Recursive Median Filters for Time-Varying Graph Signal Denoising

David B. Tay
School of Information Technology, Deakin University, Australia

Graph Signal Processing Workshop 2023

Introduction

- Graph Signal Processing (GSP): leverages pair-wise relationship between nodes of a graph (irregular domain) to formulate operators on data/feature/signal defined over the nodes.
- Most existing graph signal operators in the literature are linear and deal with time static signals. Consider time-varying signals here.
- Nonlinear operators, such as the median, is known to out perform linear operators in traditional signal processing, especially for images.
- Propose recursive graph median filters for time-varying signals.
- Application to denoising real world sensor network data where Gaussian noise and impulse noise are simultaneous present in the data.

Previous works

- S. Segarra, A. G. Marques, G. R. Arce, and A. Ribeiro, "Center-weighted median graph filters," in *2016 IEEE GlobalSIP*.
- S. Segarra, A. G. Marques, G. R. Arce, and A. Ribeiro, "Design of weighted median graph filters," in *2017 IEEE CAMSAP*.

First proposals for graph median filters for time static signals. Application to denoising not considered.

- D. B. Tay and J. Jiang, "Time-Varying Graph Signal Denoising via Median Filters," in IEEE Trans on Cct. and Sys. II, March 2021.

Nonrecursive graph median filters for time-varying signals. Application to denoising with Gaussian or impulse noise, but not simultaneously.

Motivation

- Wireless sensor networks (WSN) for environmental monitoring: cheap sensors typically have limited computational and communication resources.
- Sensors often operate in a harsh environment: measurements can be subjected to significant levels of noise.
- Gaussian noise: from thermal sources and from limitations of the cheap sensor hardware.
- Impulsive noise: models external interferences, e.g. electromagnetic, and intermittent sensor failures.
- Both types of noise will be simultaneously present, especially in harsh environments.

GSP Basics

- A graph $G \equiv (V, E)$ consist of vertices V and edges E. Vertices usually indexed as $1, \ldots, N = |V|$.
- Adjacency matrix $\mathbf{A} = [a_{i,j}]$ contain weight of edges. No connection: $a_{i,j} = 0$. Usually $\mathbf{A}$ is sparse.
- Diagonal matrix $\mathbf{D} \equiv \text{diag}(d_i)$ where $d_i = \sum_j a_{i,j}$ is the degree.
- Graph Laplacians: $\mathbf{L} \equiv \mathbf{D} - \mathbf{A}$, $\mathbf{L}_S \equiv \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$, $\mathbf{L}_R \equiv \mathbf{D}^{-1}\mathbf{L}$.
- Graph signal $f : V \to \mathbb{R}$ represented as vector $\mathbf{f} = [f(1) \ldots f(N)]^T$. $f(i)$ represent the signal/feature value at node i.
- Linear graph signal operator: $\mathbf{f}_{out} = \mathbf{H}\mathbf{f}$.
- An important class of linear operators are polynomial functions of the Laplacian $h(\mathbf{L})$: can be implemented distributively and has localization property.

Example of graph filter

Denoising using Tikhonov regularization:

- Have a noisy version of a graph signal $\mathbf{y}$ from an underlying noiseless version $\mathbf{f}$.
- Solve

$$\min_{\hat{\mathbf{f}}} \left(||\hat{\mathbf{f}} - \mathbf{y}||^2 + \frac{\gamma}{2} \hat{\mathbf{f}}^T \mathbf{L} \hat{\mathbf{f}} \right)$$

- Closed form solution $\hat{\mathbf{f}} = \mathbf{H}_{opt}\mathbf{y}$, where the linear operator is

$$\mathbf{H}_{opt} = \left(1 + \frac{\gamma}{2}\mathbf{L}\right)^{-1}$$

- This is equivalent to a smoothing (low-pass) filter and in practice a polynomial approximation is used. The simplest is first order given by

$$\mathbf{H}_{opt,approx} = 1 - \frac{\gamma}{2}\mathbf{L}$$

which is 1-hop localized.

Time-varying graph signal

- A function $f(i, t)$ of both the vertex i and time t.
- When i is fixed, we have a regular time signal and when t is fixed, we have a static graph signal.
- Matrix representation

$$\mathbf{F} = [\mathbf{f}_1 | \mathbf{f}_2 | \cdots | \mathbf{f}_T]$$

where T is the number of time instants and the column vectors $\mathbf{f}_k$ $(k = 1, \ldots, T)$ represent the graph signal at time instant $t = k$.

Time-vertex graphs

- To capture correlation across time and vertex, product graphs, from two underlying graphs, are used.
- First underlying graph with adjacency $\mathbf{A}_G$ models the pair-wise relationship between (sensor) nodes.
- Second underlying graph is an undirected line graph which models the time correlation with the adjacency (size $T \times T$) given by

$$\mathbf{A}_T = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 1 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

Time-vertex graph (cont.)

