This article has been accepted for publication in IEEE Signal Processing Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LSP.2023.3316095

Robust Recovery for Graph Signal via ℓ_0-Norm Regularization

Xiao Peng Li, Yi Yan, Ercan Engin Kuruoglu, Hing Cheung So, *Fellow, IEEE*, Yuan Chen

***Abstract*—Graph signal processing refers to dealing with irregularly structured data. Compared with traditional signal processing, it can preserve the complex interactions within irregular data. In this work, we devise a robust algorithm to recover band-limited graph signals in the presence of impulsive noise. First, the observed data vector is recast, such that the noise component is divided into two vectors, representing the dense-noise component and sparse outliers, respectively. We then exploit ℓ_0-norm to characterize the sparse vector as a regularization term. Alternating minimization is subsequently adopted as the solver for the resultant optimization problem. Besides, we suggest an approach to automatically update the penalty parameter of the ℓ_0-norm term. In addition, we analyze the computational complexity and the steady-state convergence of our algorithm. Experimental results on synthetic and temperature data exhibit the superiority of the developed method over state-of-the-art algorithms in impulsive noise environments in terms of recovery accuracy and convergence speed.**

***Index Terms*—Graph signal processing, impulsive noise, ℓ_0-norm, robust recovery**

I. Introduction

Graph signal processing (GSP) aims at handling multivariate irregular data [1] in a topological structure, such as regional temperature [2], traffic networks [3], macroeconomic models [4], social networks [5], and neuroimaging data [6]. Traditional signal processing methodologies attempt to model such data using regular structures in the form of vectors, matrices, or tensors, while GSP defines the data over a graph [7]. Therefore, GSP is able to consider both the structure (edge connections) and the data (values at elementary units).

As the earliest adaptive graph method, graph least mean squares (GLMS) [8] employs the LMS algorithm [9] to recover band-limited graph signals. Analogously, graph recursive least squares (GRLS) [10] extends the RLS technique [11] to a graph formulation for boosting the convergence speed. However, it suffers from high computational complexity, which imposes a practical challenge with a large number of nodes. To make a balance, graph normalized LMS (GNLMS) is proposed [2], resulting in faster convergence than GLMS, and lower complexity than GRLS. On the other hand, distributed solutions have been developed to process data collected in a distributed network, including adapt-then-combine diffusion method (ATC) [12], graph diffusion preconditioned LMS [13], and graph kernel LMS [14].

The aforementioned algorithms are derived under the ℓ_2-norm minimization framework, indicating that they rest on the Gaussian noise assumption. Therefore, their performance might degrade when the observed data are corrupted by non-Gaussian noise particularly with impulsive characteristics. In fact, impulsive noise frequently appears in real-world scenarios, including target estimation [15] and underwater communications [16]. To ensure reliable performance in impulsive noise scenario, GLM pth power (GLMP) [17] adopts the ℓ_p-norm [18] to restore band-limited graph signals. Meanwhile, the graph-sign algorithm [19] employs the ℓ_1-norm as its loss function, leading to a lower computational complexity than GLMP. Similar to GNLMS evolved from GLMS, GNLM pth power (GNLMP) [20] is suggested to enhance the convergence speed. That is to say, most existing robust methods for graph signal recovery adopt ℓ_p-norm to handle outliers. However, in practice, only a small portion of entries is corrupted by anomalies, and applying the ℓ_p-norm to all data including those without outliers will result in suboptimal performance.

In this work, we recast the observed graph signal model to effectively resist impulsive noise. The gross noise is modeled as the sum of two components, namely, dense and sparse constituents. The recovery problem is then formulated using ℓ_2-norm and ℓ_0-norm where the former is to handle Gaussian noise and the latter is to provide robustness against outliers in the observations. In addition, we adopt alternating minimization to tackle the resultant optimization problem. The computational complexity and steady-state convergence of the algorithm are also analyzed. Experimental results on synthetic data demonstrate that the proposed methodology attains a lower steady-state error and a faster convergence speed than the state-of-the-art algorithms in the presence of outliers. Meanwhile, in recovering temperature data, our method is superior to the existing methods in impulsive noise scenario.

The remainder of this paper is organized as follows. We introduce preliminaries and related works in Section II. In Section III, we re-express the observed data model and then reformulate the recovery problem. The suggested approach is then presented, followed by its computational complexity and steady-state convergence analysis. Numerical examples are included to evaluate our method by comparing with existing

This work was supported in part by the National Science Fund for Distinguished Young Scholars under Grant 61925108, the Key Project of International Cooperation and Exchanges of the National Natural Science Foundation of China under Grant 62220106009, and the project of Shenzhen Peacock Plan Teams under Grant KQTD20210811090051046. (*Corresponding author: Xiao Peng Li*)

Xiao Peng Li is with the State Key Laboratory of Radio Frequency Heterogeneous Integration, Shenzhen University, Shenzhen, 518060, China. (Email: x.p.li@szu.edu.cn.)

