Short communication

Adaptive sign algorithm for graph signal processing

Yi Yan [a], Ercan E. Kuruoglu [a,*], Mustafa A. Altinkaya [b]

[a] *Tsinghua-Berkeley Shenzhen Institute, Tsinghua University, Shenzhen, China*
[b] *Izmir Institute of Technology, Izmir, Turkey*

ARTICLE INFO

ABSTRACT

Efficient and robust online processing techniques for irregularly structured data are crucial in the current era of data abundance. In this paper, we propose a graph/network version of the classical adaptive Sign algorithm for online graph signal estimation under impulsive noise. The recently introduced graph adaptive least mean squares algorithm is unstable under non-Gaussian impulsive noise and has high computational complexity. The Graph-Sign algorithm proposed in this work is based on the minimum dispersion criterion and therefore impulsive noise does not hinder its estimation quality. Unlike the recently proposed graph adaptive least mean pth power algorithm, our Graph-Sign algorithm can operate without prior knowledge of the noise distribution. The proposed Graph-Sign algorithm has a faster run time because of its low computational complexity compared to the existing adaptive graph signal processing algorithms. Experimenting on steady-state and time-varying graph signals estimation utilizing spectral properties of bandlimitedness and sampling, the Graph-Sign algorithm demonstrates fast, stable, and robust graph signal estimation performance under impulsive noise modeled by alpha stable, Cauchy, Student's t, or Laplace distributions.

© 2022 Elsevier B.V. All rights reserved.

1. Introduction

Graph-based data structures are gaining popularity in recent years due to the effective power of graphs in representing multivariate irregular data in fields such as data science, information engineering, bioinformatics, and finance [1–4]. However, with this increasing popularity of the utilization of graphs, traditional data processing techniques that were optimized on structured data could not adapt to the structural irregularities and could not utilize the intrinsic relationships among data seen in or modeled by graphs, which led to a demand for algorithms that could process graph-structured data efficiently [1–3]. The recently emerged Graph signal processing (GSP) techniques provide efficient solutions to deal with the irregularities in real applications such as in modeling brain functional connectivity [5], spatial temperature data[6], transportation flows [7], monitoring 5G signal strength [8], sensor networks in smart cities [9], structuring geometric data [10], and modeling transportation flows [7]. GSP provides, in the graph domain, classical discrete signal processing concepts through graph shift, graph convolution, graph Fourier transform (GFT), and graph wavelet transform; the basic classic filtering operations such

as low-pass, high-pass filters, and band-pass filters, or the notion of FIR and IIR filters exist in GSP as well [2,3,11,12]. GSP-based algorithms have the capability of solving various classical machine learning tasks such as classification [13] and clustering [14]. GSP techniques are also seen in the foundation of spectral graphical deep learning algorithms, such as ChebNet [15] and graph convolutional networks [13], where nonlinear activations are combined with GSP techniques to incorporate non-linear modelling capability into graphs. The GSP components in the graph neural networks provide model interpretability, which was previously lacking in the non-graphical deep learning algorithms [4]. However, these algorithms can only handle static tasks that do not operate in real-time; while many real-life data are not static which underlines the need for online data processing techniques.

In classical signal processing, online estimation of time-varying signals is often accomplished using adaptive filters [16]. Adaptive GSP algorithms are inspired by classical adaptive filters to perform online estimation of steady-state and time-varying graph signals through spectral methods [6,8,17,18]. Analogous to the famous adaptive least mean squares (LMS) algorithm in classical adaptive filtering, the Graph least mean squares algorithm (GLMS) is popular due to its simplicity by using l_2-norm optimization which implicitly makes the Gaussian noise assumption [8]. Other least-squares-based adaptive GSP algorithms such as the Graph normalized LMS (GNLMS) [6] and the Graph recursive least squares (GRLS)

---

algorithm [17] are GSP analogues of classical adaptive filtering algorithms developed for faster convergence. Recent attention has focused on the distributed version of GSP algorithms [19,20] which merge the distributed GLMS algorithm with diffusion algorithms and [21] gives a distributed version of the GRLS algorithm. Similarly, distributed kernel GLMS was introduced in [22] which extended the linear graph adaptive algorithm to a nonlinear version. Recent studies have applied classical time-series analysis techniques, such as the ARMA models, to GSP [23]. To tackle the time-varying nature of some real-world data, another line of work known as Time-vertex Signal Processing was proposed to take into account the time-domain information [24]. This formulation combines DFT on the time series data and GFT on the graph.

All these mentioned work implicitly make Gaussian noise assumption. However, the ambient noise in various real-world applications is non-Gaussian with impulsive characteristics which can be modeled by heavy-tailed distributions [25–29]. Least-squares-based adaptive GSP algorithms, namely the GLMS, the GRLS, and the GNLMS, become unstable and diverge under impulsive noise due to large or infinite variance [30]. The GRLS algorithm further requires prior knowledge of the covariance matrix, which may not be accessible in many situations [21]. In (non-Graph) adaptive filtering, to effectively mitigate the influence of heavy-tailed non-Gaussian noise, the minimum absolute deviations (MAD) criterion and minimum dispersion criterion were used, which gave rise to the Sign or the least mean absolute deviation (LMAD) algorithm and the least mean p^{th} power (LMP) algorithm respectively [30,31]. Enriching the adaptation with a localised information-theoretic measure, the maximum correntropy criterion was introduced in adaptive signal processing which has demonstrated robust regression and filtering results under impulsive noise [32,33]. However, compared to the Sign or the LMAD algorithm the maximum correntropy algorithms have higher computation complexity since the updates in the Sign or the LMAD algorithm require only a sign operation on the error term explaining the popularity of the Sign algorithm in adaptive signal processing. In order to avoid the drawbacks of l_2-norm optimization when the noise distribution is impulsive and non-Gaussian in a multivariate data problem, the Graph-LMP algorithm (GLMP) was proposed which assumes that the noise is symmetric α-stable distribution (SαS) [18]. This assumption leads to the minimum dispersion criterion which can be optimized via l_p-norm minimization instead of l_2-norm minimization and the resulting algorithm is a direct extension of the classical LMP algorithm. Despite its generality for varying degrees of impulsiveness of the noise, there are two drawbacks of the GLMP algorithm: it requires additional computations to the already expensive GLMS algorithm, and its parameter selection is still based on prior knowledge of the noise.

In this paper, we propose a lower-cost alternative, namely the adaptive Graph-Sign algorithm (G-Sign) as a graph extension of the classical adaptive sign error (SIGN) or the LMAD algorithm for multivariate signals [16,31]. The proposed G-Sign algorithm is derived based on the minimum absolute deviations criterion which is a special case of the minimum least l_p-norm optimization for $p = 1$ and leads to l_1-norm optimization, which removes the need for prior knowledge of any noise parameters or noise model assumption [34]. This allows the G-Sign algorithm to avoid the instability seen in the least-squares-based algorithms when estimating graph signals under impulsive noise. In classical adaptive filtering, the SIGN algorithm is known for its simplicity and time efficiency compared to the LMS, LMP, and maximum correntropy algorithms. This characteristic is inherited in the graph case: the G-Sign algorithm further reduces computational complexity to estimate a steady-state graph signal compared to the GLMP and the GLMS algorithms, making the G-Sign algorithm time efficient. The G-Sign algorithm is robust when estimating time-varying graph

signals under impulsive noise, making it capable of performing on-line graph signal estimation. Note that the GSP adaptive algorithms update the graph-signal estimates instead of the weights for the data samples as done by classical adaptive algorithms. The desired parallelism of the inference in predicting the outcomes of typically multiple hundreds of nodes is made possible by the graph version of the signal processing algorithms. Dimensionality reduction is achieved by localizing the graph signal by spatial sampling with various sampling strategies and spectral bandlimitedness [17,35]. Introducing a large number of weights that needs to be jointly optimized is mainly avoided by preferring a data-centered approach in graph-based gradient methods.

The remaining sections of this paper are organized as follows. The background information on GSP and noise modeling are in Section 2. In Section 3, we derive the G-Sign algorithm and analyze its computational complexity. In Section 4, we provide the first-order and second-order steady-state stability analysis for the G-Sign algorithm. The experimental studies and results are in Section 5. Section 6 provides the conclusions.

