Miettinen, Jari; Vorobyov, Sergiy A.; Ollila, Esa

Modelling and studying the effect of graph errors in graph signal processing

Published in:
Signal Processing

DOI:
10.1016/j.sigpro.2021.108256

Published: 01/12/2021

Document Version
Publisher's PDF, also known as Version of record

Signal Processing 189 (2021) 108256

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Modelling and studying the effect of graph errors in graph signal processing

Jari Miettinen, Sergiy A. Vorobyov, Esa Ollila*

Dept. of Signal Processing and Acoustics, Aalto University, Otakaari 1, Espoo, FI-00076 Aalto, Finland

ARTICLE INFO

Article history:
Received 10 February 2021
Revised 20 May 2021
Accepted 20 July 2021
Available online 27 July 2021

Keywords:
Erdös-Rényi graphs
Error effect
Graph signal processing
Minimum distance index
Shift matrix

ABSTRACT

The first step for any graph signal processing (GSP) procedure is to learn the graph signal representation, i.e., to capture the dependence structure of the data into an adjacency matrix. Indeed, the adjacency matrix is typically not known a priori and has to be learned. However, it is learned with errors. A little attention has been paid to modelling such errors in the adjacency matrix, and studying their effects on GSP methods. However, modelling errors in the adjacency matrix will enable both to study the graph error effects in GSP and to develop robust GSP algorithms. In this paper, we therefore introduce practically justifiable graph error models. We also study, both analytically when possible and numerically, the graph error effect on the performance of GSP methods in different types of problems such as filtering of graph signals and independent component analysis of graph signals (graph decorrelation).

1. Introduction

In the classical signal processing setup where the digital signals are represented in terms of time series or vectors of spatial measurements (for example, measurements by sensor arrays), it is assumed that each point of a discrete signal depends on the preceding or spatially close point of the signal. When the signal is no longer adequately represented by a simple time series structure, the classical signal processing tools are no longer applicable. Thus, graph signal processing (GSP) [1–3] has emerged as a new area of signal processing and data analysis where one of the major aims is to generalize the standard signal processing methods and concepts into the context of more complex signals represented on graphs. The graph corresponding to ordered data is then just a simple special case of a digital signal graph. Graph signals and models are frequently used when studying sensor networks, brain networks, gene regulatory networks, and social networks to name just a few [4–7].

In GSP, each signal value is indexed by a node in a graph at which this signal value is measured, and edges between pairs of nodes in a graph indicate dependencies between the corresponding signal values. Then the underlying graph can be fully specified by an adjacency matrix, denoted hereafter as $\mathbf{A}$, whose (i, j)th element is nonzero if the ith and the jth nodes are connected, and the value $[\mathbf{A}]_{i,j} = a_{ij}$ describes the strength of the relationship. The adjacency matrix is the basis of GSP on which the graph Fourier transform (GFT) and graph filters [1] are built, often via the graph Laplacian matrix or other *shift operators* which are based on the adjacency matrix.[1]

The process of obtaining the adjacency matrix may take different forms. There may exist physical or friendship connections between the nodes known in advance, which directly yield a graph. At the other extreme end, there may be no auxiliary information available about the connectedness of the nodes, and the methods for finding an adjacency matrix are then entirely based on some limited data. Methods for estimating the graph Laplacian matrix have been developed in [14–17]. Third, the choice of the adjacency matrix may be based on other variables, which are not perfectly correlated with the graph signal which we are interested in. In both the second and the third cases, the process of adjacency matrix learning is necessarily stochastic. As a result, there are multiple sources of errors which lead to imperfect learning of the adjacency matrix. Even in the case when the choice of the adjacency matrix seems to be obvious and based on the known physical or friendship connections between the nodes, such choice still may not be the best one.

Our focus in this paper is to study consequences of imperfect specifications of graph signal adjacency matrix to GSP tasks. The source of errors in the adjacency matrix learning may be related

* Corresponding author.
E-mail address: esa.ollila@aalto.fi (E. Ollila).

[1] The GSP literature is developing fast and by now covers also sampling theory for graph signals [8,9], stationarity theory for graph signals [10], and percolation [11] of graph signals. Graph filters have been applied for example in distributed average consensus problem [12,13].

https://doi.org/10.1016/j.sigpro.2021.108256
0165-1684/

to or be completely irrelevant to the sources of errors in the signal measurements. However, even when the sources of errors in the adjacency matrix learning and signal value measurements are the same, the effect of errors in the adjacency matrix on GSP tasks are different from those caused by the errors in signal measurements. Considerations of the graph learning errors can be found from Ghosh and Boyd [18], Sydney et al. [19], Ceci and Barbarossa [20]. For example, in [20], the eigen-decomposition of the graph Laplacian was analyzed when small subsets of edges are added or deleted. Based on the analysis, robust methods were built for GSP tasks such as signal recovery, label propagation and clustering, when the knowledge of edgewise probabilities of errors in the graph topology is available.

There are also some works which study robustness with respect to graph topology errors in specific tasks. For example, total least-squares approach for robust graph signal recovery was proposed in [21]. Graph neural networks were found to be stable to small changes in graph topology in [22]. Node-variant graph filters were introduced in [23], and they were shown by numerical experiments to be more robust to errors in graph shift operator than node-invariant filters. For distributed average consensus problem, [24] proposed a method which optimizes worst case performance when the eigenvalue distribution is considered uncertain. In [25], a new centrality measure was introduced and the robustness with respect to edge weight perturbation was considered. The robustness to some missing edges was also part of the comparison of autoregressive moving average graph filters and finite impulse response graph filters for time-varying graph signals in [26]. For spectral graph filters, [27] defined Caley smoothness filter space where the filter perturbation is bounded by a constant times the perturbation of the graph. Also [28] derived results on the stability of polynomial spectral graph filters and empirically tested the effects of structural changes in the graph. In the presence of erroneous edges, [29] studied differences between small world graph and Erdös–Rényi graph, [30] studied clustering of nodes into two communities, and [31] studied the distribution of the edge count change. Finally, [32] have proposed tensor graph convolutional network architecture with the intention to be robust to edge perturbations. Also [33] construct graph convolutional neural networks.

The starting point for studying GSP performance under the condition of imperfect knowledge of graph signal adjacency matrix is the modelling of graph errors. As the errors in the signal modelling in the classical signal processing may lead to significant performance degradation (see for example [34]), the errors in the adjacency matrix may have a significant effect on the GSP performance. Then the development of robust GSP methods, just as the development of robust signal processing methods [34], is of a great importance. The aim of this paper is to develop justifiable and generic enough models for adjacency matrix mismatches. It also studies the effects of the mismatched adjacency matrix on the performance of some more traditional GSP applications such as filtering of graph signals as well as independent component analysis (ICA) of graph signals, which is also referred to as graph decorrelation (GraDe) in the context of separating signals based on graph structure only.

The paper is organized as follows. Section II recalls basic definitions of graphs and introduces and motivates the use of Erdös–Rényi graph as the basic GSP error/distortion model of the adjagency matrix. Then the different types of graph error models are introduced in Section III. In Section IV, the effect of adjacency matrix mismatch for several GSP applications is studied using simulations and real data examples. Section V concludes the paper. All proofs of theorems and propositions can be found from the supplementary material.