Yi Yan and Ercan Engin Kuruoglu are with the Tsinghua-Berkeley Shenzhen Institute, Shenzhen International Graduate School, Tsinghua University, Shenzhen, 518055, China.

Hing Cheung So is with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong SAR, China.

Yuan Chen is with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, 100083, China.

algorithms in Section IV. Finally, concluding remarks are given in Section V.

II. Background

A. Preliminaries

Given a node set $\mathcal{V}=\{1,2,\cdots,N\}$ and a weighted edge set $\varepsilon=\{a_{i,j}\}_{i,j\in\mathcal{V}}$, where $a_{i,j}$ is the edge weight from nodes i to j, such that if there is a link, $a_{i,j}>0$, or $a_{i,j}=0$ otherwise [1]. Then, a graph is defined as $\mathcal{G}=(\mathcal{V},\varepsilon)$, and its adjacency matrix is represented by $\boldsymbol{A}\in\mathbb{R}^{N\times N}$ whose (i,j) entry is $a_{i,j}$. Note that $\boldsymbol{A}$ is a symmetric matrix when the graph is undirected. Besides, its degree matrix is denoted as $\boldsymbol{K}=\mathrm{diag}(\boldsymbol{k})\in\mathbb{R}^{N\times N}$ where $k_i=\sum_{j=1}^{N}a_{i,j}$. Moreover, the graph Laplacian matrix is defined as $\boldsymbol{L}=\boldsymbol{K}-\boldsymbol{A}$, and its eigenvalue decomposition is $\boldsymbol{L}=\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, where $\boldsymbol{U}\in\mathbb{R}^{N\times N}$ is the orthonormal eigenvector matrix and $\boldsymbol{\Lambda}\in\mathbb{R}^{N\times N}$ is a diagonal matrix whose diagonal elements are the corresponding eigenvalues λ_i [8].

Analogous to Fourier transform (FT), graph FT (GFT) converts a graph signal $\boldsymbol{x}\in\mathbb{R}^N$ from the vertex domain into the spectral domain: $\boldsymbol{\epsilon}=\boldsymbol{U}^T\boldsymbol{x}$ [8]. Besides, the inverse GFT (IGFT) is represented by $\boldsymbol{x}=\boldsymbol{U}\boldsymbol{\epsilon}$, which transforms a graph signal from the spectral domain into the vertex domain [8]. Given a subset of vertices $\mathcal{S}\subseteq\mathcal{V}$, the vertex-limiting (or sampling) operator is defined as $\boldsymbol{D}_{\mathcal{S}}$ which is a diagonal matrix and its ith entry is one, if $i\in\mathcal{S}$, or zero otherwise. Similarly, considering a subset of frequency indices $\mathcal{F}=\{i\in\{1,\cdots,N\}:\epsilon_i\neq 0\}$, the filtering operator is given as $\boldsymbol{B}_{\mathcal{F}}=\boldsymbol{U}\boldsymbol{\Sigma}_{\mathcal{F}}\boldsymbol{U}^T$, where $\boldsymbol{\Sigma}_{\mathcal{F}}\in\mathbb{R}^{N\times N}$ is a diagonal matrix, such that its ith entry is one, if $i\in\mathcal{F}$, or zero otherwise [8]. It is worth mentioning that both $\boldsymbol{D}_{\mathcal{S}}$ and $\boldsymbol{B}_{\mathcal{F}}$ are self-adjoint and idempotent. If a graph signal is band-limited on $\mathcal{F}$, $\boldsymbol{B}_{\mathcal{F}}\boldsymbol{x}=\boldsymbol{x}$ holds [12].

We consider a band-limited graph signal $\boldsymbol{x}_0\in\mathbb{R}^N$ and its observations at time instant n are modeled as [8]:

$$\boldsymbol{y}[n]=\boldsymbol{D}_{\mathcal{S}}(\boldsymbol{x}_0+\boldsymbol{w}[n])=\boldsymbol{D}_{\mathcal{S}}\boldsymbol{B}_{\mathcal{F}}\boldsymbol{x}_0+\boldsymbol{D}_{\mathcal{S}}\boldsymbol{w}[n], \tag{1}$$

where $\boldsymbol{w}[n]\in\mathbb{R}^N$ is the noise vector which is assumed to be independent and identically distributed among different time instants. Furthermore, $\boldsymbol{w}[n]$ is assumed zero-mean [20].