2. Background

2.1. GSP preliminaries

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined with a set of N nodes $\mathcal{V} = v_1 \ldots v_N$, and a set of M edges $\mathcal{E} = e_1 \ldots e_M$ representing the connections between nodes. In this paper, the graph $\mathcal{G}$ of interest is always undirected and can be either weighted or unweighted. A graph signal $\boldsymbol{x}$ is a graph with a function value defined on the nodes. The adjacency matrix $\mathbf{A}$ of the graph $\mathcal{G}$ represents the connectivity of the edges in $\mathcal{E}$. The ij^{th} entry of $\mathbf{A}$ is the edge weight from node v_i to node v_j when $\mathcal{G}$ is weighted or simply $\mathbf{A}_{ij} = 1$ when there is an edge between node v_i and node v_j when $\mathcal{G}$ is unweighted. For an undirected graph, the adjacency matrix $\mathbf{A}$ is symmetric. If $\mathcal{G}$ is undirected and unweighted, the number of edges a node v_i has is the node degree d_i, where we can formulate a diagonal matrix called the degree matrix $\mathbf{D} = \text{diag}(d_1 \ldots d_N)$. In the weighted case, the degree of a node is the summation of all of the edge weights instead of simply counting the number of edges. The graph Laplacian matrix $\mathbf{L}$, which combines the information from $\mathbf{A}$ and $\mathbf{D}$, is defined simply as $\mathbf{L} = \mathbf{D} - \mathbf{A}$.

The GFT is based on the eigenvector decomposition $\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$, where $\mathbf{U}$ is the orthonormal eigenvector matrix and $\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_N]^T$. The GFT transforms a graph signal $\boldsymbol{x}$ from spatial-domain to spectral domain by projecting $\boldsymbol{x}$ onto $\mathbf{U}$: $\boldsymbol{s} = \mathbf{U}^T\boldsymbol{x}$. Spectral-domain operations could be performed similarly as in the classical Fourier transform case by defining a filter $\mathbf{H}(\lambda)$ and then applied using the convolution property of Fourier transform. A graph signal transformed to the spectral domain could utilize the inverse graph Fourier transform (IGFT) $\boldsymbol{x} = \mathbf{U}\boldsymbol{s}$ to transform back to the spatial domain. Here is a basic yet complete GSP procedure to apply the filter to $\boldsymbol{x}$ and generate a processed graph signal $\boldsymbol{x}_p = \mathbf{U}\mathbf{H}(\lambda)\mathbf{U}^T\boldsymbol{x}$. A graph signal is sparse in the spectral sense when it is bandlimited in the spectral domain. A bandlimiting filter $\boldsymbol{\Sigma}$ can be defined based on a frequency set $\mathcal{F}$, where $\mathbf{H}(\lambda) = \boldsymbol{\Sigma} = \text{diag}(\mathbf{1}_{\mathcal{F}}(\lambda))$, with $\mathbf{1}_{\mathcal{F}}(\lambda_i) = 1$ if $\lambda_i \in \mathcal{F}$ and 0 otherwise. A graph signal with reduced number of nodes sampled based on a node sampling set $\mathcal{S} \subseteq \mathcal{V}$ is sparse in the spatial domain; $\mathbf{D}_{\mathcal{S}}$ is the sampling matrix and its only non-zero elements are given with $\mathbf{D}_{\mathcal{S}_{ii}} = 1 \forall v_i \in \mathcal{S}$ [17]. Both $\mathbf{D}_{\mathcal{S}}$ and $\boldsymbol{\Sigma}$ are idempotent and self-adjoint matrices.

2.2. Impulsive distributions

Algorithms based on least-squares estimation are based on the Gaussian noise assumption, which has proven to be an oversimpli-
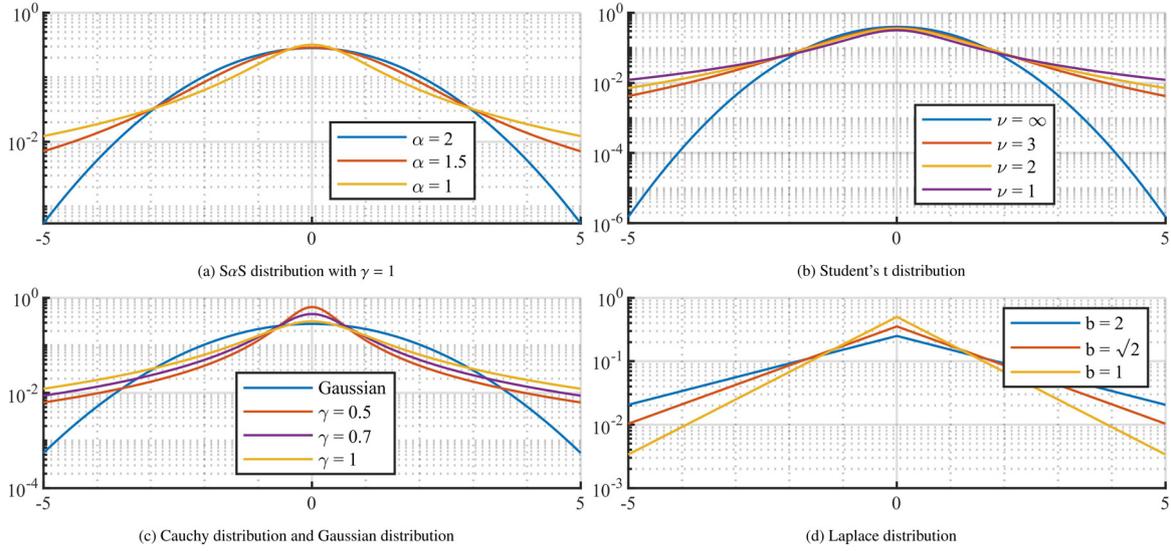
Fig. 1. PDFs of various non-Gaussian distributions all with $\mu = 0$ in logarithmic scale.

fication of noise in various real-world applications. In this paper, the SαS, Cauchy, Student's t, and Laplace distributions are considered as additive noise to reflect the impulsiveness of real-life noises: in underwater communications the noise can be modeled using Cauchy-Gaussian mixture [25] or using Student's t distribution [26], in powerline communications and in sensing applications such as source localization the noise is modeled using α-stable distributions [27,36], in carbon nanotube detectors the electrical noise follows Cauchy distribution [28], and in [29] the noise in matrix factorization problems of computer vision tasks is assumed to follow a mixture of Laplace distribution. The Probability Density Functions (PDF) of these non-Gaussian distributions are illustrated in Fig. 1 and are discussed in this section. As shown in Fig. 1, in certain parameter set up of these distortions, heavy-tailed behavior is observed, which generate outliers in the data which can be considered as additive noise. Least squares based algorithms show unstable behavior under impulsive noise due to large or infinite variance, therefore least l_p-norm based methods were proposed which provide more stable estimators [31,37].

The SαS distribution satisfies the generalized central limit theorem and is governed by the characteristic exponent α, the location parameter μ, and the scale parameter γ. The mean of SαS is only defined when $\alpha > 1$: with μ being the mean when $1 < \alpha \leqslant 2$, and the median when $\alpha < 1$. The variance of SαS is defined only when $\alpha = 2$, and in the other cases the scale parameter is given by the dispersion [31]. Gaussian distribution is obtained by setting $\alpha = 2$, and Cauchy distribution when $\alpha = 1$. From Fig. 1a we can observe that the tail of the SαS distribution is algebraic and hence heavy except for $\alpha = 2$. The SαS has no analytic PDF for other α values but has the characteristic function

$$\boldsymbol{\phi}(t) = \exp\{j\mu t - \gamma |t|^{\alpha}\}. \tag{1}$$

The Student's t distribution is a heavy-tailed distribution for estimating the mean of Gaussian distribution under unknown variance and small sample size. The Student's t distribution is parameterised by degrees of freedom ν and can be defined as the distribution of the location of the sample mean relative to the true mean, divided by the sample standard deviation. The distribution has infinite variance when $1 < \nu \leqslant 2$, and undefined variance when $\nu \leqslant 1$. The Student's t distribution becomes the Gaussian distribution when $\nu = \infty$. Observing Fig. 1b, we can see that the Student's t distribution is heavy-tailed other than the Gaussian case. The PDF

of Student's t distribution is given by

$$\boldsymbol{f}(t) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\left(1 + \frac{t^2}{\nu}\right)^{\frac{-\nu+1}{2}}, \tag{2}$$

where Γ is the gamma function.

The Cauchy distribution is heavy-tailed with mean and variance undefined and is a special case of Student's t distribution with the $\nu = 1$ and of the SαS at $\alpha = 1$:

$$\boldsymbol{f}(t, \gamma) = \frac{1}{\pi\gamma\left[1 + \left(\frac{t-\mu_\alpha}{\gamma}\right)^2\right]}. \tag{3}$$

In Fig. 1c, it is clear that the Cauchy distribution is heavy-tailed when compared to the Gaussian distribution.

Finally, the Laplace distribution, or the double exponential distribution, is a special case of generalized Gaussian distribution and is characterised by the location parameter μ and the scale parameter b. The PDF of the Laplace distribution can be expressed as

$$\boldsymbol{f}(t, \mu, b) = \frac{1}{2b}\exp\left(-\frac{|t-\mu|}{b}\right). \tag{4}$$

Looking at Fig. 1d, we can see that the tail of the Laplace distribution gets heavier when b increases.