Notations: We use boldface capital letters for matrices, boldface lowercase letters for vectors, and capital calligraphic letters for sets. The exceptions are $\mathbf{1}_N$ which is the N-dimensional vector full of ones, the $M \times N$ matrix full of ones $\mathbf{1}_{M\times N} = \mathbf{1}_M \mathbf{1}_N^\top$, and $\mathbf{1}_{\mathbf{A}}$ is a matrix of the same size as $\mathbf{A}$, such that $[\mathbf{1}_{\mathbf{A}}]_{i,j} = 1$, if $a_{i,j} \neq 0$ and $[\mathbf{1}_{\mathbf{A}}]_{i,j} = 0$, if $a_{i,j} = 0$. The matrix $\mathbf{I}_{N\times N}$ is the $N \times N$ identity matrix. The notations $(\cdot)^\top$, $\odot$, $\|\cdot\|$, $\mathrm{tr}\{\cdot\}$, $\mathbb{P}(\cdot)$, $\mathbb{E}\{\cdot\}$, and $\mathrm{var}(\cdot)$ stand for the transpose, Hadamard product, Euclidian norm of a vector, trace of a matrix, probability, mathematical expectation, and variance, respectively. The notation for diagonal elements of a matrix $\mathbf{A}$ is $\mathrm{diag}(\mathbf{A})$, and the notation $N(0, \sigma^2)$ stands for Gaussian zero-mean distribution with variance σ^2.

2. Basic construction and motivation

2.1. Basic building blocks

Let us first go through the basics of graphs, and define the graph models which will be used in the paper.

Let $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ be a directed graph that represents the basis of a graph signal, where $\mathcal{N}$ is the set of N nodes and $\mathcal{E}$ is the set of edges. The adjacency matrix of the graph $\mathcal{G}$, denoted as $\mathbf{A}$, is a matrix that satisfies the conditions $a_{i,i} = 0$ for $i = 1, \ldots, N$ and $a_{i,j} \neq 0$ if and only if $(i, j) \in \mathcal{E}$, i.e., there is an edge from node j to node i.

For developing our graph error models, we will use the Erdös–Rényi model according to which a random graph is constructed by connecting nodes randomly with a constant probability [35]. The corresponding graph is denoted as $\mathcal{G} = (\mathcal{N}, \epsilon)$ and its adjacency matrix $\mathbf{\Delta}_\epsilon$ is a random $N \times N$ matrix such that $\mathbb{P}([\mathbf{\Delta}_\epsilon]_{i,j} = 1) = \epsilon$ and $\mathbb{P}([\mathbf{\Delta}_\epsilon]_{i,j} = 0) = 1 - \epsilon$ for all $i \neq j$, and $[\mathbf{\Delta}_\epsilon]_{i,i} = 0$ for $i = 1, \ldots, N$, where each element of the matrix is generated independently from the other elements.

Another graph model which will be used in this paper is the stochastic block model (SBM) [36], where each node belongs to one of r communities and connections within a community are more common than connections between nodes in different communities. Let $N_1, \ldots, N_r$ denote the sizes of the communities and let us assume that the nodes are ordered so that the ith node belongs to the jth community if and only if $\sum_{l=1}^{j-1} N_l < i \leq \sum_{l=1}^{j} N_l$. The probability of the edge existence from a node of the jth community to a node of the ith community is denoted as $p_{i,j}$. A special case of the stochastic block model when $p_{i,i} = p_{j,j} = p$ for all i and j, and $p_{i,j} = p_{k,l} = q$ for any $i \neq j$ and $k \neq l$, is called planted partition model (PPM).

Because of the errors in the adjacency matrix learning, the estimated adjacency matrix deviates from the true one. The graph error models to be defined in Section 3, all share the same additive structure, i.e., have the form

$$\mathbf{W} = \mathbf{A} + \mathbf{E}, \tag{1}$$

where $\mathbf{W}$ presents the estimated adjacency matrix, $\mathbf{A}$ is the correct adjacency matrix, and $\mathbf{E}$ is an unknown error matrix which can be viewed as an analog of the additive error/distortion/noise component (that applies not to the signal points, but to the signal structure) in the traditional signal processing and time series analysis. Notice that $\mathbf{E}$ needs to be a function of $\mathbf{A}$ because it depends on $\mathbf{A}$ whether distorting an edge means adding or deleting it. Moreover, the Erdös–Rényi graph is the basic GSP error/distortion/noise model analogous to the basic Gaussian noise model in the traditional signal processing as will be shown in the next subsection.

The correct adjacency matrix for a given graph signal and a GSP method can be defined as the one which yields the best results. In the case when the data generating process is known, the optimal adjacency matrix in that sense can often be found. On the other hand, when the optimal adjacency matrix is not known, we might still want to use the graph error models to see how much the results vary when the adjacency matrix is perturbed. Then the role

of correct adjacency matrix $\mathbf{A}$ can be assigned to an estimated adjacency matrix as well.

2.2. Motivation for Erdös-Rényi graph error models

In the context of our paper, an important characteristic of the Erdös-Rényi graph is that it does not allow for formation of communities [37], and if applied on the top of another graph, it will not change the essential structure of that graph. Instead, it just disturbs the spectrum of the original graph as will be shown later in this section.

The simplicity and the above-mentioned properties are the main reasons for the use of Erdös-Rényi model as the main block for modelling the mismatches in the graphical structures of graph signals. However, we will further justify it by bringing an analogy to the basic nature of the Gaussian additive noise for modelling errors related to imperfect fit between signal models and actual measured signal values.

Similar to the fact that the key for modelling additive noise is the noise probability density function, the key for modelling graph errors will be the graph function or *graphon*. The term graphon was introduced in [38], in the context of graph sequence theory, and it refers to measurable and symmetric functions mapping from $[0,1]^2$ to [0,1]. Graphons have been used, for example, as graph limit objects [38,39], and as building blocks in the kernel graph model [40]. Moreover, they have been used for spectral analysis of the adjacency matrix [37]. See the Supplementary material for a brief introduction to graphons.

In traditional signal processing, the main reason for the use of Gaussian random variables as error terms between the analytic signal model and actual measurements is the central limit theorem (CLT). This theorem states that the distribution of a sum of independent and identically distributed random variables tends to the Gaussian distribution as the number of summands grows. With respect to modelling errors in graphs, the Erdös-Rényi graph plays a special role due to its simplicity. However, the importance of the Erdös-Rényi graph also follows from the fact that similar to the Gaussian distribution for modelling measurement errors, the Erdös-Rényi graphon leads to a theorem that plays a similar role to that of CLT in the case of modelling the graph structure errors.

Let $\mathcal{W}_\epsilon$ denote the set of all graphons W, which is a weighted graph with uncountable set of nodes, satisfying

$$\int_0^1 \int_0^1 W(x,y)dxdy = \epsilon,$$

where $W(x,y)$ is the edge weight between nodes x and y in the graphon W (see the Supplementary material for an introduction to graphons), and $0 < \epsilon < 1$. Then the following theorem is in order.

Theorem 1. *Let $W_1, \ldots, W_M$ be random samples from $\mathcal{W}_\epsilon$ and $c_1, \ldots, c_M$ be random samples from a distribution with support on [0,1] and expected value c. Assume that $W_1, \ldots, W_M, c_1, \ldots, c_M$ are mutually independent. Then for all $0 \le x, y \le 1$, we almost surely have*

$$\bar{W}(x,y) = \frac{1}{M}\sum_{i=1}^{M} c_i W_i(x,y) \to c\epsilon \quad \text{as} \quad M \to \infty.$$

The proof of the theorem is given at the supplementary material.

