrm{sym}})^{1/2}(\mathbf{x}^{(m)}-\mathbf{H}^{-1}\mathbf{y})\|_2 \le \|\mathbf{I}-\widehat{\mathbf{H}}\|_2^m \big\|(\mathbf{P}_{\mathbf{H}}^{\mathrm{sym}})^{1/2}(\mathbf{x}^{(0)}-\mathbf{H}^{-1}\mathbf{y})\big\|_2.$$

IV. Numerical Simulations

Let $\mathcal{G}_N=(V_N,E_N), N\ge 2$, be random geometric graphs with N vertices deployed on $[0,1]^2$ and an undirected edge between two vertices if their physical distance is not larger than $\sqrt{2/N}$ [11], [26]. In the first simulation, we consider the inverse filtering procedure associated with the graph filter $\mathbf{H}=\mathbf{H}_o+(\mathbf{L}_{\mathcal{G}_N}^{\mathrm{sym}})^2$, where $K\ge 1$, $\mathbf{L}_{\mathcal{G}_N}^{\mathrm{sym}}$ is the normalized Laplacian on the graph $\mathcal{G}_N$, the filter $\mathbf{H}_o=(H_o(i,j))_{i,j\in V_N}$ is defined by $H_o(i,j)=0$ if $\rho(i,j)\ge 3$ and

$$H_o(i,j)=\exp\bigg(-2K\|(i_x,i_y)-(j_x,j_y)\|_2^2 - \frac{\|(i_x,i_y)+(j_x,j_y)\|_2^2}{2}\bigg)+\frac{\gamma_{ij}+\gamma_{ji}}{2} \text{ if } \rho(i,j)\le 2, \quad (15)$$

(i_x,i_y) is the coordinator of a vertex $i\in V_N$ and γ_{ij} are i.i.d random noises uniformly distributed on $[-\gamma,\gamma]$. Let $\mathbf{x}_o$ be the blockwise polynomial consisting of four strips and imposes $(0.5-2i_x)$ on the first and third diagonal strips and $(0.5+i_x^2+i_y^2)$ on the second and fourth strips respectively [11], [19]. In the simulation, the signals $\mathbf{x}=\mathbf{x}_o+\boldsymbol{\eta}$ are obtained by a blockwise polynomial $\mathbf{x}_o$ corrupted by noises $\boldsymbol{\eta}$ with their components being i.i.d. random variables with uniform distribution on $[-\eta,\eta]$, and the observations $\mathbf{y}$ of the filtering procedure are given by $\mathbf{y}=\mathbf{H}\mathbf{x}$, see the left and middle images of Fig. 1.

In the simulation, we use the SPGDA (15) and the PGDA (10) with zero initial to implement the inverse filtering procedure $\mathbf{y}\mapsto\mathbf{H}^{-1}\mathbf{y}$, and also we compare their performances with the gradient decent algorithm; $\mathbf{x}^{(m)}=(\mathbf{I}-\beta_{op}\mathbf{H}^T\mathbf{H})\mathbf{x}^{(m-1)}+\beta_{op}\mathbf{H}^T\mathbf{y}$, $m\ge 1$, with zero initial and optimal step length β_{op} selected in [12], [19], [20],