3. Derivation and complexity analysis of the adaptive G-sign algorithm

Let us consider a bandlimited graph signal $\boldsymbol{x_0} \subseteq \mathbb{R}^N$ and its noisy observation $\boldsymbol{y}[k] = \mathbf{D}_{\mathcal{S}}(\boldsymbol{x_0} + \boldsymbol{w}[k])$, where partial observations are modeled using a sampling matrix $\mathbf{D}_{\mathcal{S}}$, and k represents the kth discrete time step or iteration ranging from 0 to $k_{\max}$. $\boldsymbol{w}[k]$ is the noise which obeys the assumption below.

Assumption 1. The noise $\boldsymbol{w}[k]$ is a zero-mean (or zero-median when the mean is undefined) random variable and serves as additive noise that is i.i.d. among N different nodes and across different $k_{\max}$ time steps.

The computation $\mathbf{U\Sigma U}^T$ can be reduced by defining $\mathbf{U}_{\mathcal{F}} = \mathbf{U\Sigma}$ then dropping the all zeros columns, resulting in $\mathbf{U}_{\mathcal{F}}\mathbf{U}_{\mathcal{F}}^T\boldsymbol{x} = \mathbf{B}\boldsymbol{x}$ [17]. For a perfectly bandlimited graph signal $\boldsymbol{x_0}$ with frequency bands $\mathcal{F}$, $\boldsymbol{x_0} = \mathbf{B}\boldsymbol{x_0}$ [17]. In GLMS, the cost function is a convex optimization problem that aims to minimize the error between $\boldsymbol{y}[k]$ and

$\mathbf{D}_{\mathcal{S}}\mathbf{B}\hat{\mathbf{x}}[k]$, where $\hat{\mathbf{x}}[k]$ is the current step estimate of $\mathbf{x_0}$:

$$J\big(\hat{\mathbf{x}}[k]\big) = \mathbb{E}\left\|\mathbf{y}[k] - \mathbf{D}_{\mathcal{S}}\mathbf{B}\hat{\mathbf{x}}[k]\right\|_2^2. \tag{5}$$

In order to make one-step-ahead prediction, the spatial-domain update can be derived by stochastic gradient descent:

$$\hat{\mathbf{x}}[k+1] = \hat{\mathbf{x}}[k] + \mu_{lms}\mathbf{B}\mathbf{D}_{\mathcal{S}}(\mathbf{y}[\mathbf{k}] - \hat{\mathbf{x}}[\mathbf{k}]), \tag{6}$$

where μ_{lms} is the step size.

Even though the GLMS algorithm is simple, it is not time efficient and suffers from instability. The GLMP algorithm is an extension of the GLMS algorithm and has stable estimation performance compared to GLMS when estimating a graph signal under SαS noise but has additional computational complexity [18]. In classical adaptive filtering, the LMS algorithm is used extensively due to its simplicity of implementation, and the Sign-Error algorithm or the LMAD algorithm is an extension of the LMS algorithm to further increase the run-speed and decrease the algorithm complexity, with additional robustness gained by the l_1-norm cost function. To improve the time-efficiency and robustness of adaptive GSP algorithms under impulsive noise, we use a special case of the minimum dispersion criterion to form the cost function reducing it to a l_1-norm optimization problem as in the LMAD algorithm in one-dimensional adaptive filtering [31]:

$$J\big(\hat{\mathbf{x}}[k]\big) = \mathbb{E}\left\|\mathbf{y}[k] - \mathbf{D}_{\mathcal{S}}\mathbf{B}\hat{\mathbf{x}}[\mathbf{k}]\right\|_1^1. \tag{7}$$

l_1-norm error function provides a very plausible choice when the distribution is SαS, Cauchy, Laplace, or Student's t, with unknown density parameters [37]. The cost function (7) can be viewed as recovering the mean $\mathbf{x_0}$ from distribution $\mathbf{y}[k]$ and is LMAD sense optimal for SαS and Cauchy noise. Eq. (7) is also the optimal Maximum Likelihood Estimator for parameter estimation in Laplace distribution. Using the bandlimitedness property $\mathbf{B}\hat{\mathbf{x}}[k] = \hat{\mathbf{x}}[k]$, the update function of the G-Sign algorithm is obtained by stochastic gradient as in (6)

$$\begin{aligned}\hat{\mathbf{x}}[k+1] &= \hat{\mathbf{x}}[k] - \mu_s\frac{\partial \left\|\mathbf{y}[k] - \mathbf{D}_{\mathcal{S}}\mathbf{B}\hat{\mathbf{x}}[\mathbf{k}]\right\|_1^1}{\partial \hat{\mathbf{x}}[k]} \\ &= \hat{\mathbf{x}}[k] + \mu_s\mathbf{B}\mathbf{D}_{\mathcal{S}}\mathrm{Sign}\big(\mathbf{D}_{\mathcal{S}}\big(\mathbf{y}[k] - \hat{\mathbf{x}}[k]\big)\big).\end{aligned} \tag{8}$$

The Sign(.) function in the update equation results from taking the derivative of the l_1-norm cost function with the consideration of the point of discontinuity of the derivative at 0. This makes the update function resemble the form seen in the weight update of classical LMAD or Sign-Error algorithm [16,31]. A step size parameter μ_s is added by following classical adaptive filtering convention to control the amount of update. The ith 0 in the diagonal of $\mathbf{D}_{\mathcal{S}}$ corresponds to a 0 in the ith element of $\mathrm{sign}(\mathbf{D}_{\mathcal{S}}\big(\mathbf{y}[k] - \hat{\mathbf{x}}[k]\big))$. So, we can safely refactor (8) into

$$\hat{\mathbf{x}}[k+1] = \hat{\mathbf{x}}[k] + \mu_s\mathbf{B}\mathrm{Sign}\big(\mathbf{D}_{\mathcal{S}}\big(\mathbf{y}[k] - \hat{\mathbf{x}}[k]\big)\big). \tag{9}$$

This fixed amount of update from the minimum dispersion criterion is unaffected by impulsive noise because Sign($\cdot$) is only defined between -1 and 1, which clips the amplitude of extreme noise outliers [31,38]. In the GRLS algorithm, the update contains the covariance matrix of the noise [17,21]. In the GLMP algorithm, the exponent p of the update term is determined based on α of the SαS noise [18]. Unlike the algorithms that select the parameters using prior information from noise statistics, the G-Sign algorithm requires no prior information to determine the only parameter μ_s, which is in correspondence with the classical LMAD or the Sign-Error algorithm [34].

Eq. (9) reduces the number of operations by $2(N - |\mathcal{S}|)$ because the zeros in $\mathbf{D}_{\mathcal{S}}$ make $\mathrm{Sign}\big(\mathbf{D}_{\mathcal{S}}\big(\mathbf{y}[k] - \hat{\mathbf{x}}[k]\big)\big)$ sparse. The Sign($\cdot$) operation essentially compares the non-zero elements in $\mathcal{S}$, where in the worst case $\mathbf{y}[k] = \hat{\mathbf{x}}[k]$ it compares all the digits of $\hat{\mathbf{x}}[k]$ and $\mathbf{y}[k]$, without performing and algebraic operation. Since $\hat{\mathbf{x}}[k]$ is a noisy

Table 1
Computational complexity analysis.

	GLMS	GLMP	G-Sign
Addition	$N^2 + N$	$N^2 + 2N$	$N^2 + |S|$
Multiplication	$N^2 + 2N$	$N^2 + 3N$	$N^2 + N + |S|$
pth power	0	N	0
Sign($\cdot$)	0	N	$|S|$

estimation of an observation $\mathbf{y}[k]$, this worst case $\hat{\mathbf{x}}[k] = \mathbf{y}[k]$ is unlikely to happen. The comparison of the computational complexity of our G-Sign algorithm to GLMS and GLMP algorithm is in Table 1.

4. Convergence analysis under steady-State estimation

To estimate the steady-state performance of the G-Sign algorithm, we provide two analyses of the update function (9) using the Mean Squared Deviations (MSD) and the Mean Absolute Deviations (MAD). Based on a second-order analysis of the G-Sign algorithm, the MSD at step k is calculated by:

$$\mathrm{MSD}[k] = \mathbb{E}\left\|\hat{\mathbf{x}}[k] - \mathbf{x_0}\right\|^2. \tag{10}$$

It is worth mentioning that the GLMS algorithm in [8] and the GLMP algorithm in [18] both use the same MSD metric shown in (10). The MAD at step k is used for first-order analysis and is defined as

$$\mathrm{MAD}[k] = \mathbb{E}\left|\hat{\mathbf{x}}[k] - \mathbf{x_0}\right|. \tag{11}$$

Prior to the analyses, we first make two assumptions for the conditions to conduct the MSD analysis and the MAD analysis.

Assumption 2. The MSD analysis of the G-Sign algorithm is obtainable when the noise $\mathbf{w}[k]$ has finite variance.