B. Related Works

To recover the band-limited $\boldsymbol{x}_0$, GLMS [8] formulates the restoration problem as

$$\min_{\boldsymbol{x}}\mathbb{E}\{\|\boldsymbol{y}[n]-\boldsymbol{D}_{\mathcal{S}}\boldsymbol{B}_{\mathcal{F}}\boldsymbol{x}\|_2^2\},\ \text{s.t.}\ \boldsymbol{B}_{\mathcal{F}}\boldsymbol{x}=\boldsymbol{x}, \tag{2}$$

where $\mathbb{E}\{\cdot\}$ is the expectation operator. Then, stochastic gradient descent [21] is adopted as the solver, resulting in

$$\begin{aligned}\boldsymbol{x}[n+1]&=\boldsymbol{x}[n]+\mu\boldsymbol{B}_{\mathcal{F}}\boldsymbol{D}_{\mathcal{S}}(\boldsymbol{y}[n]-\boldsymbol{D}_{\mathcal{S}}\boldsymbol{B}_{\mathcal{F}}\boldsymbol{x}[n]) &(3\text{a})\\ &=\boldsymbol{x}[n]+\mu\boldsymbol{B}_{\mathcal{F}}\boldsymbol{D}_{\mathcal{S}}(\boldsymbol{y}[n]-\boldsymbol{x}[n]), &(3\text{b})\end{aligned}$$

where $\boldsymbol{x}[n]$ is the instantaneous estimate of $\boldsymbol{x}_0$ at the nth iteration and $\mu>0$ is the step-size (or learning rate). Since GLMS exploits ℓ_2-norm as the loss function, its recovery performance might be degraded in the presence of outliers.

To handle anomalies, GLMP [17] suggests employing the ℓ_p-norm with $1<p<2$, leading to

$$\min_{\boldsymbol{x}}\mathbb{E}\{\|\boldsymbol{y}[n]-\boldsymbol{D}_{\mathcal{S}}\boldsymbol{B}_{\mathcal{F}}\boldsymbol{x}\|_p^p\},\ \text{s.t.}\ \boldsymbol{B}_{\mathcal{F}}\boldsymbol{x}=\boldsymbol{x}. \tag{4}$$

Since the ℓ_p-norm with $p\in(1,2)$ is convex and smooth, the GLMP also exploits stochastic gradient descent to solve (4).

III. Proposed Algorithm

A. Algorithm Development

We first model the impulsive noise as a sum of the dense-noise constituent and sparse component, resulting in

$$\begin{aligned}\boldsymbol{y}[n]&=\boldsymbol{D}_{\mathcal{S}}(\boldsymbol{x}_0+\boldsymbol{w}[n])=\boldsymbol{D}_{\mathcal{S}}(\boldsymbol{x}_0+\widetilde{\boldsymbol{d}}[n]+\widetilde{\boldsymbol{s}}[n]) &(5\text{a})\\ &=\boldsymbol{D}_{\mathcal{S}}\boldsymbol{B}_{\mathcal{F}}\boldsymbol{x}_0+\widehat{\boldsymbol{d}}[n]+\widehat{\boldsymbol{s}}[n], &(5\text{b})\end{aligned}$$

where $\widetilde{\boldsymbol{d}}[n]\in\mathbb{R}^N$ and $\widetilde{\boldsymbol{s}}[n]\in\mathbb{R}^N$ are dense vector and sparse vector, respectively, while $\widehat{\boldsymbol{d}}[n]=\boldsymbol{D}_{\mathcal{S}}\widetilde{\boldsymbol{d}}[n]\in\mathbb{R}^N$ and $\widehat{\boldsymbol{s}}[n]=\boldsymbol{D}_{\mathcal{S}}\widetilde{\boldsymbol{s}}[n]\in\mathbb{R}^N$. Due to the sampling operator $\boldsymbol{D}_{\mathcal{S}}$, the sparsity of $\widehat{\boldsymbol{s}}[n]$ is higher than that of $\widetilde{\boldsymbol{s}}[n]$. Compared with (1), (5b) enables to exploit two norms to individually handle different noise components. We suggest adopting the ℓ_0-norm to accurately characterize $\widehat{\boldsymbol{s}}[n]$. For $\widehat{\boldsymbol{n}}[n]$, we assume that the sampled entries obey the Gaussian distribution and thus ℓ_2-norm is employed. Consequently, the recovery problem is formulated as:

$$\min_{\boldsymbol{x},\boldsymbol{s}}\mathbb{E}\{\|\boldsymbol{y}[n]-\boldsymbol{D}_{\mathcal{S}}\boldsymbol{B}_{\mathcal{F}}\boldsymbol{x}-\boldsymbol{s}\|_2^2+\lambda\|\boldsymbol{s}\|_0\},\ \text{s.t.}\ \boldsymbol{B}_{\mathcal{F}}\boldsymbol{x}=\boldsymbol{x}, \tag{6}$$