Theorem 1 states that the average of many independent graphons converges to a constant, i.e., to the Erdös-Rényi graphon. Thus, the theorem further motivates the use of Erdös-Rényi graphs as the main building block in developing graph error models for GSP, when the summands are considered as noise factors which impede the estimation of the graph structure. In this sense, Theorem 1 is an analog of the CLT.

3. GSP graph error models

3.1. A basic model for unweighted graphs

We start by considering unweighted graphs, for which the adjacency matrix becomes $\mathbf{A} = \mathbf{1}_\mathbf{A}$. First, making a somewhat idealistic assumption that the outcome of the graph signal adjacency matrix learning is accurate enough, that is, assuming that incorrect graph edge learning is equally probable for any edge in the graph, the learning errors can be simply modelled by the Erdös-Rényi[2] graph. Then the actually available learned adjacency matrix of a graph signal can be modelled in terms of the following inaccurate version of $\mathbf{A}$:

$$\mathbf{W} = \mathbf{A} + \mathbf{\Delta}_\epsilon \odot (\mathbf{1}_{N\times N} - 2\mathbf{A}). \tag{M1}$$

According to (M1), the true adjacency matrix of a graph signal is distorted because of the imperfect learning, and this distortion follows the Erdös-Rényi model, where the level of distortion depends on a single parameter, which is the probability ϵ. As a result of the distortion captured by model (M1), an edge can be added with probability ϵ, when there is no edge in the true graph, or an edge of the true graph can be removed with the same probability, if there exists an edge in the true graph. It corresponds to flipping the value from 0 to 1 or from 1 to 0 in the adjacency matrix $\mathbf{1}_\mathbf{A}$ in the positions corresponding to value 1 in the Erdös-Rényi adjacency matrix $\mathbf{\Delta}_\epsilon$. This basic model is introduced due to its simplicity of having only a single parameter, ϵ. However, it may not be sufficiently flexible to describe signal graph errors that appear as outcome in any practical graph learning process. For example, it will change the number of edges significantly if ϵ is large and the number of edges is small.

Model (M1) can be easily modified/revised for applying to undirected graphs by defining lower triangular matrices $\mathbf{\Delta}_\epsilon^l$ analogously to $\mathbf{\Delta}_\epsilon$, and then replacing $\mathbf{\Delta}_\epsilon$ by $\mathbf{\Delta}_\epsilon^l + (\mathbf{\Delta}_\epsilon^l)^\top$. It is also worth noting that all the following graph error models will be given for directed graphs and the same technique as above can be used to derive the corresponding model for undirected graphs.

3.2. Different probabilities of missed and mislearned edges

Basic model (M1) can be easily extended to the case where the probability of removing an edge as a result of distortion from a graph which correctly captures a graph signal, denoted as ϵ_1, is not the same as the probability of adding an edge, which does not exist in the true graph, denoted as ϵ_2. The corresponding inaccurately learned adjacency matrix of a graph signal can then be modelled as

$$\mathbf{W} = \mathbf{A} - \mathbf{\Delta}_{\epsilon_1} \odot \mathbf{A} + \mathbf{\Delta}_{\epsilon_2} \odot (\mathbf{1}_{N\times N} - \mathbf{A}). \tag{M2}$$

The second term on the right-hand side in (M2) corresponds to edge removal and the third term corresponds to edge addition. This is a straightforward extension of basic model (M1), which is, however, very useful for modelling the typical situation in the graph structure learning when particular algorithms are more or less likely to miss an actually existing edge rather than to mislearn an actully non-existing edge [14]. Model (M2) can be interpreted as an application of two Erdös–Rényi graphs on the top of the true graph, where one Erdös–Rényi graph $\mathcal{G} = (\mathcal{N}, \epsilon_2)$ can only add edges which do not exist in the true graph, while the other Erdös-Rényi graph $\mathcal{G} = (\mathcal{N}, \epsilon_1)$ can only erroneously remove actually existing edges. It is easy to see that model (M2) is equivalent to model (M1) when $\epsilon_1 = \epsilon_2 = \epsilon$.

[2] Even though the results in [35] are for undirected graphs only and often Erdös-Rényi model is considered to imply undirected graph, we will use the term Erdös-Rényi for both undirected and directed graphs.

To analyze the perturbation that the above models produce, let us start with a trivial observation that if $\mathbf{A}$ is given by an Erdös-Rényi graph with probability parameter α, then $\mathbf{W}$ from model (M2) is also an Erdös-Rényi graph, and the parameter takes value $\alpha(1-\epsilon_1)+(1-\alpha)\epsilon_2$.

Moreover, if $\mathbf{A}$ is given by the SBM, after applying graph error model (M2), the SBM and the communities in the SBM remain the same as is shown in the following proposition.

Proposition 1. *Assume that* $\mathbf{A}$ *follows the SBM with probability* $p^A_{i,j}$ *for edges from a node in the* j*th community to a node in the* i*th community. Then* $\mathbf{W}$ *from graph error model (M2) follows SBM with the same communities as in* $\mathbf{A}$ *and the probabilities are given by*

$$p^W_{i,j} = p^A_{i,j}(1-\epsilon_1) + (1-p^A_{i,j})\epsilon_2 = (1-\epsilon_1-\epsilon_2)p^A_{i,j} + \epsilon_2.$$

The spectrum of the adjacency matrix can be found in closed form for simple models such as the undirected (for the error model of undirected graphs, see the end of the next subsection) PPM with equal community sizes [41]. Thus, we also have the following proposition for the spectrum perturbation due to the graph error given by (M2).

Proposition 2. *Assume that* $\mathbf{A}$ *follows the PPM with* M *equally large communities, and let* p *be the probability for edges within communities and* q *be the probability for edges between communities. If* $\mathbf{W}$ *follows graph error model (M2), the expected difference between the largest eigenvalues of the true adjacency matrix* $\mathbf{A}$ *and the mismatched one* $\mathbf{W}$ *is given by*

$$\mathbb{E}\{\lambda_1(\mathbf{A}) - \lambda_1(\mathbf{W})\} = N\frac{(\epsilon_1+\epsilon_2)(p+(M-1)q) - M\epsilon_2}{M}.$$

Moreover, the expected difference between the second to Mth largest eigenvalues for $\mathbf{A}$ *and* $\mathbf{W}$*, are mutually equal and given by*

$$\mathbb{E}\{\lambda_k(\mathbf{A}) - \lambda_k(\mathbf{W})\} = N\frac{(\epsilon_1+\epsilon_2)(p-q)}{M}, \quad k = 2,\dots,M.$$

The relative change of the largest eigenvalue is thus $\epsilon_1 + (1 - M/(p+(M-1)q))\epsilon_2$ and the relative change of next $M-1$ largest eigenvalues is $\epsilon_1+\epsilon_2$.

3.3. Generalized graph error model

It is assumed in models (M1) and (M2) that mislearning the edge status is equally probable (although the probabilities of adding non-existing edge and removing existing edge may be different) for all pairs of nodes. For some adjacency matrix estimation methods, this assumption might not hold strictly even if the pairwise connections were equally strong. To allow the differences in learning accuracy, caused by the structure of the graph or the connectivity strength differences, we formulate the following generalized graph error model.