In Section 4.1 a second-order analysis is conducted based on the update function (9) when Assumption 2 holds true. In assumption 1, we assumed that $\mathbf{w}[k]$ is additive, so this means that the second-order analysis of (9) might be infinite without further constraint on the second order statistics of $\mathbf{w}[k]$. In the context of this paper, the MSD analysis is for the cases where the noise is described by impulsive distributions which possess finite variance, such as generalized Gaussian distribution or some cases of Student's t distribution. Note that the Sign($\cdot$) function will clip the actual estimation output $\hat{\mathbf{x}}[k+1]$ in (9), so a MSD analysis is possible on the actual output when executing the algorithm.

Assumption 3. The MAD analysis is obtainable when the difference $\hat{\mathbf{x}}[k] - \mathbf{x_0}$ can be calculated.

Looking at the MAD definition in (11), it is clear that the MAD analysis of (9) is related to the first-order statistics of $\mathbf{w}[k]$. In assumption 1, we assumed that $\mathbf{w}[k]$ has finite first-order statistics; this means that we can always conduct the MAD analysis for the G-Sign algorithm. In the context of this paper, we provide the MAD analysis for those noise distributions that possess infinite variance.

4.1. Mean-squared deviations stability analysis under steady-state estimation

Let the error of estimating $\mathbf{x_0}$ at step k be $\tilde{\mathbf{x}}[k] = \hat{\mathbf{x}}[k] - \mathbf{x_0}$, then the error of the update (9) is

$$\tilde{\mathbf{x}}[k+1] = \tilde{\mathbf{x}}[k] + \mu_s\mathbf{B}\mathrm{Sign}(\mathbf{D}_{\mathcal{S}}(\mathbf{w}[k] - \tilde{\mathbf{x}}[k])). \tag{12}$$

The term $\mathbf{w}[k] - \tilde{\mathbf{x}}[k]$ is unlikely to be zero due to the estimation inaccuracy and the noise, then $\mathrm{Sign}(\mathbf{w}[k] - \tilde{\mathbf{x}}[k]) = \mathbf{h}(\mathbf{w}[k] - \tilde{\mathbf{x}}[k]) = (\mathbf{w}[k] - \tilde{\mathbf{x}}[k])/|\mathbf{w}[k] - \tilde{\mathbf{x}}[k]|$ for $\mathbf{w}[k] - \tilde{\mathbf{x}}[k] \neq 0$; here the division

is the element-wise division. For spectral analysis, (12) is transformed to the spectral domain using GFT by multiplying $\mathbf{U}_{\mathcal{F}}^T$ on both sides of (12):

$$\tilde{\mathbf{s}}[k+1] = \tilde{\mathbf{s}}[k] + \mu_s \mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{h}(\mathbf{w}[k] - \tilde{\mathbf{x}}[k]), \tag{13}$$

where $\tilde{\mathbf{s}}[k]$ is the GFT of $\tilde{\mathbf{x}}[k]$.

The error or the deviation from the ground-truth value in the mean-square sense can be obtained by taking the expectation of l_2-norm squared (13), leading to

$$\mathbb{E}\|\tilde{\mathbf{s}}[k+1]\|^2 = \mathbb{E}\|\tilde{\mathbf{s}}[k]\|^2 + \mathbb{E}\|\mu_s \mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{h}(\mathbf{w}[k] - \tilde{\mathbf{x}}[k])\|^2. \tag{14}$$

Considering that $\mathbf{w}[k] \gg \tilde{\mathbf{x}}[k]$ when k is large, we use the first-order Taylor series approximation of $\mathbf{h}(\mathbf{w}[k] - \tilde{\mathbf{x}}[k])$ around $\tilde{\mathbf{x}}[k] = 0$ to split the RHS of (14) into the error caused by noise and error caused by algorithm estimation, resulting in $\mathbf{h}(\mathbf{w}[k] - \tilde{\mathbf{x}}[k]) \approx \mathbf{h}(\mathbf{w}[k]) - \tilde{\mathbf{x}}[k]\mathbf{h}'(\mathbf{w}[k])$ which can be seen as a sigmoid function approximation to the sign algorithm which makes it differentiable. Using the inverse GFT, $\tilde{\mathbf{x}}[k]$ can be written as $\mathbf{U}_{\mathcal{F}}\tilde{\mathbf{s}}[k]$ and (14) becomes

$$\begin{aligned}\mathbb{E}\|\tilde{\mathbf{s}}[k+1]\|^2 &= \mathbb{E}\|\tilde{\mathbf{s}}[k]\|^2 + \mathbb{E}\|\mu_s \mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S\big(\mathbf{h}(\mathbf{w}[k]) - \mathbf{U}_{\mathcal{F}}\tilde{\mathbf{s}}[k]\mathbf{h}'(\mathbf{w}[k])\big)\|^2 \\ &= \mathbb{E}\|\tilde{\mathbf{s}}[k]\|_{\mathbf{\Phi}}^2 + \mu_s^2 \mathbb{E}\|\mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{R}_w \mathbf{w}[k]\|^2,\end{aligned} \tag{15}$$

where $\|\tilde{\mathbf{s}}[k]\|_{\mathbf{\Phi}}^2$ is the weighted Euclidean norm $\tilde{\mathbf{s}}^T[k]\mathbf{\Phi}\tilde{\mathbf{s}}[k]$ and $\mathbf{\Phi}$ can be expressed as

$$\mathbf{\Phi} = \big(\mathbf{I} - \mu_s \mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{R}_p \mathbf{U}_{\mathcal{F}}\big)^T \big(\mathbf{I} - \mu_s \mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{R}_p \mathbf{U}_{\mathcal{F}}\big). \tag{16}$$

The expectation of the element-wise division $1/|\mathbf{w}[k]|$ in $\mathbf{h}(\mathbf{w}[k])$ can be approximated using the FLOM $\mathbb{E}|\mathbf{w}[k]|^{-p}$ as seen in the GLMP algorithm [18], where p is modified from 1 to 0.99 which gives us $\mathbf{R}_p = (1-p)\mathbb{E}|\mathbf{w}[k]|^{-p}\mathbf{I}$, and $\mathbf{R}_w = \mathbb{E}|\mathbf{w}[k]|^{-p}\mathbf{I}$. Equation(15) can be factorized using the trace trick $\mathbb{E}\{\mathbf{X}^T\mathbf{Y}\mathbf{X}\} = \mathrm{Tr}\,(\mathbb{E}\{\mathbf{X}\mathbf{X}^T\mathbf{Y}\})$:

$$\mathbb{E}\|\tilde{\mathbf{s}}[k+1]\|^2 = \mathbb{E}\|\tilde{\mathbf{s}}[k]\|_{\mathbf{\Phi}}^2 + \mu_s^2 \mathrm{Tr}\big(\mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{C} \mathbf{D}_S \mathbf{U}_{\mathcal{F}}\big), \tag{17}$$

where $\mathbf{C} = \mathbb{E}\|\mathbf{R}_w \mathbf{w}[k]\|^2$ is the covariance matrix of $\mathbf{R}_w \mathbf{w}[k]$ and has a similar structure to the partial correlation matrix or the normalized covariance matrix of $\mathbf{w}[k]$. In Assumption 1 we assumed that the noises across different nodes are i.i.d., so the correlation of the noise between two different nodes is zero, leading to $\mathbf{C} = \mathbf{I}$. Combining with the idempotent and self-adjoint property of the sampling matrix $\mathbf{D}_S$, (17) can be simplified to

$$\mathbb{E}\|\tilde{\mathbf{s}}[k+1]\|^2 = \mathbb{E}\|\tilde{\mathbf{s}}[k]\|_{\mathbf{\Phi}}^2 + \mu_s^2 \mathrm{Tr}\big(\mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{U}_{\mathcal{F}}\big). \tag{18}$$

Using the properties $\mathrm{Tr}\{\mathbf{Y}\mathbf{X}\} = \mathrm{vec}(\mathbf{X}^T)^T \mathrm{vec}(\mathbf{Y})$ and $\mathrm{vec}(\mathbf{X}\mathbf{Y}\mathbf{Z}) = (\mathbf{Z}^T \otimes \mathbf{X})\mathrm{vec}(\mathbf{Y})$, (18) can be written into a recursive relationship starting from time step 0:

$$\begin{aligned}\mathbb{E}\|\tilde{\mathbf{s}}[k+1]\|^2 &= \mathbb{E}\|\tilde{\mathbf{s}}[0]\|_{\mathbf{\Phi^k}}^2 + \mu_s^2 \sum_{i=0}^{k} \mathrm{Tr}\big(\Phi^i \mathbf{G}\big) \\ &= \mathbb{E}\|\tilde{\mathbf{s}}[0]\|_{\mathbf{\Phi^k}}^2 + \mu^2 \mathrm{vec}(\mathbf{G})^T \sum_{i=0}^{k} \mathbf{Q}^i \mathrm{vec}(\mathbf{I}),\end{aligned} \tag{19}$$

where $\tilde{\mathbf{s}}[0]$ is the error at $k=0$, $\mathbf{G} = \mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{U}_{\mathcal{F}}$, and $\mathbf{Q} = (\mathbf{I} - \mu_s \mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{R}_p \mathbf{U}_{\mathcal{F}})^T \otimes (\mathbf{I} - \mu_s \mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{R}_p \mathbf{U}_{\mathcal{F}})$. From (19), we see that $\mathbb{E}\|\tilde{\mathbf{s}}[k+1]\|^2$ converges to a steady value if the RHS of (19) converges.