where $\lambda>0$ is the penalty parameter to control the sparsity of $\boldsymbol{s}$ and $\|\boldsymbol{s}\|_0$ is the ℓ_0-norm to count the number of non-zero entries in $\boldsymbol{s}$. Since (6) contains two variables to be determined, we exploit the alternating minimization concept [22], [23] to handle it, leading to

$$\begin{aligned}\boldsymbol{s}[n]&=\arg\min_{\boldsymbol{s}}\|\boldsymbol{y}[n]-\boldsymbol{D}_{\mathcal{S}}\boldsymbol{B}_{\mathcal{F}}\boldsymbol{x}[n]-\boldsymbol{s}\|_2^2+\lambda\|\boldsymbol{s}\|_0, &(7\text{a})\\ \boldsymbol{x}[n+1]&=\boldsymbol{x}[n]+\mu\boldsymbol{B}_{\mathcal{F}}\boldsymbol{D}_{\mathcal{S}}(\boldsymbol{y}[n]-\boldsymbol{x}[n]-\boldsymbol{s}[n]). &(7\text{b})\end{aligned}$$

For (7a), an optimal solution is [24]:

$$\boldsymbol{s}[n]=\mathcal{T}_{\lambda}(\widehat{\boldsymbol{w}}[n])=\begin{cases}\widehat{w}[n]_i, & \text{if } |\widehat{w}[n]_i|\geq\sqrt{\lambda},\\ 0, & \text{otherwise.}\end{cases} \tag{8}$$

where $\widehat{\boldsymbol{w}}[n]=\boldsymbol{y}[n]-\boldsymbol{D}_{\mathcal{S}}\boldsymbol{B}_{\mathcal{F}}\boldsymbol{x}[n]$ and $\widehat{w}[n]_i$ is the ith entry of $\widehat{\boldsymbol{w}}[n]$. That is, given $\boldsymbol{y}[n]$ and $\boldsymbol{x}[n]$, $\boldsymbol{s}[n]$ depends on the selection of λ. We first explain (7a) and (7b), and then discuss how to set λ. For (7a), it updates $\boldsymbol{s}$ while fixing $\boldsymbol{x}[n]$, such that outliers can be separated from the fitting error, namely, $\widehat{\boldsymbol{w}}[n]$. In iterative multi-variable optimization, it is often not so beneficial to optimize one variable perfectly while fixing the remaining variables, then the other, and repeat the whole procedure [25]. This is because a perfect optimization of one part is rendered obsolete when the other part is altered. Therefore, (7b) exploits gradient descent to update $\boldsymbol{x}$ with one iteration.

In (8), $\sqrt{\lambda}$ can be deemed as the threshold to differentiate normal entries and anomalies in the residual $\widehat{\boldsymbol{w}}[n]$. Under the assumption that the mean of $\widehat{\boldsymbol{w}}[n]$ is 0, $-\sqrt{\lambda}\leq w_i\leq\sqrt{\lambda}$ can be considered as a confidence interval to recognize outliers. Therefore, we suggest adopting the normalized median absolute deviation method [26], [27] to determine $\sqrt{\lambda}$:

$$\sqrt{\lambda}=\zeta\times 1.4826\times\mathrm{Med}(|\widehat{\boldsymbol{w}}[n]-\mathrm{Med}(\widehat{\boldsymbol{w}}[n])|), \tag{9}$$

where $\mathrm{Med}(\cdot)$ is the sample median operator and ζ controls the confidence interval range. Using a rule of thumb, we suggest

This article has been accepted for publication in IEEE Signal Processing Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LSP.2023.3316095

Algorithm 1 GAM for robust graph signal recovery

Input: $\boldsymbol{y}[n]$, $\boldsymbol{D}_{\mathcal{S}}$, $\boldsymbol{B}_{\mathcal{F}}$, μ, ζ, and maximum iteration number $N_{\max}$

Initialize: $\boldsymbol{x}[0] = \boldsymbol{0}$

for $n = 1, 2, \cdots$ **do**

1. Update $\lambda = (\zeta \times 1.4826 \times \mathrm{Med}(|\widehat{\boldsymbol{w}}[n] - \mathrm{Med}(\widehat{\boldsymbol{w}}[n])|))^2$
2. Update $\boldsymbol{s}[n] = \mathcal{T}_{\lambda}(\boldsymbol{y}[n] - \boldsymbol{D}_{\mathcal{S}}\boldsymbol{B}_{\mathcal{F}}\boldsymbol{x}[n])$
3. Update $\boldsymbol{x}[n+1] = \boldsymbol{x}[n] + \mu \boldsymbol{B}_{\mathcal{F}}\boldsymbol{D}_{\mathcal{S}}(\boldsymbol{y}[n] - \boldsymbol{x}[n] - \boldsymbol{s}[n])$

Stop if $n = N_{\max}$.

end for

Output: $\boldsymbol{x}[n+1]$.

setting the value of ζ between 2 and 5. Specifically, if the noise distribution has a heavy tail, ζ should be small. This is because a small ζ generates a narrow confidence interval to identify more outliers. When the tail is light, ζ can be set as a large value.