Consider an unweighted adjacency matrix $\mathbf{A}$. Define $\mathcal{D} = \{\mathbf{D}_1,\dots,\mathbf{D}_K\}$, where $\mathbf{D}_1,\dots,\mathbf{D}_K$ are $N\times N$ matrices satisfying $[\mathbf{D}_k]_{ij} \in \{0,1\}$ for all $k = 1,\dots,K$ and $i,j = 1,\dots,N$, and $\sum_{k=1}^K \mathbf{D}_k = \mathbf{1}_{N\times N} - \mathbf{I}_{N\times N}$. Each matrix $\mathbf{D}_k$ presents the pairs of nodes, indicated by ones, for which the probabilities of mislearning the existence of the edges are equal. The probabilities of removing edges in the subsets are given in $\boldsymbol{\epsilon}_1 = \{\epsilon_{11},\dots,\epsilon_{K1}\}$, and the probabilities of adding edges are given in $\boldsymbol{\epsilon}_2 = \{\epsilon_{12},\dots,\epsilon_{K2}\}$. The graph error model specified by $\mathcal{D}$, $\boldsymbol{\epsilon}_1$, and $\boldsymbol{\epsilon}_2$ can then be written as

$$\mathbf{W} = \mathbf{A} - \sum_{k=1}^K \Delta_{\epsilon_{k1}} \odot \mathbf{D}_k \odot \mathbf{A} + \sum_{k=1}^K \Delta_{\epsilon_{k2}} \odot \mathbf{D}_k \odot (\mathbf{1}_{N\times N} - \mathbf{A}). \tag{M3}$$

This construction allows to build very flexible and accurate graph error models, which can adjust to basically any type of graph topology learning errors. However, flexible model requires large K, i.e., a lot of parameters. Therefore, it is of interest in practice to only approximate the graph error by using a reasonable value of K. Also observe that the error model (M2) is a special case of graph error model (M3) obtained by letting $K=1$ and $\mathbf{D}_1 = \mathbf{1}_{N\times N} - \mathbf{I}_{N\times N}$.

To obtain further insights of the generalized graph error model (M3), we take as an example the SBM. Let $\mathbf{C}_{k,m}$ denote an $N\times N$ matrix such that $[\mathbf{C}_{k,m}]_{i,j} = 1$ if $i \neq j$, $\sum_{l=1}^{k-1} N_l < i \leq \sum_{l=1}^{k} N_l$ and $\sum_{l=1}^{m-1} N_l < j \leq \sum_{l=1}^{m} N_l$, and $[\mathbf{C}_{k,m}]_{i,j} = 0$, otherwise. Then we can set, for example, $K = r^2$, $\mathbf{D}_1 = \mathbf{C}_{1,1}$, $\mathbf{D}_2 = \mathbf{C}_{1,2},\cdots,\mathbf{D}_{r^2} = \mathbf{C}_{r,r}$, and we obtain a model where the probability of mislearning an edge depends on which communities the start and the end nodes belong to. For such model, the following result about the structure of the true adjacency matrix $\mathbf{A}$ and that of the mismatched one $\mathbf{W}$ as modelled by (M3) is of interest.

Proposition 3. *Assume that the true adjacency matrix* $\mathbf{A}$ *follows the SBM with* r *communities, the mismatched adjacency matrix* $\mathbf{W}$ *follows model (M3), and* $\mathbf{D}_1,\dots,\mathbf{D}_K$ *are chosen according to the SBM structure of* $\mathbf{A}$*. Then* $\mathbf{W}$ *retains SBM with the same communities.*

It is also interesting to note that a natural graph error model related to the previously used PPM can be derived using model (M3) by letting $K=2$, $\mathbf{D}_1 = \sum_{k=1}^r \mathbf{C}_{k,k}$, and $\mathbf{D}_2 = \mathbf{1}_{N\times N} - \mathbf{I}_{N\times N} - \sum_{k=1}^r \mathbf{C}_{k,k}$.

3.4. Models for weighted graphs

The above defined models can be extended to signals on weighted graphs. Then, in addition to adding and removing edges, we have to consider changes in the edge weights and also figure out how to generate the weights for added edges in a mismatched adjacency matrix.

Let us start with extending graph error model (M2) (extension of graph error model (M1) is similar). Let $\mathcal{A}$ denote the set of nonzero elements of the true graph adjacency matrix $\mathbf{A}$. The inaccurately learned weighted adjacency matrix can be then modelled as

$$\begin{aligned}\mathbf{W} =& \mathbf{A} + (\mathbf{1}_{N\times N} - \boldsymbol{\Delta}_{\epsilon_1}) \odot \mathbf{1}_{\mathbf{A}} \odot \boldsymbol{\Sigma}_c - \boldsymbol{\Delta}_{\epsilon_1} \odot \mathbf{A} \\ &+ \boldsymbol{\Delta}_{\epsilon_2} \odot \mathbf{B} \odot (\mathbf{1}_{N\times N} - \mathbf{1}_{\mathbf{A}}).\end{aligned} \tag{M2w}$$

Like in graph error model (M2), ϵ_1 is the probability to erroneously removing edges while ϵ_2 is the probability of erroneously adding edges. The second term in (M2w) perturbs the weights of the remaining edges using an $N\times N$ matrix $\boldsymbol{\Sigma}_c$ whose elements are drawn from a zero mean Gaussian distribution with variance $c\cdot\sigma^2$, where σ^2 is the sample variance of $\mathcal{A}$ and c is the variance multiplier parameter. The third term in (M2w) models the erroneous removal of edges and the fourth term in (M2w) models the erroneous addition of edges with weights given by the matrix $\mathbf{B}$ which is an $N\times N$ matrix whose elements are derived by the same rules which were used for edge weights of $\mathbf{A}$ if applicable, or alternatively the elements can be drawn from $\mathcal{A}$ with replacement. Typically for practical GSP tasks, only positive edge weights are considered, and then, if the above described process produces negative elements in $\mathbf{W}$, they should be put to zero.

Similarly, the weighted generalized graph error model can be written as a weighted extension of graph error model (M3), that is,

$$\begin{aligned}\mathbf{W} = \mathbf{A} + \sum_{k=1}^K \Big(&(\mathbf{1}_{N\times N} - \boldsymbol{\Delta}_{\epsilon_{k1}}) \odot \mathbf{D}_k \odot \mathbf{1}_{\mathbf{A}} \odot \boldsymbol{\Sigma}_{ck} \\ &- \boldsymbol{\Delta}_{\epsilon_{k1}} \odot \mathbf{D}_k \odot \mathbf{A} + \boldsymbol{\Delta}_{\epsilon_{k2}} \odot \mathbf{B}_k \odot \mathbf{D}_k \odot (\mathbf{1}_{N\times N} - \mathbf{1}_{\mathbf{A}}) \Big),\end{aligned} \tag{M3w}$$

where ϵ_{k1}, ϵ_{k2} and $\mathbf{D}_k$, $k = 1,\dots,K$ are defined in the same way as in Section 3.3, and $\mathbf{B}_k$ and $\boldsymbol{\Sigma}_{ck}$, $k = 1,\dots,K$ are analogous to $\mathbf{B}$ and $\boldsymbol{\Sigma}_c$ above.