Considering (16), when the initial error $\tilde{\mathbf{s}}[0]$ is bounded, we demand that the weighted Euclidean norm $\mathbb{E}\|\tilde{\mathbf{s}}[0]\|_{\mathbf{\Phi^k}}^2$ converges to zero at large k, requiring $\|(\mathbf{I} - \mu_s \mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{R}_p \mathbf{U}_{\mathcal{F}})\| < 1$. Following this condition, the summation term of the RHS of (19) will be a geometric series that converges to a constant value. For a diagonalizable matrix $\mathbf{X}$ of size $N \times N$ and a vector $\mathbf{z}$ of size N, the property $\|\mathbf{X}\mathbf{z}\|^2 = \sum \lambda_i z_i$ results in the inequality $\|\mathbf{X}\| \leq |\lambda_{\max}|$. Here $\{\lambda_i, i = 1 \ldots N\}$ are the eigenvalues of $\mathbf{X}$ and $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{X}$. Let $\mathbf{X} = \mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{R}_p \mathbf{U}_{\mathcal{F}}$ and use the above property; satisfying the condition $\|(\mathbf{I} - \mu_s \mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{R}_p \mathbf{U}_{\mathcal{F}})\| < 1$ is equivalent to

satisfying the condition $0 < |1 - \mu_s \lambda_{\max}| < 1$. Since the parameter μ_s is the step size, it should always have a positive value. Knowing that $\lambda_{\max}$ is the maximum eigenvalue of $\mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{R}_p \mathbf{U}_{\mathcal{F}}$, $\lambda_{\max}$ is positive because the FLOM calculated in $\mathbf{R}_p$ is positive. As a result, the error of the G-Sign algorithm will converge to a steady value by choosing the only free parameter μ_s to satisfy the inequality $0 < |1 - \mu_s \lambda_{\max}| < 1$, leading to the condition

$$0 < \mu_s < \frac{2}{\lambda_{\max}}. \tag{20}$$

Even though the condition (20) has a structure similar to the convergence condition seen in GLMS, the bound $2/\lambda_{\max}$ is obtained using a different expression than the one for GLMS. This bound is derived based on the $\mathrm{Sign}(\cdot)$ update of (9) that originated from the non-Gaussian noise assumption and the l_1-norm optimization problem (7). Under condition (20), the G-Sign algorithm has a stable MSD behavior.

As $k \to \infty$ and μ_s satisfies (20), the difference between $\tilde{\mathbf{s}}[k]$ and $\tilde{\mathbf{s}}[k]$: $\lim_{k \to \infty} \tilde{\mathbf{s}}[k] = \lim_{k \to \infty} \tilde{\mathbf{s}}[k+1]$. Using the energy conservation property of GFT seen in the GLMS in [8], which is a direct extension from the energy conservation of the classical LMS proven in [39], the relationship between the spatial domain MSD and the spectral domain MSD is $\lim_{k \to \infty} \mathbb{E}\|\tilde{\mathbf{x}}[k]\|^2 = \lim_{k \to \infty} \mathbb{E}\|\tilde{\mathbf{s}}[k]\|^2$. In GFT, this energy preserving property (a special case of Parseval's relation), can be easily proven using the property of orthonormal basis $\mathbf{U}^T = \mathbf{U}^{-1}$ we have in Section 2, where $\|\tilde{\mathbf{s}}[k]\|^2 = (\mathbf{U}^T \tilde{\mathbf{x}}[k])^T \mathbf{U}^T \tilde{\mathbf{x}}[k] = \|\tilde{\mathbf{x}}[k]\|^2$ [11]. Revisiting (19), after selecting a μ_s that satisfies (20), the theoretical steady-state MSD can be calculated:

$$\begin{aligned}\mathrm{MSD}[k] &= \lim_{k \to \infty} \mathbb{E}\|\tilde{\mathbf{x}}[k]\|^2 = \lim_{k \to \infty} \mathbb{E}\|\tilde{\mathbf{s}}[k]\|^2 \\ &= \mu_s^2 \mathrm{vec}(\mathbf{G})^T (\mathbf{I} - \mathbf{Q})^{-1} \mathrm{vec}(\mathbf{I}).\end{aligned} \tag{21}$$

4.2. Mean-Absolute deviations stability analysis under steady-state estimation

The MSD analysis of Section 4.1 may not be applicable for certain noise statistics considered in the paper; we also derive a MAD analysis for the G-Sign algorithm under steady-state estimation. To cope with the spectral domain analysis in Section 4.1, we will conduct a first-order statistics analysis using the spectral-domain MAD based on (13); it can be factored into the following error update equation:

$$\begin{aligned}\tilde{\mathbf{s}}[k+1] &= \big(\mathbf{I} - \mu_s \mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{R}_p \mathbf{U}_{\mathcal{F}}\big)\tilde{\mathbf{s}}[k] + \mu_s \mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{R}_w \mathbf{w}[k-1] \\ &= \mathbf{\Phi}_1^k \tilde{\mathbf{s}}[0] + \mu_s \sum_{i=0}^{k} \mathbf{\Phi}_1^k \mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{R}_w \mathbf{w}[k-i],\end{aligned} \tag{22}$$

where $\mathbf{\Phi}_1 = (\mathbf{I} - \mu_s \mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{R}_p \mathbf{U}_{\mathcal{F}})$. By taking the limit of the expected absolute value of (22) as $k \to \infty$ and using the same approximation for $\mathbf{R}_p$ and $\mathbf{R}_w$ in Section 4.1, the spectral domain absolute error update can be expressed as

$$\lim_{k \to \infty} \mathbb{E}|\tilde{\mathbf{s}}[k]| = \lim_{k \to \infty} \mathbb{E}\left|\mathbf{\Phi}_1^k \tilde{\mathbf{s}}[0] + \mu \sum_{i=0}^{k} \mathbf{\Phi}_1^i \mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{1}\right|, \tag{23}$$

where $\mathbb{E}\{\mathbf{R}_w \mathbf{w}[k-1]\} \approx \mathbf{1}$ for $p = .99$ and $\mathbf{1}$ is an all-ones vector of size $N \times 1$. For the RHS of (23) to converge, it requires that $\|(\mathbf{I} - \mu_s \mathbf{U}_{\mathcal{F}}^T \mathbf{D}_S \mathbf{R}_p \mathbf{U}_{\mathcal{F}})\| < 1$, so that the summation becomes a geometric series and $\mathbf{\Phi}_1^k \tilde{\mathbf{s}}[0]$ converges to zero for a bounded $\tilde{\mathbf{s}}[0]$. This condition is the same as the MSD case in Section 4.1, which will lead to condition (20).

5. Experimental results

We would like to test the performance of the G-Sign algorithm in estimating graph signals under impulsive noise. Steady-state experiments are conducted in Sections 5.1–5.3 using the random sensor graph generated by Python package PyGSP shown in Fig. 2 with
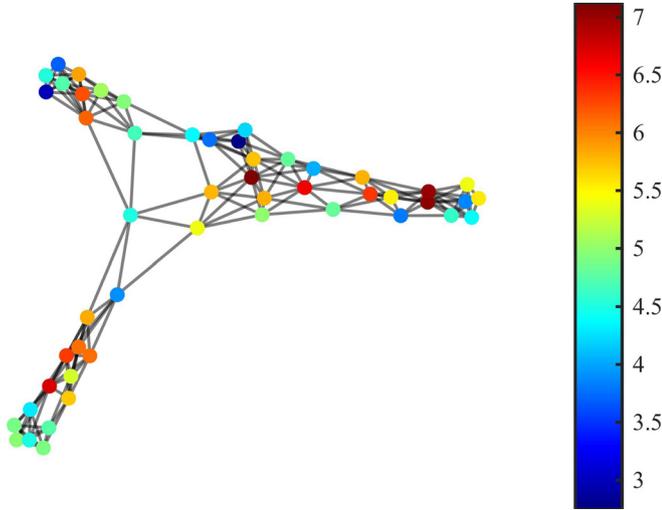
Fig. 2. The graph signal of a sensor network and its topology.
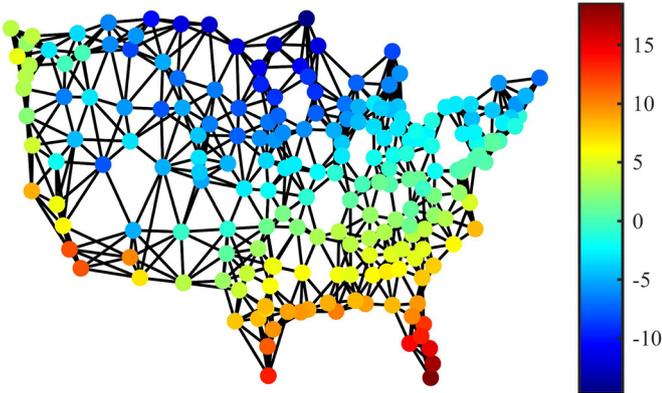
Fig. 3. The first time instance of a real time-varying graph signal and its graph topology.