OpGD in abbreviation, and the iterative matrix inverse approximation algorithm, $\mathbf{x}^{(m)}=(\mathbf{I}-\tilde{\mathbf{D}}\mathbf{H})\mathbf{x}^{(m-1)}+\tilde{\mathbf{D}}\mathbf{y}$, $m\ge 1$, IMIA in abbreviation, where $\mathbf{x}^{(0)}=\mathbf{0}$ and the diagonal matrix $\tilde{\mathbf{D}}$ has entries $H(i,i)/(\sum_{\rho(j,i)\le 2}|H(i,j)|^2), i\in V$, see [10, Eq. (3.4)] with $\tilde{\sigma}=0$. Shown in Fig. 1 is the average of the relative inverse filtering error $E_2(m), 1\le m\le 200$ over 1000 trials, and it reaches the relative error 5% at about 57th iteration for IMIA, 118th iteration for SPGDA, and more than 3000 iterations for PGDA and OpGD. This confirms that $\mathbf{x}^{(m)}, m\ge 1$, in the SPGDA, PGDA, OpGD and IMIA converge exponentially to the output $\mathbf{x}$ of the inverse filtering, and the convergence rate are spectral radii of matrices $\mathbf{I}-(\mathbf{P}_{\mathbf{H}}^{\mathrm{sym}})^{-1}\mathbf{H}$, $\mathbf{I}-\mathbf{P}_{\mathbf{H}}^{-2}\mathbf{H}^T\mathbf{H}$, $\mathbf{I}-\beta_{op}\mathbf{H}^T\mathbf{H}$ and $\mathbf{I}-\tilde{\mathbf{D}}\mathbf{H}$, see Theorems II.3 and III.1. Here the average of spectral radii in SPGDA, PGDA, OpGD and IMIA shown in Fig. 1 are $0.9786, 0.9996, 0.9993, 0.9566$ respectively. We remark that the reason for PGDA and OpGD to have slow convergence in the above simulation could be that their spectral radii are too close to 1. Our simulation shows that for the graph filter on some random geometric graphs of order $N=1024$, which has one as its diagonal entries and nondiagonal entries of $\mathbf{H}_o$ in (18) with $\gamma=0$ and $K=512$ as its nondiagonal entries, the corresponding PGDA, OpGD, SPGDA converge and the IMIA diverges. Let $\mathcal{G}_T=(V_T,E_T)$ be the undirected graph with 218 locations in the United States as vertices and edges constructed by the 5 nearest neighboring locations, and let $\mathbf{x}_{12}$ be the recorded temperature vector of those 218 locations on August 1st, 2010 at 12:00 PM, see Fig. 2 [19], [27]. In the second simulation, we consider to implement the inverse filtering procedure $\tilde{\mathbf{x}}=(\mathbf{I}+\alpha\mathbf{L}_{\mathcal{G}_T}^{\mathrm{sym}})^{-1}\mathbf{b}$ arisen from the minimization problem $\tilde{\mathbf{x}}:=\arg\min_{\mathbf{z}}\|\mathbf{z}-\mathbf{b}\|_2^2+\alpha\mathbf{z}^T\mathbf{L}_{\mathcal{G}_T}^{\mathrm{sym}}\mathbf{z}$ in denoising the hourly temperature data $\mathbf{x}_{12}$, where $\mathbf{L}_{\mathcal{G}_T}^{\mathrm{sym}}$ is the normalized Laplacian on $\mathcal{G}_T$, α is a penalty constraint and $\mathbf{b}=\mathbf{x}_{12}+\boldsymbol{\eta}$ is the temperature vector corrupted by i.i.d. random noise $\boldsymbol{\eta}$ with its components being randomly selected in $[-\eta,\eta]$ in a uniform distribution [19], [27]. Shown in Fig. 2 is the performance of the SPGDA, PGDA, OpGD, IMIA and ICPA to implement the above inverse filtering procedure with noise level $\eta=35$ and the penalty constraint $\alpha=0.9075$ [19], where ICPA is the iterative Chebyshev polynomial approximation algorithm of order one [16], [17], [19]. This indicates that the 3 rd term in ICPA, the 5th term in IMIA, the 8th term of SPGDA, the 10th term of OpGD and the 30th term of PGDA can be used as the denoised temperature vector $\tilde{\mathbf{x}}$.

To implement the inverse filter procedure (3) on SDNs, we observe from the above two simulations that OpGD outperforms PGDA while the selection of optimal step length in OpGD is computationally expensive. If the filter is positive definite, SPGDA, IMIA and ICPA may have better performance than OpGD and PGDA have. On the other hand, SPGDA always converges, but the requirement in [10, Theorem 3.2] to guarantee the convergence of IMIA may not be satisfied and ICPA is applicable for polynomial filters.

REFERENCES

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

[2] 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.

[3] R. Hebner, "The power grid in 2030," *IEEE Spectr.*, vol. 54, no. 4, pp. 50–55, Apr. 2017.

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

[5] 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.

[6] A. Sandryhaila and J. M. F. 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, Sep. 2014.

[7] 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.

[8] 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.

[9] J. Jiang, D. B. Tay, Q. Sun and S. Ouyang, "Recovery of time-varying graph signals via distributed algorithms on regularized problems," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 540–555, 2020.

[10] J. Jiang, and D. B. Tay, "Decentralised signal processing on graphs via matrix inverse approximation," *Signal Process.*, vol. 165, pp. 292–302, Dec. 2019.

[11] 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.

[12] 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, Sep. 2015.

[13] 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.

[14] 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, Sep. 2019.

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

[16] 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.

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

[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, 2020.

[19] N. Emirov, C. Cheng, J. Jiang, and Q. Sun, "Polynomial graph filter of multiple shifts and distributed implementation of inverse filtering," Mar. 2020. *arXiv:2003.11152*.

[20] 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.

[21] A. Loukas and D. Foucard, "Frequency analysis of time-varying graph signals," in *Proc. IEEE Global Conf. Signal Inf. Process.*, 2016, pp. 346–350.

[22] K. Qiu, X. Mao, X. Shen, X.Wang, T. Li, and Y. Gu, "Time-varying graph signal reconstruction," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 870883, Sep. 2017.

[23] 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.

[24] A. W. Bohannon, B. M. Sadler, and R. V. Balan, "A filtering framework for time-varying graph signals," in *Vertex-Frequency Analysis Graph Signals*, Cham, Switzerland: Springer, 2019, pp. 341–376.

[25] B. Nadler, S. Lafon, I. Kevrekidis, and R. Coifman, "Diffusion maps, spectral clustering and eigenfunctions of Fokker–Planck operators," in *Advances in Neural Information. Processing Systems. 18*, Y. Weiss, B. Schölkopf, and J. Platt, Eds. Cambridge, U.K.: MIT Press, 2006, pp. 955–962.

[26] P. Nathanael, J. Paratte, D. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, "GSPBOX: A toolbox for signal processing on graphs," Aug. 2014, *arXiv:1408.5781*.

[27] J. Zeng, G. Cheung, and A. Ortega, "Bipartite approximation for graph wavelet signal decomposition," *IEEE Trans. Signal Process.*, vol. 65, no. 20, pp. 5466–5480, Oct. 2017.