4. Numerical studies

4.1. Graph error effect on GMA graph filter

As in any other GSP task, the adjacency matrix has a key role in graph filtering. In fact, any linear and shift-invariant filter (commutes with other shift-invariant filters) can be written as a polynomial of the form [1]

$$F_{\mathrm{GMA}}(\mathbf{W}) = h_0\mathbf{I}_{N\times N} + h_1\mathbf{W} + \cdots + h_K\mathbf{W}^K,$$

where K is smaller than or equal to the degree of the minimal polynomial of $\mathbf{W}$, and GMA stands for graph moving average.

The coefficients $h_0, \ldots, h_K$ can be found by using least squares for solving the system of equations

$$\begin{aligned} h_0 + h_1\lambda_1 + \cdots + h_K\lambda_1^K &= \alpha_1 \\ &\vdots \\ h_0 + h_1\lambda_N + \cdots + h_K\lambda_N^K &= \alpha_N \end{aligned}$$

where $\lambda_1, \ldots, \lambda_N$ are eigenvalues of $\mathbf{W}$ ordered in increasing order of their distance to the maximum magnitude, i.e., $d_n = |\max\{|\lambda_1|, \ldots, |\lambda_N|\} - \lambda_n|$. Then $\lambda_1, \ldots, \lambda_N$ can be interpreted as graph frequencies from lowest to highest [42], and $\alpha_1, \ldots, \alpha_N$ are the corresponding frequency responses.

In this example, we study the filter sensitivity to the choice of the adjacency matrix, when the high-pass filter is defined by the following frequency responses

$$\alpha_n = \begin{cases} 1, & \text{if } d_n > \text{median}\{d_1, \ldots, d_N\} \\ 0, & \text{otherwise} \end{cases}$$

and it is applied to detecting malfunctioning sensors as in [42]. The data are the daily temperatures in the year 2016 from 150 Finnish weather stations [43]. Stations that had at most one missing observation were selected, and since those missing observations were from the same day in November, after dropping that day out, we have a clean data of 365 days (leap year). The technique to detect the outlying measurements, presented in [42], is to high-pass filter the data and threshold the Fourier transform coefficients of the output by the maximum absolute value of the GFT coefficients from the three previous days. If any of the coefficients exceed the threshold value, it is diagnosed that at least one of the sensors is not working properly.

In this example, there is no obvious choice of a benchmark adjacency matrix,[3] but we choose a weighted 6-nearest neighbors graph like in [42] with edge weights

$$a_{kl} = \frac{e^{-(d_{kl}/20)^2}}{\sqrt{\sum_{j\in\mathcal{N}_k} e^{-(d_{kj}/20)^2} \sum_{j\in\mathcal{N}_l} e^{-(d_{lj}/20)^2}}}, \tag{2}$$

where d_{kl} is the distance between the locations of the kth and lth sensors in kilometers. A better graph might be obtained by using more information such as altitude and distance from the sea. Hence, the graph errors could arise from disregarding useful information. Since we only have distances to generate the benchmark graph, it is not optimal and the graph errors have to be created randomly without any realistic mechanism. We use the general graph error model because it is not sensible to connect two stations that are far away from each other. We set the threshold to 250 km above which a pair of stations is treated as no longer connected. Otherwise, the probabilities of removing and creating connections have constant values ϵ_1 and ϵ_2 between the pairs. Hence, in graph error model (M3w), which is then applicable here, we have the matrices

$$[\mathbf{D}_1]_{kl} = \begin{cases} 1, & \text{if } d_{kl} \le 250 \\ 0, & \text{otherwise} \end{cases}, \quad [\mathbf{D}_2]_{kl} = \begin{cases} 1, & \text{if } d_{kl} > 250 \\ 0, & \text{otherwise} \end{cases}$$

and probabilities $\epsilon_{11} = \epsilon_1$, $\epsilon_{12} = \epsilon_2$, $\epsilon_{21} = \epsilon_{22} = 0$.

By perturbing the edge weights, we deviate slightly from graph error model (M3w), so that the variance of the Gaussian component related to incoming edges of a given node is the variance multiplier c times the variance of the incoming edge weights of that node in the 6-nearest neighbors graph, not the variance of the edge weights of the whole 6-nearest neighbors graph. We do not allow negative weights, but use zero weights instead.

In the experiment, we change one sensor value at a time to +20 Celsius degrees and perform the test described above using adjacency matrices given by all combinations of edge removing probabilities $\epsilon_1 = 0, 0.01, 0.02, 0.03$, edge adding probabilities $\epsilon_2 = 0, 0.1, 0.2, 0.3$, and variance parameter values $c = 0, 0.005, 0.01, 0.015$.

Finding the malfunctioning sensors can be seen as a binary classification task. Theoretically, all classifiers (the same method with different graphs) have equal proportion of false positives of about 25%, because the maximum GFT coefficient today is larger than the maximum of the GFT coefficients from the three previous days one out of four times when there are no outliers. Hence, we can focus on the accuracy in finding the true positives. The benchmark adjacency matrix given by $\epsilon_1 = \epsilon_2 = c = 0$ found the malfunctioning sensors with 84% accuracy. For the other combinations, 200 realizations of the adjacency matrix are generated.

In Fig. 1, we give the averages for different values of parameters of interest when other parameters are zeros. There are no visible patterns of interactions between the parameters, i.e., the shapes of the curves look identical at all levels of the other two parameters. The results show that adding edges to the graph has no effect, but removing edges weakens the performance. Also, the perturbation of the weights has negative impact, though the reduction is significant only for values from 0 to 0.005, and for $c = 0.005, 0.01, 0.015$, the numbers are approximately the same. This suggests that the number of edges in the 6-nearest neighbors graph might be too small for these purposes, but that the edge weights are good as they are, because small changes in them worsen the performance. The reason why addition of edges has smaller effect can be partly explained by the fact that they have smaller edge weight than the edges in the original nearest neighbors graph.

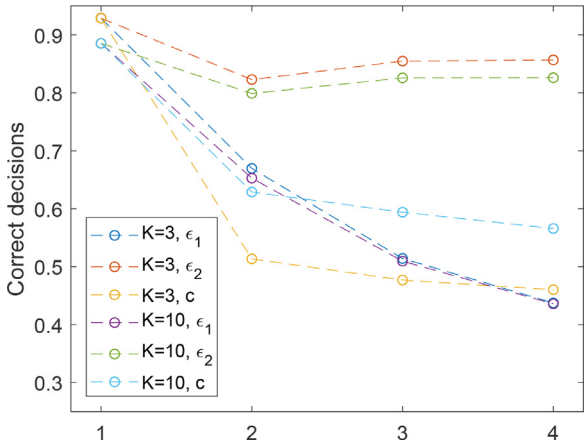

Fig. 1. The average share of correctly detected outlying sensor values when only one of the parameters, ϵ_1, ϵ_2, or c varies while the remaining parameters are set to zero. The varying parameter can take 4 different values indicated in the x-axis; $\epsilon_1 = 0, 0.01, 0.02, 0.03$, $\epsilon_2 = 0, 0.1, 0.2, 0.3$ and $c = 0, 0.005, 0.01, 0.015$.

[3] In other examples, there always exist an obvious choice for benchmark adjacency matrix.