The proposed method is referred to as **g**raph **a**lternating **m**inimization (GAM) whose procedure is summarized in Algorithm 1.

B. Computational Complexity

In this subsection, we analyze the computational complexity of the developed method. For λ, $\mathrm{Med}(\cdot)$ operator requires performing the sorting procedure and the complexity of the heap sort method is $\mathcal{O}(N\log(N))$. Besides, the complexity of updating $\boldsymbol{s}$ and $\boldsymbol{x}$ is dominated by the operation of $\boldsymbol{B}_{\mathcal{F}}\boldsymbol{D}_{\mathcal{S}}$. In general, $\boldsymbol{B}_{\mathcal{F}}\boldsymbol{D}_{\mathcal{S}}$ requires the complexity of $\mathcal{O}(N^3)$. However, since $\boldsymbol{D}_{\mathcal{S}}$ is a diagonal matrix, its complexity reduces to $\mathcal{O}(N^2)$. Therefore, the overall complexity is $\mathcal{O}(N^2 + N\log(N))$ per iteration.

Table I tabulates the computational requirements of the proposed method as well as GLMP and GNLM, and it is clear that GAM involves the minimum complexity.

TABLE I: Complexity comparison of robust algorithms

Method	Computational complexity
GAM	$\mathcal{O}(N^2 + N\log(N))$
GLMP	$\mathcal{O}(N^3)$
GNLMP	$\mathcal{O}(N^3 + N\log(N))$

C. Steady-State Convergence

To study the steady-state convergence behavior, we first introduce the following lemma.

Lemma 1. *For truncated operator $\mathcal{T}_{\lambda}(\widehat{\boldsymbol{w}}[n])$, it is equivalent to $\widetilde{\boldsymbol{D}}[n]\widehat{\boldsymbol{w}}[n]$, where $\widetilde{\boldsymbol{D}}[n]$ is a binary diagonal matrix, such that*

$$d_{i,i} = \begin{cases} 1, & \text{if } |\widehat{w}_i[n]| \geq \sqrt{\lambda}, \\ 0, & \text{otherwise}. \end{cases}$$

Proof: It is easy to verify Lemma 1 since the truncated operator retains the entries with large magnitudes and sets the rest to zero.

Then, the steady-state convergence behavior is analyzed in the following theorem.

Theorem 1. *If the learning rate satisfies:*

$$0 < \mu < \frac{2}{\widetilde{\delta}_{\max}}, \tag{10}$$

where $\widetilde{\delta}_{\max}$ is the largest eigenvalue of $\boldsymbol{B}\boldsymbol{D}_{\mathcal{S}}(\boldsymbol{I} - \widetilde{\boldsymbol{D}}[i])$ for $i = 1, \cdots, n$, the steady-state mean-squared deviation (MSD) of GAM is convergent.

The proof is provided in the supplementary material.

IV. Experimental Results

In this section, we compare GAM with GLMS [8], GNLMS [2], GLMP [17], and GNLMP [20] on two different impulsive noise models, i.e., symmetric α-stable (SαS) distribution and Gaussian mixture model (GMM). The characteristic function of the SαS distribution with zero-location is $\varphi(\omega) = \exp(-\gamma^{\alpha}|\omega|^{\alpha})$ [28], where $0 < \alpha \leq 2$ is the characteristic exponent that describes the tail of the distribution and $\gamma > 0$ is a scale factor. When $\alpha < 2$, the SαS distribution exhibits heavy tails and thus is impulsive [29]. On the other hand, the probability density function (PDF) of two-term GMM is $p_{\omega}(\omega) = \frac{c_1}{\sqrt{2\pi}\sigma_1}\exp\left(-\frac{\omega^2}{2\sigma_1^2}\right) + \frac{c_2}{\sqrt{2\pi}\sigma_2}\exp\left(-\frac{\omega^2}{2\sigma_2^2}\right)$ [30], where $c_1 + c_2 = 1$ with $0 < c_i < 1$, σ_1^2 and σ_2^2 are the corresponding variances. When $\sigma_1^2 \geq \sigma_2^2$ and $c_1 < c_2$, GMM indicates that sparse large samples with σ_1^2 and c_1 are embedded in Gaussian background noise with σ_2^2 and c_2. We set $\sigma_1^2 = 100\sigma_2^2$ and $c_1 = 0.1$, signifying that 10% noise samples are considered as outliers. Besides, the signal-to-noise ratio (SNR) in dB is defined as $\mathrm{SNR} = 10\log_{10}\left(\frac{\|\boldsymbol{x}_0\|_2^2}{N(c_1\sigma_1^2 + c_2\sigma_2^2)}\right)$, where $c_1\sigma_1^2 + c_2\sigma_2^2$ is the total noise variance.