N = 50, bandlimited frequencies $|\mathcal{F}| = 20$, and greedy sampling strategy in [6] with $|\mathcal{S}| = 30$. A real time-varying graph signal with the topology shown in Fig. 3 is estimated in Section 5.4. The graph signal in Fig. 3 represents hourly temperature recorded across the U.S. at 205 different locations [40]. We use geography-based graph generation with 8 nearest neighbors seen in [6] to form the topology shown in Fig. 3 with $N = 205$. In the experiment, the sampling technique is the same greedy strategy as in [6] with $|\mathcal{S}| = 130$ and $|\mathcal{F}| = 125$. All the experiments are averaged over 100 independent runs. The experiments were conducted in MATLAB 2020b on a computer with AMD Ryzen 5 3600 CPU and 32GB of RAM.

5.1. The effect of the step size parameter

In the G-Sign algorithm the step size μ_s is the only user-defined parameter. We would like to observe the effect of changing the value of μ_s. The experiment is conducted using the graph signal shown in Fig. 2, and the graph signal is corrupted by SαS noise with $\alpha = 1.06$ and $\gamma = 0.1$. We tested four different values of μ_s and run the G-sign algorithm for $k_{max} = 3000$ iterations. The performance of the G-Sign algorithm is measured comparing the output graph signal estimation with the clean bandlimited graph signal using the MSD defined in (10) and the MAD defined in (11).

Both the MAD and the MSD of this experiment are shown in Fig. 4. We can see from Fig. 4 that as μ_s decreases, the G-Sign algorithm will get more accurate but will also require more iterations to converge to a steady value. It can also be confirmed that

Table 2
Run time comparison of steady-state experiments.

	SαS	Cauchy	Student's t	Laplace
GLMS	0.0307(s)	0.0289(s)	0.0313(s)	0.0292(s)
GLMP	0.0462(s)	0.0468(s)	0.0340(s)	0.0339(s)
G-Sign	**0.0055(s)**	**0.0062(s)**	**0.0059(s)**	**0.0058(s)**

the MAD and the MSD performance of the G-Sign algorithm shares the same behavior.

5.2. Steady-state MSD under non-Gaussian noises

The G-Sign algorithm is compared with the GLMS algorithm and the GLMP algorithm for estimating a partially observed steady-state graph signal under SαS, Cauchy, Student's t, and Laplace noises. The aim is to compare the stability of estimation, the iterations until convergence, and the run-time. In order to fairly compare the algorithms under each noise scenario, the step sizes are tuned so the algorithms behave similarly in MSD when there is a stable estimation. Notice that SαS becomes Cauchy when $\alpha = 1$, and the p parameter of the GLMP algorithm is defined only for $1 < p < 2$ with $p < \alpha$, so we test the GLMP algorithm under a near Cauchy SαS noise with $\alpha = 1.06$ with $p = \alpha - 0.05$ (as suggested in [18]) for the Cauchy experiment. For the experiment under SαS noise, $p = \alpha - 0.05$ for the GLMP algorithm, and at other noise distributions we set $p = 1.5$ for the GLMP algorithm. The MSD of the experiments is in Fig. 5 with the theoretical MSD calculated using (21). The run-time of running $k_{max} = 2400$ iterations of each algorithm for different experiments is in Table 2.

From Fig. 5, we can see that the GLMS algorithm is unstable when estimating the graph signal under SαS, Cauchy, and Student's t noise. This instability is introduced by the heavy tail behavior of the noises [37]. In Fig. 5d, the GLMS algorithm is stable under Laplace noise but requires about 5 times the run-time that the G-Sign algorithm needs to converge.

The G-Sign algorithm behaves similarly to the GLMP algorithm under Cauchy/near Cauchy noise in Fig. 5b because both algorithms are derived based on the minimum dispersion criterion. Under SαS noise, even though the iterations to reach the same steady-state MSD for G-Sign algorithm and GLMP algorithm in Fig. 5b are 1200 iterations and 650 iterations respectively, when looking at Table 2 we see that the G-Sign algorithm only takes 1/8 the time that the GLMP algorithm needs to complete the experiment. The MSD performance of the G-Sign algorithm matches the theoretical results under the four impulsive noises. In Table 2, we see that the G-Sign algorithm has the fastest run-time under all scenarios, which is in correspondence with the analysis in Table 1 that the G-Sign algorithm has the lowest computational complexity. Combining with Fig. 5, we conclude that for steady-state graph signal estimation with missing node values under non-Gaussian noise, the proposed G-Sign algorithm is able to make a stable estimation and faster run-time compared to the GLMS algorithm and the GLMP algorithm.

5.3. Steady-State MAD under non-Gaussian noises

In this section, we would like to observe the MAD performance of the G-Sign algorithm under SαS noise with $\alpha = 1.5$ and Cauchy noise due to the fact that these two noise settings in Section 5.2 do not have a finite mean-squared statistics. The MAD results of these two noises are shown in Fig. 6, which are calculated simultaneously along with the MSD in Fig. 5a and b. From Fig. 6, we can observe that the MAD performance of the G-Sign algorithm under SαS and Cauchy noises are stable, while the GLMS does not converge.

(a) MAD

(b) MSD
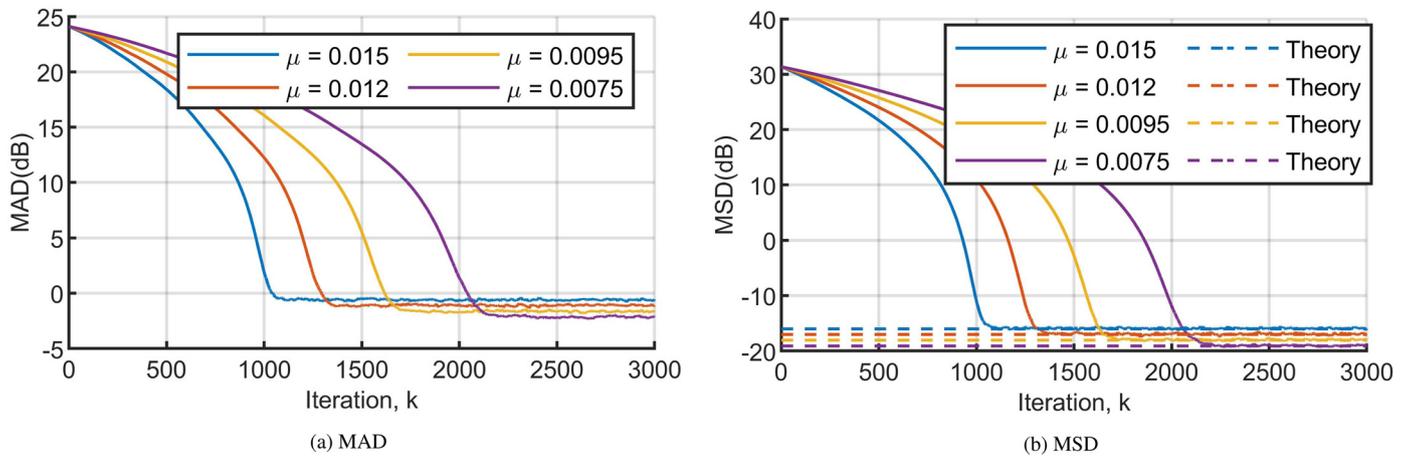
Fig. 4. MAD and MSD performances for estimating a steady-state graph signal under SαS noise using different μ_s.