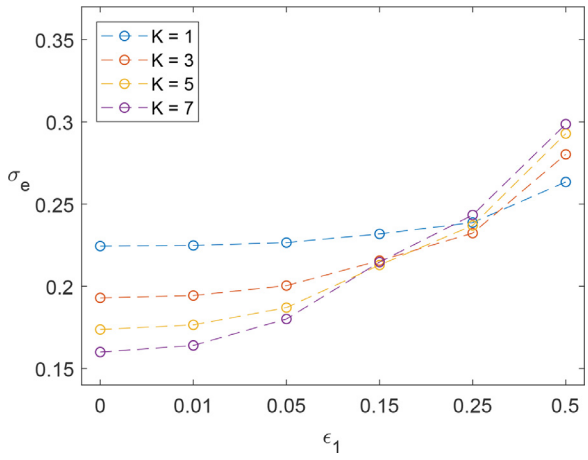

Fig. 2. Averages of σ_e over 2000 repetitions for GARMA filter orders $K = 1, 3, 5, 7$ and probabilities ϵ_1 in graph error model (M2), when probability of adding an edge $\epsilon_2 = 0$.

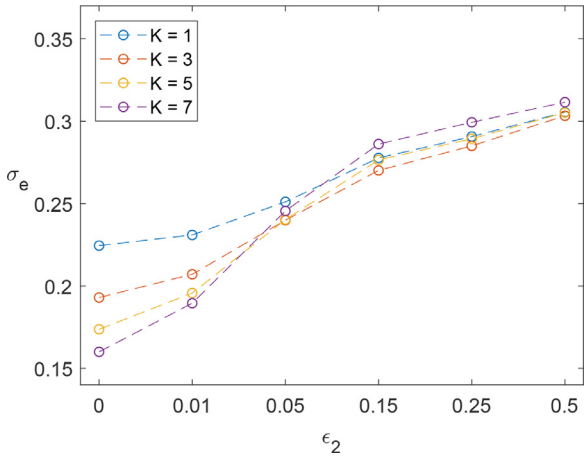

Fig. 3. Averages of σ_e over 2000 repetitions for GARMA filter orders $K = 1, 3, 5, 7$ and probabilities ϵ_2 in graph error model (M2), when probability of removing an edge $\epsilon_1 = 0$.

4.2. Graph error effect on graph autoregressive moving average filter

Next we study the graph error effect on graph autoregressive moving average (GARMA) filter [44]. Specifically, we investigate the relation of the used filter order with respect to the errors in the estimated adjacency matrix $\mathbf{W}$.

First order GARMA filter has coefficients ϕ, ψ and c, and it uses a graph Laplacian matrix $\mathbf{L}$. Here we use the translated normalized Laplacian $\mathbf{L} = -\mathbf{T}^{-1/2}\mathbf{W}\mathbf{T}^{-1/2}$, where $\mathbf{T}$ is a diagonal matrix of node degrees and $\mathbf{W}$ is a symmetric adjacency matrix. Note that the eigenvalues of $\mathbf{L}$ are in the interval $[-1, 1]$, but as in the GMA filter case, they can be very different for the cases of the true $\mathbf{A}$ and the estimated adjacency matrix $\mathbf{W}$.

The output of GARMA(1) filter is given by $\mathbf{z} = \mathbf{y} + c\mathbf{x}$, where $\mathbf{x}$ is the input and $\mathbf{y}$ is the state after convergence of recursion $\mathbf{y}_{t+1} = \phi\mathbf{L}\mathbf{y}_t + \psi\mathbf{x}$. GARMA filter of order K can be constructed from K parallel GARMA(1) filters, whose coefficients are designed based on the eigenvalues of $\mathbf{L}$. Again, the errors in the adjacency matrix $\mathbf{W}$ have a significant effect on the filter directly via $\mathbf{L}$, but also via distorting the spectrum of $\mathbf{L}$.

We follow the simulation setup in [26]. Undirected and unweighted graphs are created by generating $N = 100$ random points on the area $[0, 1] \times [0, 1]$ using the uniform distribution, and connecting two points if the distance between them is less than $0.15\sqrt{2}$.

Let $\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$ be the eigendecomposition of the translated normalized graph Laplacian, and let λ_n and $\mathbf{v}_n$ denote the nth diagonal element of $\mathbf{\Lambda}$ and the nth column vector of $\mathbf{V}$, i.e., the nth eigenvalue and eigenvector, respectively. The eigenvalues are in the interval $[-1, 1]$. Then the graph signal is given by $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{n}$, where $\bar{\mathbf{x}}$ is a low frequency signal satisfying $\bar{\mathbf{x}}^\top\mathbf{v}_n = 1$, if $\lambda_n < 0$, and $\bar{\mathbf{x}}^\top\mathbf{v}_n = 0$, otherwise, and $\mathbf{n}$ is Gaussian noise with zero mean and covariance matrix $0.1\mathbf{I}_{N\times N}$.

Frequency responses of GARMA graph filters of orders $K = 1, 3, 5, 7$ are designed to match the frequency content of the signal $\bar{\mathbf{x}}$. Outputs of the filters, denoted by $\mathbf{z}^{(e)}$, are compared to the output of the ideal filter, denoted by $\mathbf{z}^{(d)}$. The performance is then measured using the square root of the mean square error $\sigma_e = [\mathrm{tr}(\mathbf{e}\mathbf{e}^\top)/N]^{1/2}$, where $\mathbf{e} = \mathbf{z}^{(e)} - \mathbf{z}^{(d)}$.

The values in Figs. 2 and 3 are averages over 2000 runs for each pair (ϵ_1, ϵ_2) in graph error model (M2) that is applicable here. The results show that higher order GARMA filters are more accurate when the correct adjacency matrix is used. On the other hand, the lower order filters are more robust to graph errors, and thus all filters perform almost equally when the error probabilities grow.

4.3. Graph error effect on ICA of graph signals

In this example, we use both numerical study and theoretical results to investigate the effect of non-optimal signal graph knowledge to the performance of GraDe [7,45] for graph signal separation.

Let $\mathbf{X} \in \mathbb{R}^{P\times N}$ denote centered P-dimensional graph signal generated as a mixture of independent components according to the model [45]

$$\mathbf{X} = \mathbf{\Omega}\mathbf{Z}$$

where $\mathbf{\Omega} \in \mathbb{R}^{P\times P}$ is a full rank mixing matrix, $\mathbf{Z} \in \mathbb{R}^{P\times N}$ is the matrix of P mutually independent graph signals with zero means and unit variances. The goal of ICA is to estimate the unmixing matrix $\mathbf{\Gamma} = \mathbf{\Omega}^{-1}$ using only the signal matrix $\mathbf{X}$.