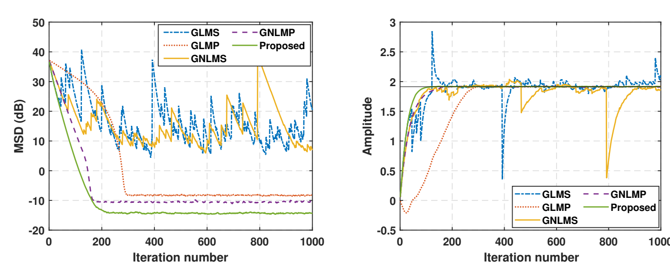
The steady-state MSD is adopted as the performance metric, defined as

$$\mathrm{MSD}[n] = 10\log_{10}(\mathbb{E}\{\|\boldsymbol{x}[n] - \boldsymbol{x}_0\|_2^2\}), \tag{11}$$

that is, the average MDS at the nth step based on 100 independent trials.

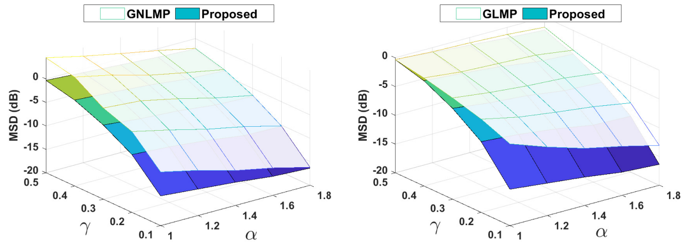
A. Synthetic Data

We adopt the Python PyGSP package [31] to generate a random sensor graph with 50 nodes, where $\boldsymbol{A} \in \mathbb{R}^{50\times 50}$ and $\boldsymbol{x}_0 \in \mathbb{R}^{50}$. As this $\boldsymbol{x}_0$ is not bandlimited, we exploit the strategy in [10] to make it bandlimited, and produce $\boldsymbol{B}_{\mathcal{F}} \in \mathbb{R}^{50\times 50}$ and $\boldsymbol{D}_{\mathcal{S}} \in \mathbb{R}^{50\times 50}$ with $|\mathcal{F}| = 20$ and $|\mathcal{S}| = 30$.



(a) MSD versus iteration number (b) Recovery amplitude versus iteration number

Fig. 1: Performance comparison in SαS noise with $\alpha = 1.2$ and $\gamma = 0.1$



(a) GNLMP versus GAM (b) GLMP versus GAM

Fig. 2: Phase transition in SαS noise.

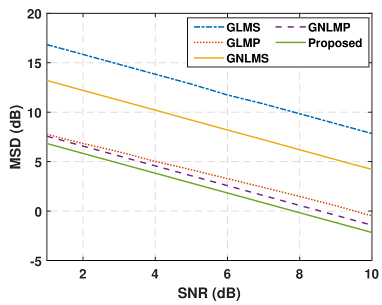


Fig. 3: MSD versus SNR in GMM noise.

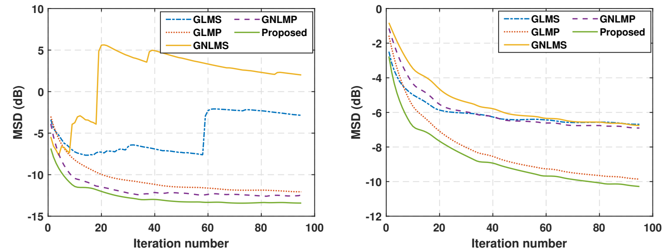
In addition, impulsive noise vector $\boldsymbol{w} \in \mathbb{R}^{50}$ is generated using SαS distribution or GMM. Note that the parameter ζ of our method is set as 4, while the values of p of the ℓ_p-norm-based algorithms follow their recommendations.

We first compare the proposed method with the competing algorithms in SαS noise. Fig. 1 (a) shows MSD versus iteration number. The learning rates of GLMP, and GNLMP are set as 0.1 and 0.028 which are their recommendation values. Then, the learning rate of GAM is chosen as 0.1 such that it attains a comparable convergence speed. We observe that the GAM, GLMP, and GNLMP achieve better recovery performance than GLMS and GNLMS. In addition, the GAM attains lower MSDs than the existing robust methods. On the other hand, we plot the recovery amplitude of one node in Fig. 1 (b). It is observed that the GAM quickly recovers the node amplitude and attains smaller MSDs than its competitors.