(a) SαS noise with $\alpha = 1.5$ and $\gamma = 0.1$

(b) Cauchy with $\mu = 0$ and $\gamma = 0.1$ (near Cauchy SαS when GLMP)

(c) Student's t with $\nu = 3$

(d) Laplace with $\mu = 0$ and $b = \sqrt{2}$
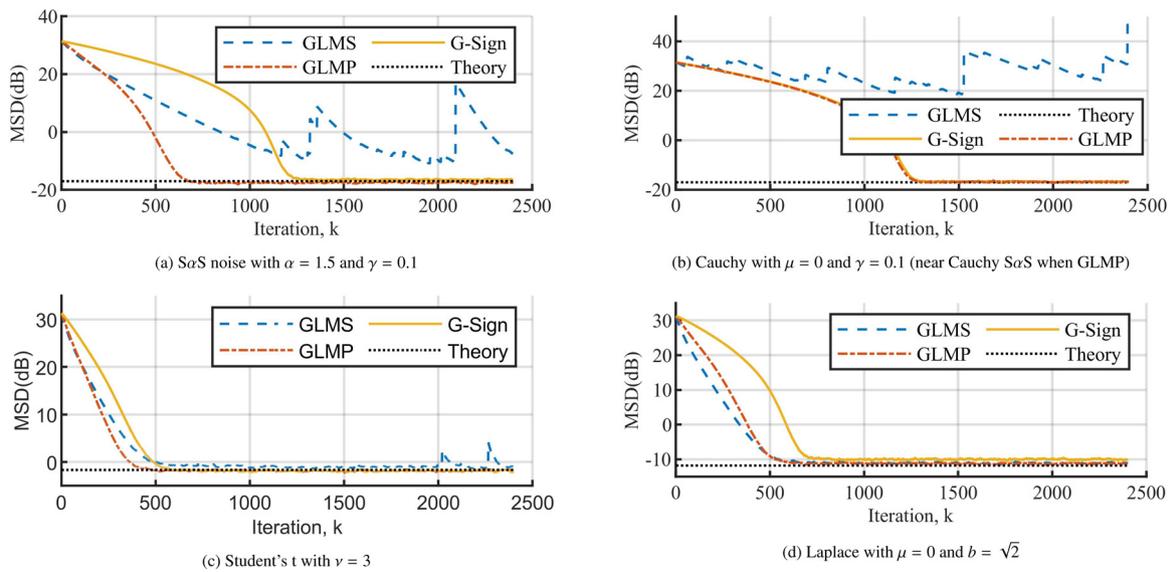
Fig. 5. theoretical and experimental MSDs for steady-state graph signal estimates under different noises.

(a) SαS noise with $\alpha = 1.5$ and $\gamma = 0.1$

(b) Cauchy with $\mu = 0$ and $\gamma = 0.1$ (near Cauchy SαS when GLMP)
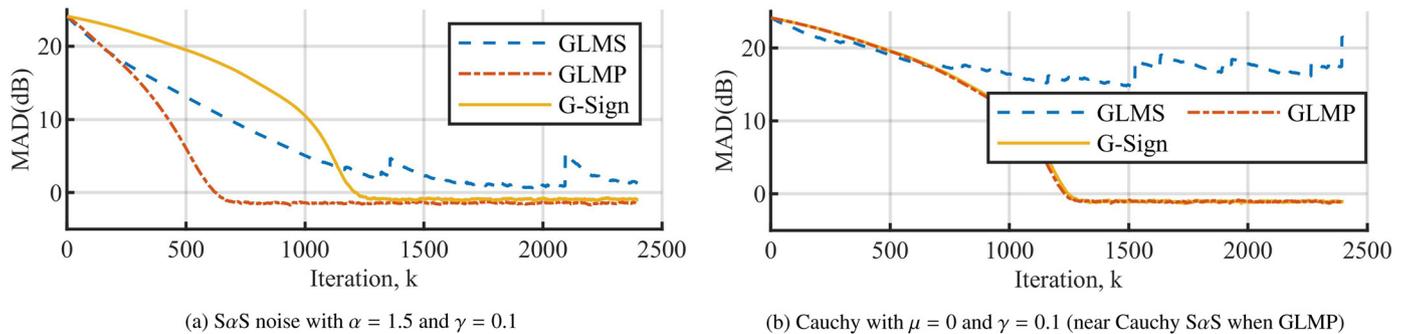
Fig. 6. MAD performances for estimating a steady-state graph signal under different noises.

5.4. Time-varying estimation under impulsive noise

In this section, the G-Sign algorithm is used to estimate a time-varying graph signal corrupted by noise modeled by SαS, Cauchy, Student's t, and Laplace distributions. The G-Sign algorithm is compared to the GLMP and GLMS algorithms. The duration of this time-varying graph signal is 95 hours, making $k_{\max} = 95$. At each iteration each adaptive GSP algorithms outputs an online estimate

of the graph signal. Again, for the Cauchy experiment we run the GLMP algorithm under a near Cauchy SαS noise with $\alpha = 1.06$ with $p = \alpha - 0.05$. An illustration of one time step of the graph signal is shown in Fig. 3. To make a fair comparison, the step sizes are $\mu_s = 1.5$ for all the algorithms.

Fig. 7 illustrates the estimation of one selected node with a time-varying graph signal. Notice that the GLMS algorithm is again unstable under SαS, Cauchy, and Student's t noises, whereas the G–
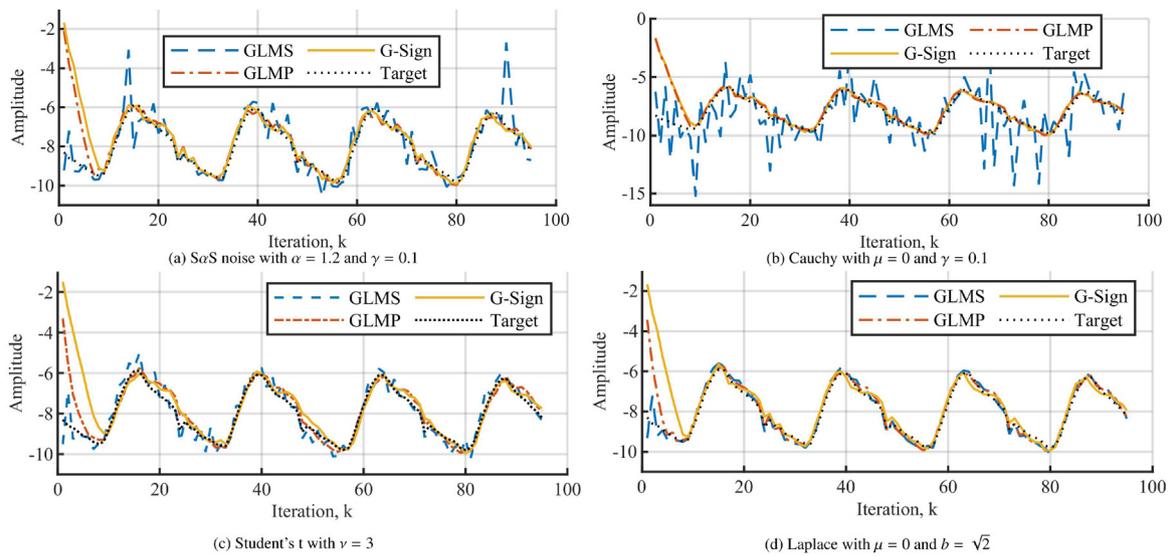
Fig. 7. Estimation for one selected node from a time-varying graph signal under different noises.

Table 3
Run time comparison of time-varing experiments.

	SαS	Cauchy	Student's t	Laplace
GLMS	0.0173(s)	0.0174(s)	0.0169(s)	0.0170(s)
GLMP	0.0196(s)	0.0198(s)	0.0192(s)	0.0189(s)
G-Sign	**0.0028(s)**	**0.0026(s)**	**0.0028(s)**	**0.0027(s)**

Sign algorithm is not influenced by any of these impulsive noises. The run-times for GLMS, GLMP, and G-Sign algorithms to finish this experiment are shown in Table 3. From Table 3, we can see that the G-Sign algorithm remains the fastest among all compared algorithms under the time-varying setting. From Fig. 7 and the run-time comparisons, we conclude that the G-Sign algorithm is able to track a time-varying graph signal under non-Gaussian noise in a time-efficient manner.

6. Conclusion

In this paper, we proposed the G-Sign algorithm for online estimation of partially observed steady-state and time-varying graph signals under impulsive noise. The G-Sign algorithm is derived using the minimum dispersion criterion which is stable and robust under impulsive noise. Experimental results confirm that the G-Sign algorithm is of low complexity, time-efficient and robust.

Declaration of Competing Interest

None.

CRediT authorship contribution statement

Yi Yan: Software, Validation, Data curation, Formal analysis, Writing – original draft, Visualization. **Ercan E. Kuruoglu:** Conceptualization, Methodology, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Mustafa A. Altinkaya:** Conceptualization, Methodology, Writing – review & editing.