Let $\mathbf{X}_\mathrm{w} = \hat{\mathbf{S}}_0^{-1/2}\mathbf{X}$ be the whitened signals, where $\hat{\mathbf{S}}_0$ is the sample covariance matrix of $\mathbf{X}$. In GraDe, the unmixing matrix estimate is obtained by diagonalizing/jointly diagonalizing one or more graph autocorrelation matrices defined by

$$\hat{\mathbf{S}}_k(\mathbf{W}) = \frac{1}{N-k}(\mathbf{X}_\mathrm{w}\mathbf{W}^k\mathbf{X}_\mathrm{w}^\top), \quad k = 1, \ldots, K,$$

i.e., by finding the orthogonal $\mathbf{U}$ which maximizes the objective function

$$\sum_{k=1}^{K} \|\mathrm{diag}(\mathbf{U}\hat{\mathbf{S}}_k(\mathbf{W})\mathbf{U}^\top)\|^2.$$

The unmixing matrix estimate is then given as $\hat{\mathbf{\Gamma}} = \mathbf{U}\hat{\mathbf{S}}_0^{-1/2}$. A fast algorithm for the joint diagonalization is available in [46] and it is applicable for the case when the shift matrix $\mathbf{W}$ is chosen to be symmetric, or the graph-autocorrelation matrices are symmetrized. The unmixing matrix estimate for an inaccurately learned adjacency matrix $\mathbf{W}$ is denoted as $\hat{\mathbf{\Gamma}}(\mathbf{W})$. Notice that GraDe reduces to the well-known second-order blind identification (SOBI) estimator [47], when $\mathbf{W}$ is the cyclic graph.

We will use the following (see [48] for details) asymptotic result, derived in the context of the SOBI estimator, for an unmixing matrix estimate $\hat{\mathbf{\Gamma}}$ obtained using joint diagonalization of matrices $\hat{\mathbf{S}}_1, \ldots, \hat{\mathbf{S}}_K$. When $\mathbf{\Omega} = \mathbf{I}_{P\times P}$, for $i \neq j$, we have

$$\sqrt{N}\,(\hat{\gamma}_{ii} - 1) = -\frac{1}{2}\sqrt{N}\,([\hat{\mathbf{S}}_0]_{ii} - 1) + o_p(1)$$

and

$$\sqrt{N}\,\hat{\gamma}_{ij} = \frac{\sum_k(\lambda_{ki} - \lambda_{kj})(\sqrt{N}\,[\hat{\mathbf{S}}_k]_{ij} - \lambda_{ki}\sqrt{N}\,[\hat{\mathbf{S}}_0]_{ij})}{\sum_k(\lambda_{ki} - \lambda_{kj})^2} + o_p(1), \qquad (3)$$

Table 1
$R(\mathbf{A},\mathbf{W})$ for $\mathbf{A}$ with $\alpha = 0.05$ and $\mathbf{W}$ given by $\epsilon_1 = 0, 0.1, \ldots, 0.5$ and $\epsilon_2 = 0, 0.01, \ldots, 0.05$.

$\epsilon_1 \backslash \epsilon_2$	0	0.01	0.02	0.03	0.04	0.05
0	1.00	0.81	0.68	0.58	0.51	0.45
0.1	0.88	0.71	0.58	0.49	0.43	0.38
0.2	0.77	0.60	0.49	0.41	0.35	0.30
0.3	0.66	0.50	0.40	0.33	0.28	0.24
0.4	0.56	0.41	0.31	0.26	0.21	0.18
0.5	0.46	0.31	0.24	0.19	0.15	0.13

Table 2
$\hat{R}(\mathbf{A},\mathbf{W})$ from 1000 repetitions for $\alpha = 0.05$ and $\epsilon_1 = 0, 0.1, \ldots, 0.5$ and $\epsilon_2 = 0, 0.01, \ldots, 0.05$.

$\epsilon_1 \backslash \epsilon_2$	0	0.01	0.02	0.03	0.04	0.05
0	1.00	0.81	0.67	0.56	0.48	0.44
0.1	0.88	0.65	0.56	0.47	0.43	0.35
0.2	0.76	0.59	0.46	0.40	0.34	0.29
0.3	0.62	0.46	0.37	0.32	0.27	0.25
0.4	0.52	0.37	0.29	0.25	0.21	0.20
0.5	0.43	0.31	0.25	0.20	0.17	0.16

where $\lambda_{ki} \triangleq \mathbb{E}\{[\mathbf{S}_k]_{ii}\}$, and $o_p(1)$ stands for negligible terms. The diagonal elements of $\hat{\mathbf{\Gamma}}$ do not depend asymptotically on $\hat{\mathbf{S}}_1, \ldots, \hat{\mathbf{S}}_K$, and thus, in the case of graph signals ICA, do not depend on $\mathbf{W}$. Therefore, the sum of variances of the off-diagonal elements

$$\mathrm{SOV}(\hat{\mathbf{\Gamma}}(\mathbf{W})) = N \sum_{j \neq i} \mathrm{var}(\hat{\mathbf{\Gamma}}(\mathbf{W})_{ij}) \tag{4}$$

can be used when comparing the separation efficiencies induced by different choices of $\mathbf{W}$. We will use the ratio, $R(\mathbf{W}_1, \mathbf{W}_2) = \mathrm{SOV}(\hat{\mathbf{\Gamma}}(\mathbf{W}_1))/\mathrm{SOV}(\hat{\mathbf{\Gamma}}(\mathbf{W}_2))$, where $\mathbf{W}_1$ is the adjacency matrix which is used for generating the source components and $\mathbf{W}_2$ is a perturbed version of the adjacency matrix.

We consider ICA model where the independent components are GMA signals of order 1, denoted GMA(1), i.e., $\mathbf{z} = \mathbf{y} + \theta\mathbf{A}\mathbf{y}$, where θ denotes the MA coefficient (which is different for each independent component) and $\mathbf{A}$ is symmetric and unweighted adjacency matrix. We evaluate the performance of the GraDe estimate with $K = 1$, so only one autocorrelation matrix is diagonalized. The matrix $\mathbf{W}$ is a mismatched version $\mathbf{A}$ obtained according to graph error model (M2) for undirected graphs. In this model, the asymptotic variances of $\hat{\mathbf{\Gamma}}(\mathbf{W})$, which are needed for computing $\mathrm{SOV}(\hat{\mathbf{\Gamma}}(\mathbf{W}))$, can be calculated using (3). Expressions for these asymptotic variances can be derived in the same way as the corresponding formulas for the variances in the time series context in [48].[4]

The performance in the numerical simulations is measured using the minimum distance (MD) index [49]

$$D(\hat{\mathbf{\Gamma}}) \triangleq \frac{1}{\sqrt{P-1}} \inf_{\mathbf{C} \in \mathcal{C}} \|\mathbf{C}\hat{\mathbf{\Gamma}}\mathbf{\Omega} - \mathbf{I}_{P\times P}\|$$

where $\mathcal{C} \triangleq \{\mathbf{C} \quad : \text{ each row and column of } \mathbf{C} \text{ has exactly one non-zero element}\}$. The MD index takes values between zero and one, and it is invariant with respect to the mixing matrix. Also, there is a connection between the minimum distance index and the sum of variances of the off-diagonal elements when $\mathbf{\Omega} = \mathbf{I}_{P\times P}$, given as

$$N(P-1)\mathbb{E}\{D(\hat{\mathbf{\Gamma}})^2\} \to \mathrm{SOV}(\hat{\mathbf{\Gamma}}), \text{ as } N \to \infty \tag{5}$$

where SOV is defined in (4).

For two sets of estimates, $\mathbf{W}_1$ and $\mathbf{W}_2$, we define

$$\hat{R}(\mathbf{W}_1, \mathbf{W}_2) = \mathrm{ave}\{D(\hat{\mathbf{\Gamma}}(\mathbf{W}_1))^2\}/\mathrm{ave}\{D(\hat{\mathbf{\Gamma}}(\mathbf{W}_2))^2\},$$

where the averages are found over 1000 Monte Carlo trials. Eq. (5) implies that $\hat{R}(\mathbf{W}_1, \mathbf{W}_2) \approx R(\mathbf{W}_1, \mathbf{W}_2)$ for large N.