- Define adjacency of *K-hop graph*.

$$\mathbf{A}_{G,K} = \left(\left(\sum_{k=1}^{K} \mathbf{A}_G^k \right) > \mathbf{0}_N \right) - \mathbf{I}_N$$

  direct connection of all nodes (sensors) that are K-hops away, denoted as $\mathcal{N}_G(i, K)$.

- Strong product graph:

$$\mathbf{A}_{SP} = \mathbf{I}_T \otimes \mathbf{A}_{G,K} + \mathbf{A}_T \otimes (\mathbf{A}_{G,K} + \mathbf{I}_N)$$

- Each row (or column) of the product graph adjacency corresponds to a unique combination of (sensor) node i and time instant t. The pair (i, t) is known as a *time-vertex node*.

Correlation structure

1. The adjacency $\mathbf{A}_{SP}$ defines the set of nodes whose signal values are correlated:

$$\mathcal{N}_{SP}(i,t;K) = \mathcal{N}_t^1(i) \cup \mathcal{N}_t^2(i) \cup \mathcal{N}_t^3(i) \cup \mathcal{N}_t^-(i) \cup \mathcal{N}_t^*(i) \cup \mathcal{N}_t^+(i)$$

   Partitioning into disjoint subsets.
2. Single centre time-vertex nodes at different times:

$$\mathcal{N}_t^1(i) \equiv \{(i,t-1)\}; \;\; \mathcal{N}_t^2(i) \equiv \{(i,t)\}; \;\; \mathcal{N}_t^3(i) \equiv \{(i,t+1)\}$$

3. Neighbourhood time-vertex nodes at different time instants:

$$\mathcal{N}_t^-(i) \equiv \{(j,t-1) : j \in \mathcal{N}_G(i,K)\}$$

$$\mathcal{N}_t^*(i) \equiv \{(j,t) : j \in \mathcal{N}_G(i,K)\}$$

$$\mathcal{N}_t^+(i) \equiv \{(j,t+1) : j \in \mathcal{N}_G(i,K)\}$$

Median time-vertex filter

- Median operator on a set of L numerical values $\mathcal{F} = \{f_1, f_2, \ldots, f_L\}$:

$$\Gamma(\mathcal{F}) \equiv \tilde{\mathbf{f}}_{rank} \equiv [\tilde{f}_1 \;\; \tilde{f}_2 \;\; \ldots \;\; \tilde{f}_L]; \qquad \mathrm{MED}(\mathcal{F}) \equiv \tfrac{1}{2}(\tilde{f}_{\lfloor (L+1)/2 \rfloor} + \tilde{f}_{\lfloor L/2 \rfloor + 1})$$

- When L is odd, the median gives the middle value.
- Sometimes need to repeat the values to give more weight, e.g. if $P = 2$,

$$P \diamond \{-1, 1, 3, 3\} = \{-1, -1, 1, 1, 3, 3, 3, 3\}$$

- Notation: $f(\mathcal{N}_t^*)$ denote the set of signal values over the set of nodes $\mathcal{N}_t^*$.

Median time-vertex filter (cont.)

Let $f(i,t)$ and $y(i,t)$ denote the input and output. The filter operate sequentially from time $t=1$ till $t=T$ as follows:

1. When $t=1$, the output is given by:

$$y(i,1) = \text{MED}\left(P \diamond f(\mathcal{N}^2 \cup \mathcal{N}^3) \cup f(\mathcal{N}^* \cup \mathcal{N}^+)\right)$$

2. For $t=2,\ldots,T-1$, the output is given by:

$$y(i,t) = \text{MED}\left(P \diamond (y(\mathcal{N}^1) \cup f(\mathcal{N}^2 \cup \mathcal{N}^3)) \cup y(\mathcal{N}^-) \cup f(\mathcal{N}^* \cup \mathcal{N}^+)\right)$$

3. When $t=T$, the output is given by:

$$y(i,T) = \text{MED}\left(P \diamond (y(\mathcal{N}^1) \cup f(\mathcal{N}^2)) \cup y(\mathcal{N}^-) \cup f(\mathcal{N}^*)\right)$$

PREVIOUS DENOISED VALUES ARE USED IN DENOISING CURRENT VALUES, I.E. RECURSIVE.

Noise model

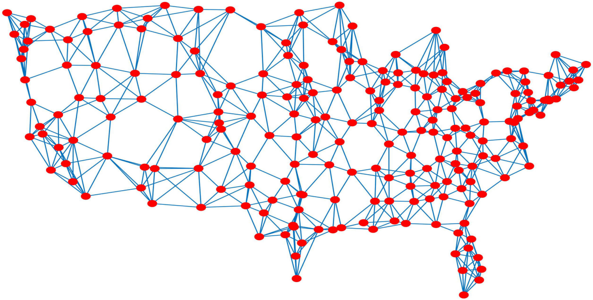
$$f(i,t) = \begin{cases} x_{min} & \text{with probability } d/2 \\ x(i,t) + n & \text{with probability } 1-d \\ x_{max} & \text{with probability } d/2 \end{cases} \quad (1)$$

- n has a Gaussian distribution with zero mean and variance σ^2.
- d is the probability (percentage) of corruption due to impulsive noise.
- x_{min} (x_{max}) is the minimum (maximum) value of the noiseless signal $x(i,t)$.