Acknowledgements

References

[1] A. Sandryhaila, J.M. Moura, Big data analysis with signal processing on graphs: representation and processing of massive data sets with irregular structure, IEEE Signal Process. Mag. 31 (5) (2014) 80–90, doi:10.1109/MSP.2014.2329213.
[2] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, 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. 30 (2013) 83–98, doi:10.1109/MSP.2012.2235192.
[3] A. Ortega, P. Frossard, J. Kovaevi, J.M.F. Moura, P. Vandergheynst, Graph signal processing: overview, challenges, and applications, Proc. IEEE 106 (5) (2018) 808–828, doi:10.1109/JPROC.2018.2820126.
[4] X. Dong, D. Thanou, L. Toni, M.M. Bronstein, P. Frossard, Graph signal processing for machine learning: areview and new perspectives, IEEE Signal Process. Mag. 37 (2020) 117–127, doi:10.1109/MSP.2020.3014591.
[5] W. Huang, T.A.W. Bolton, J.D. Medaglia, D.S. Bassett, A. Ribeiro, D. Van De Ville, A graph signal processing perspective on functional brain imaging, Proc. IEEE 106 (5) (2018) 868–885, doi:10.1109/JPROC.2018.2798928.
[6] M.J.M. Spelta, W.A. Martins, Normalized lms algorithm and data-selective strategies for adaptive graph signal estimation, Signal Processing 167 (2020) 107326, doi:10.1016/j.sigpro.2019.107326.
[7] D.M. Mohan, M.T. Asif, N. Mitrovic, J. Dauwels, P. Jaillet, Wavelets on graphs with application to transportation networks, IEEE ITSC (2014).
[8] P. Di Lorenzo, S. Barbarossa, P. Banelli, S. Sardelletti, Adaptive least mean squares estimation of graph signals, IEEE Trans. Signal Inf. Process. Netw. 2 (4) (2016) 555–568, doi:10.1109/TSIPN.2016.2613687.
[9] I. Jaboski, Graph signal processing in applications to sensor networks, smart grids, and smart cities, IEEE Sens. J. 17 (23) (2017) 7659–7666, doi:10.1109/JSEN.2017.2733767.
[10] W. Hu, J. Pang, X. Liu, D. Tian, C.-W. Lin, A. Vetro, Graph signal processing for geometric data and beyond: theory and applications, IEEE Trans. Multimedia (Early Access) (2021), doi:10.1109/TMM.2021.3111440.
[11] N. Tremblay, P. Gonalves, P. Borgnat, Chapter 11 - design of graph filters and filterbanks, in: P.M. Djuric, C. Richard (Eds.), Cooperative and Graph Signal Processing, Academic Press, 2018, pp. 299–324.
[12] D.K. Hammond, P. Vandergheynst, R. Gribonval, Wavelets on graphs via spectral graph theory, Appl. Comput. Harmon. Anal. 30 (2) (2011) 129–150.
[13] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, ICLR (2017).
[14] N. Tremblay, P. Borgnat, Graph wavelets for multiscale community mining, IEEE Trans. Signal Process. 62 (20) (2014) 5227–5239, doi:10.1109/TSP.2014.2345355.
[15] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, NeurIPS (2016).
[16] P. Diniz, Adaptive Filtering: Algorithms and Practical Implementation, Springer, 2008.
[17] P. Di Lorenzo, P. Banelli, E. Isufi, S. Barbarossa, G. Leus, Adaptive graph signal processing: algorithms and optimal sampling strategies, IEEE Trans. Signal Process. 66 (13) (2018) 3584–3598, doi:10.1109/TSP.2018.2835384.
[18] N.H. Nguyen, K. Dogancay, W. Wang, Adaptive estimation and sparse sampling for graph signals in alpha-stable noise, Digit. Signal Process. 105 (2020) 102782, doi:10.1016/j.dsp.2020.102782.
[19] P. Di Lorenzo, P. Banelli, S. Barbarossa, S. Sardellitti, Distributed adaptive learning of graph signals, IEEE Trans. Signal Process. 65 (16) (2017) 4193–4208, doi:10.1109/TSP.2017.2708035.

[20] F. Hua, R. Nassif, C. Richard, H. Wang, A.H. Sayed, Online distributed learning over graphs with multitask graph-filter models, IEEE Trans. Signal Inf. Process. Netw 6 (2020) 63–77, doi:10.1109/TSIPN.2020.2964214.

[21] P. Di Lorenzo, E. Isufi, P. Banelli, S. Barbarossa, G. Leus, Distributed recursive least squares strategies for adaptive reconstruction of graph signals, in: EU-SIPCO, 2017, pp. 2289–2293.

[22] V.R.M. Elias, V.C. Gogineni, W.A. Martins, S. Werner, Adaptive graph filters in reproducing kernel hilbert spaces: design and performance analysis, IEEE Trans. Signal Inf. Process. Netw. 7 (2021) 62–74, doi:10.1109/TSIPN.2020.3046217.

[23] E. Isufi, A. Loukas, A. Simonetto, G. Leus, Autoregressive moving average graph filtering, IEEE Trans. Signal Process. 65 (2) (2017) 274–288, doi:10.1109/TSP.2016.2614793.

[24] F. Grassi, A. Loukas, N. Perraudin, B. Ricaud, A time-vertex signal processing framework: scalable processing and meaningful representations for time-series on graphs, IEEE Trans. Signal Process. 66 (3) (2018) 817–829, doi:10.1109/TSP.2017.2775589.

[25] S. Banerjee, M. Agrawal, Underwater acoustic communication in the presence of heavy-tailed impulsive noise with bi-parameter Cauchy-Gaussian mixture model, SYMPOL (2013) 1–7, doi:10.1109/SYMPOL.2013.6701903.

[26] Q. Li, Y. Ben, S.M. Naqvi, J.A. Neasham, J.A. Chambers, Robust students t based cooperative navigation for autonomous underwater vehicles, IEEE Trans. Instrum. Meas. 67 (8) (2018) 1762–1777, doi:10.1109/TIM.2018.2809139.

[27] O. Karakus, E. Kuruoglu, M. Altinkaya, Modelling impulsive noise in indoor powerline communication systems, Signal Image Video Process. 14 (2020) 1655–1661.

[28] I. Lee, X. Liu, C. Zhou, B. Kosko, Noise-enhanced detection of subthreshold signals with carbon nanotubes, IEEE Trans. Nanotechnol. 5 (6) (2006) 613–627, doi:10.1109/TNANO.2006.883476.

[29] N. Wang, D.-Y. Yeung, Bayesian robust matrix factorization for image and video processing, in: 2013 IEEE International Conference on Computer Vision, 2013, pp. 1785–1792.

[30] E.E. Kuruoglu, P.J. Rayner, W.J. Fitzgerald, Least lp-norm impulsive noise cancellation with polynomial filters, Signal Process. 69 (1) (1998) 1–14, doi:10.1016/S0165-1684(98)00083-8.

[31] M. Shao, C. Nikias, Signal processing with fractional lower order moments: stable processes and their applications, Proc. IEEE 81 (7) (1993) 986–1010.

[32] B. Chen, L. Xing, J. Liang, N. Zheng, J.C. Prncipe, Steady-state mean-square error analysis for adaptive filtering under the maximum correntropy criterion, IEEE Signal Process. Lett. 21 (7) (2014) 880–884, doi:10.1109/LSP.2014.2319308.

[33] V.C. Gogineni, S.P. Talebi, S. Werner, D.P. Mandic, Fractional-order correntropy adaptive filters for distributed processing of α-stable signals, IEEE Signal Process. Lett. 27 (2020) 1884–1888, doi:10.1109/LSP.2020.3029702.

[34] L. Wang, The L1 penalized lad estimator for high dimensional linear regression, J. Multivar. Anal. 120 (2013) 135–151, doi:10.1016/j.jmva.2013.04.001.

[35] A. Anis, A. Gadde, A. Ortega, Efficient sampling set selection for bandlimited graph signals using graph spectral proxies, IEEE Trans. Signal Process. 64 (14) (2016) 3775–3789, doi:10.1109/TSP.2016.2546233.

[36] N.H. Nguyen, K. Dogancay, E.E. Kuruoglu, An iteratively reweighted instrumental-variable estimator for robust 3-D AOA localization in impulsive noise, IEEE Trans. Signal Process. 67 (18) (2019) 4795–4808, doi:10.1109/TSP.2019.2931210.

[37] Y. Chen, H.C. So, E.E. Kuruoglu, Variance analysis of unbiased least lp-norm estimator in non-Gaussian noise, Signal Process. 122 (2016) 190–203, doi:10.1016/j.sigpro.2015.12.003.

[38] V. Bhatia, B. Mulgrew, A.T. Georgiadis, Stochastic gradient algorithms for equalisation in α-stable noise, Signal Process. 86 (4) (2006) 835–845, doi:10.1016/j.sigpro.2005.06.013.

[39] A.H. Sayed, V.H. Nascimento, Energy Conservation and the Learning Ability of LMS Adaptive Filters, Least-Mean-Square Adaptive Filters, Wiley, 2003.

[40] A. Arguez, I. Durre, S. Applequist, M. Squires, R. Vose, X. Yin, R. Bilotta, NOAA's U.S. climate normals (1981–2010), NOAA National Centers for Environmental Information, 2010.