Moreover, we investigate recovery performance versus noise intensity in SαS noise or GMM noise. Fig. 2 depicts the results of three robust algorithms in SαS noise. Since the SNR of the SαS noise is meaningless with $\alpha < 2$, we exploit different γ and α to control noise intensity. It is observed that the GAM attains lower MSDs than GNLMP and GLMP in the whole ranges of γ and α. Besides, when noise is strong, the superiority of GAM is significant. Fig. 3 plots MSD versus SNR in GMM noise. We see that GAM, GNLMP, and GLMP achieve better recovery performance than GLMS and GNLMS. In addition, GAM attains smaller MSDs than the remaining robust methods.

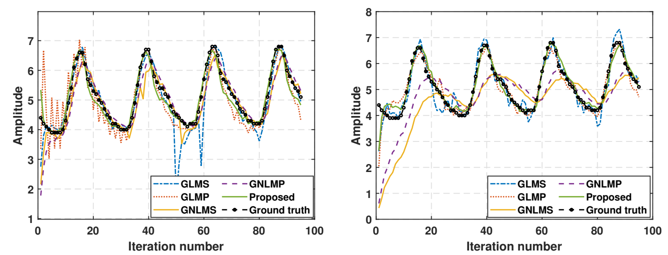
B. Temperature Data

We further compare different methods using temperature data, namely, the graph signal of hourly temperature collected from national centers for environmental information[1]. The graph topology is generated using the approach in [2], resulting in 197 nodes and 95 time slot numbers, respectively. These temperature data are then added with the synthetic impulsive noise in the following experiments.



(a) SαS noise with $\alpha = 1.2$ and $\gamma = 0.1$ (b) GMM noise with $\mathrm{SNR} = 1$

Fig. 4: MSD versus iteration number on temperature data.



(a) SαS noise with $\alpha = 1.2$ and $\gamma = 0.1$ (b) GMM noise with $\mathrm{SNR} = 1$

Fig. 5: Recovery amplitude of a selected node versus iteration number in impulsive noise.

Figs. 4 (a) and (b) plot the recovery performance in the presence of SαS noise and GMM noise, respectively. It is observed that the proposed method achieves lower MSDs than its competitors in both cases.

Moreover, Figs. 5 (a) and (b) show the recovery behavior of one node with SαS noise and GMM noise, respectively. It is seen that our algorithm attains better fitting performance than the existing methods in both noise environments.

V. Conclusion

In this work, the noise component of the observed graph data vector corrupted by impulsive noise is considered to be the sum of Gaussian noise and sparse outliers. We exploit ℓ_2-norm and ℓ_0-norm to handle Gaussian noise and sparse anomalies, respectively, in order to attain robust recovery. Alternating minimization is applied to solve the resultant problem. Besides, we propose a strategy to update the penalty parameter of the ℓ_0-norm regularization term to boost recovery. In addition, the algorithm computational complexity and steady-state convergence are analyzed. Numerical results on synthetic and temperature data demonstrate that the devised method attains higher recovery accuracy than existing algorithms in SαS noise or GMM noise.

[1]https://www.ncei.noaa.gov/data/normals-hourly/2006-2020/

REFERENCES

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

[2] M. J. M. Spelta and W. A. Martins, "Normalized LMS algorithm and data-selective strategies for adaptive graph signal estimation," *Signal Process.*, vol. 167, p. 107326, 2020.

[3] D. M. Mohan, M. T. Asif, N. Mitrovic, J. Dauwels, and P. Jaillet, "Wavelets on graphs with application to transportation networks," in *Intell. Transp. Syst.*, Qingdao, China, Oct. 2014, pp. 1707–1712.

[4] A. A. Keller, "Graph theory and economic models: From small to large size applications," *Electron. Notes Discrete Math.*, vol. 28, pp. 469–476, 2007.

[5] F. Ji and W. P. Tay, "A Hilbert space theory of generalized graph signal processing," *IEEE Trans. Signal Process.*, vol. 67, no. 24, pp. 6188–6203, 2019.

[6] M. Ménoret, N. Farrugia, B. Pasdeloup, and V. Gripon, "Evaluating graph signal processing for neuroimaging through classification and dimensionality reduction," in *IEEE Glob. Conf. Signal Inf. Process.*, Montreal, Quebec, Canada, Nov. 2017, pp. 618–622.