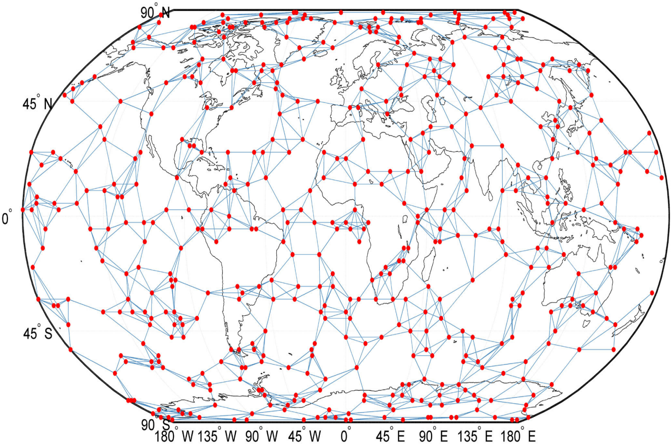
Experiments

- Use real world sensor measurements from three sources.
- Corrupted simultaneously by Gaussian noise and impulsive noise
- Compare with non-recursive median filter and linear filter.
- Found that the simplest filter with with $K = 1$ (1-hop) and $P = 1$ (no weighting) generally gives the best results.
- Many results but will only show a representative.

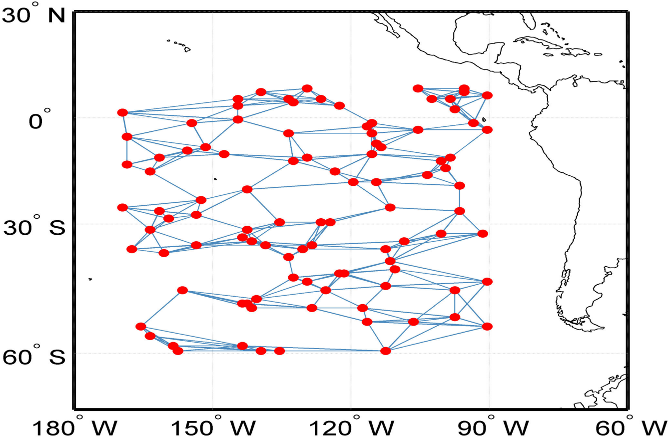
US temperature graph



Global sea-pressure graph



Pacific ocean sea-temperature graph



Results

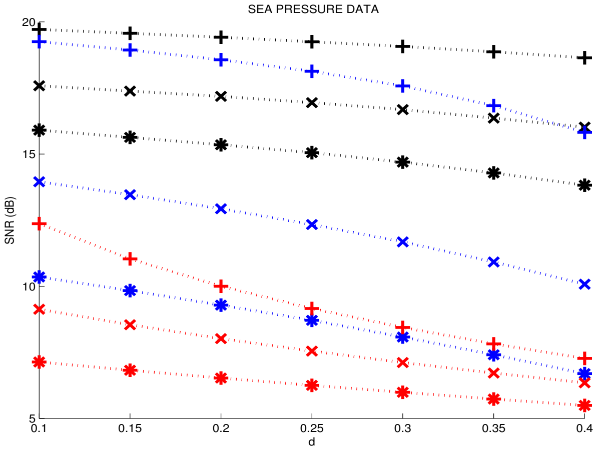


Figure: Black: REC_{SP} Blue: NMED_{SP}, Red: LIN_{SP}. Three different σ values for Gaussian noise. '+': $\sigma = 0.1$, 'x': $\sigma = 0.25$, '*': $\sigma = 0.4$. The horizontal axis d is the % corruption with impulsive noise.

Results

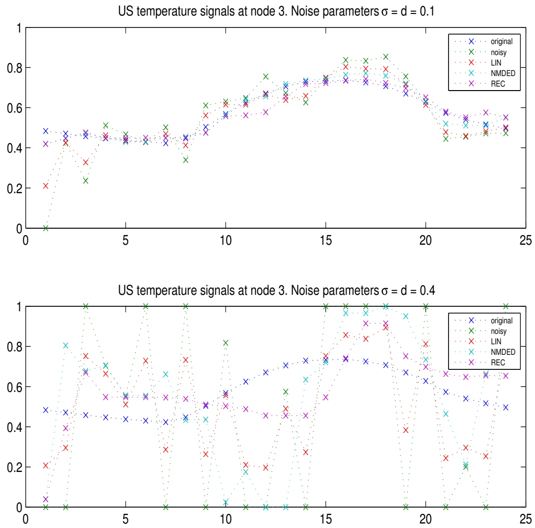


Figure: Various time signals of US temperature data at node 3.

Conclusions

- Efficient time-vertex median filters have been proposed for denoising time-varying graph signals.
- Filters can be implemented distributively using only information from immediate neighbours: suitable for resource limited sensor nodes.
- Performance tested and compared with the linear counterpart, under varying noise levels.
- Median filters superior and sometimes better by a large margin in high levels of noise situations.