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

[8] P. Di Lorenzo, S. Barbarossa, P. Banelli, and S. Sardellitti, "Adaptive least mean squares estimation of graph signals," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 4, pp. 555–568, 2016.

[9] S. Haykin and B. Widrow, *Least-Mean-Square Adaptive Filters*. Wiley Online Library, 2003.

[10] P. Di Lorenzo, P. Banelli, E. Isufi, S. Barbarossa, and G. Leus, "Adaptive graph signal processing: Algorithms and optimal sampling strategies," *IEEE Trans. Signal Process.*, vol. 66, no. 13, pp. 3584–3598, 2018.

[11] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2275–2285, 2004.

[12] P. Di Lorenzo, P. Banelli, S. Barbarossa, and S. Sardellitti, "Distributed adaptive learning of graph signals," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4193–4208, 2017.

[13] F. Hua, R. Nassif, C. Richard, H. Wang, and A. H. Sayed, "Online distributed learning over graphs with multitask graph-filter models," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 63–77, 2020.

[14] V. R. Elias, V. C. Gogineni, W. A. Martins, and S. Werner, "Adaptive graph filters in reproducing kernel Hilbert spaces: Design and performance analysis," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 7, pp. 62–74, 2020.

[15] X. P. Li and H. C. So, "Robust low-rank tensor completion based on tensor ring rank via $\ell_{p,\epsilon}$-norm," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3685–3698, May 2021.

[16] J. Wang, J. Li, S. Yan, W. Shi, X. Yang, Y. Guo, and T. A. Gulliver, "A novel underwater acoustic signal denoising algorithm for Gaussian/non-Gaussian impulsive noise," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 429–445, 2020.

[17] N. H. Nguyen, K. Doğançay, and W. Wang, "Adaptive estimation and sparse sampling for graph signals in alpha-stable noise," *Digital Signal Process.*, vol. 105, p. 102782, 2020.

[18] Y. Chen, H. C. So, and E. E. Kuruoglu, "Variance analysis of unbiased least ℓ_p-norm estimator in non-Gaussian noise," *Signal Process.*, vol. 122, pp. 190–203, 2016.

[19] Y. Yan, E. E. Kuruoglu, and M. A. Altinkaya, "Adaptive sign algorithm for graph signal processing," *Signal Process.*, vol. 200, p. 108662, 2022.

[20] Y. Yan, R. Adel, and E. E. Kuruoglu, "Graph normalized-LMP algorithm for signal estimation under impulsive noise," *J. Signal Process. Syst.*, pp. 1–12, Aug. 2022.

[21] A. H. Sayed, *Adaptive Filters*. New York, NY, USA: John Wiley & Sons, 2011.

[22] U. Niesen, D. Shah, and G. W. Wornell, "Adaptive alternating minimization algorithms," *IEEE Trans. Inf. Theory*, vol. 55, no. 3, pp. 1423–1429, 2009.

[23] X. P. Li, Q. Liu, and H. C. So, "Rank-one matrix approximation with ℓ_p-norm for image inpainting," *IEEE Signal Process. Lett.*, vol. 27, pp. 680–684, 2020.

[24] A. Beck, *First-Order Methods in Optimization*. SIAM, 2017.

[25] Q. Liu and X. Li, "Efficient low-rank matrix factorization based on $\ell_{1,\varepsilon}$-norm for online background subtraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4900–4904, Jul. 2021.

[26] A. M. Zoubir, V. Koivunen, E. Ollila, and M. Muma, *Robust Statistics for Signal Processing*. Cambridge Univ. Press, 2018.

[27] A. M. Zoubir, V. Koivunen, Y. Chakhchoukh, and M. Muma, "Robust estimation in signal processing: A tutorial-style treatment of fundamental concepts," *IEEE Signal Process. Mag.*, vol. 29, no. 4, pp. 61–80, 2012.

[28] N. H. Nguyen, K. Doğançay, and E. E. Kuruoğlu, "An iteratively reweighted instrumental-variable estimator for robust 3-D AOA localization in impulsive noise," *IEEE Trans. Signal Process.*, vol. 67, no. 18, pp. 4795–4808, 2019.

[29] Y. Chen, E. E. Kuruoglu, and H. C. So, "Optimum linear regression in additive Cauchy–Gaussian noise," *Signal Process.*, vol. 106, pp. 312–318, 2015.

[30] X. P. Li, Z.-L. Shi, Q. Liu, and H. C. So, "Fast robust matrix completion via entry-wise ℓ_0-norm minimization," *IEEE Trans. Cybern.*, 2022, Early Access.

[31] M. Defferrard, L. Martin, R. Pena, and N. Perraudin, "PyGSP: Graph signal processing in python," *[Online]. Available: https://github.com/epfl-lts2/pygsp*, 2017.