Erdös–Rényi matrices $\mathbf{A} = \mathbf{\Delta}_\alpha$ with different values of α are used as the adjacency matrices of GMA signals. The estimate $\hat{\mathbf{\Gamma}}(\mathbf{A})$ (with true $\mathbf{A}$) is a natural benchmark to which we compare the estimates obtained using $\mathbf{W}$.

In Tables 1 and 2, the values of $R(\mathbf{A},\mathbf{W})$ and $\hat{R}(\mathbf{A},\mathbf{W})$ are shown, respectively, when $\mathbf{A}$ is 1000×1000 matrix with $\alpha = 0.05$ and there are $P = 4$ independent components genered as GMA(1) signals with MA coefficients given by $\theta = 0,\ 0.2,\ 0.4$, and 0.6. For Table 2, we generated 1000 datasets for each pair (ϵ_1, ϵ_2) and always generate a new $\mathbf{W}$. In Table 1, the sum of variances is an average for ten $\mathbf{W}$'s, even though $\mathrm{SOV}(\hat{\mathbf{\Gamma}}(\mathbf{W}))$ is quite stable for fixed ϵ_1 and ϵ_2. As can be noted, the simulation results match the theoretical values quite well. When looking at the results with respect to error probabilities ϵ_1 and ϵ_2, it seems that GraDe is more sensitive to adding irrelevant edges than missing the real edges.

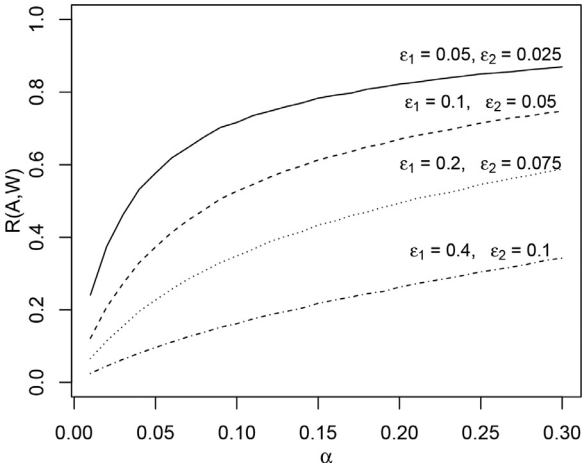

Fig. 4. Ratio of the theoretical variances as a function of α for four choices of (ϵ_1, ϵ_2).

For four selected pairs (ϵ_1, ϵ_2), Fig. 4 plots $R(\mathbf{A},\mathbf{W})$ as a function of parameter α that was used in creating the adjagency matrix $\mathbf{A}$. The curves display the averages of ten values given by different $\mathbf{W}$'s. As expected, the efficiency loss caused by inaccuracy in the adjacency matrix is the larger, the more sparse the graph is.

5. Conclusions and discussion

In this paper, the effect of graph adjacency matrix mismatch has been analyzed and graph error models for different types of graphs, e.g., directed/undirected and weighted/unweighted graphs, have been developed. The complexity of the error models varies from simple Erdös-Rényi type model to one which captures non-constant edge mislearning probabilities. The latter is of interest because it has been reported in GSP literature that deleting one edge can have immensely larger effect than deleting another edge. Better understanding and formalization of what kind of edges are important or why connecting some pairs of nodes is more harmful than others is therefore crucial. The graph error models have been applied for studying graph error effects in graph signal filtering and graph signal ICA applications, where both theoretical arguments and numerical studies for real and synthetic data have been used. The results for different application examples differed in whether missing or extra links were more detrimental, and in the GARMA filter example, it was observed that the higher-order filters were more sensitive to graph errors. These findings suggest

[4] The extension of asymptotic variance expression in [48] from the time series context to the context of graph signals is straightforward, but the formula appears to be too long to be presented here because of the space limitation.

that the graph error effects need to be studied case by case, and that competing GSP methods may differ in terms of their robustness to graph errors. Therefore, studying the robustness properties of the GSP methods as well as developing robust GSP methods is of high importance for GSP in general.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.sigpro.2021.108256

CRediT authorship contribution statement

Jari Miettinen: Writing – review & editing, Conceptualization, Methodology, Software, Visualization, Formal analysis. **Sergiy A. Vorobyov:** Writing – review & editing, Conceptualization, Supervision. **Esa Ollila:** Writing – review & editing, Conceptualization, Supervision.

References

[1] A. Sandryhaila, J.M.F. Moura, Discrete signal processing on graphs, IEEE Trans. Signal Process. 61 (7) (2013) 1644–1656.
[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 (3) (2013) 83–98.
[3] A. Ortega, P. Frossard, J. Kovacevic, J.M.F. Moura, P. Vandergheynst, Graph signal processing: overview, challenges, and applications, Proc. IEEE 106 (5) (2018) 808–828.
[4] S.Y. Tu, A.H. Sayed, Mobile adaptive networks, IEEE J. Select. Top. Signal Process. 5 (4) (2011) 649–664.
[5] D. Acemoglu, G. Como, F. Fagnani, A. Ozdaglar, Opinion fluctuations and disagreement in social networks, Math. Oper. Res. 38 (1) (2013) 1–27.
[6] W. Huang, L. Goldsberry, N.F. Wymbs, S.T. Grafton, D.S. Bassett, A. Ribeiro, Graph frequency analysis of brain signals, IEEE J. Select. Top. Signal Process. 10 (7) (2016) 1189–1203.
[7] F. Blöchl, A. Kowarsch, F.J. Theis, Second-order source separation based on prior knowledge realized in a graph model, in: V. Vigneron, V. Zarzoso, E. Moreau, R. Gribonval, E. Vincent (Eds.), Latent Variable Analysis and Signal Separation, LNCS, vol. 6365, Springer, Heidelberg, 2010, pp. 434–441.
[8] S. Chen, R. Varma, A. Sandryhaila, J. Kovacevic, Discrete signal processing on graphs: sampling theory, IEEE Trans. Signal Process. 63 (24) (2015) 6510–6523.
[9] A.G. Marques, S. Segarra, G. Leus, A. Ribeiro, Sampling of graph signals with successive local aggregations, IEEE Trans. Signal Process. 64 (7) (2016) 1832–1843.
[10] A.G. Marques, S. Segarra, G. Leus, A. Ribeiro, Stationary graph processes and spectral estimation, IEEE Trans. Signal Process. 65 (22) (2017) 5911–5926.
[11] S. Segarra, A.G. Marques, G. Leus, A. Ribeiro, Reconstruction of graph signals through percolation from seeding nodes, IEEE Trans. Signal Process. 64 (16) (2016) 4363–4378.
[12] A. Sandryhaila, S. Kar, J.M. Moura, Finite-time distributed consensus through graph filters, in: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., Florence, Italy, 2014, pp. 1080–1084.
[13] S. Safavi, U.A. Khan, Revisiting finite-time distributed algorithms via successive nulling of eigenvalues, IEEE Signal Process. Lett. 22 (1) (2015) 54–57.
[14] X. Dong, D. Thanou, P. Frossard, P. Vandergheynst, Learning laplacian matrix in smooth graph signal representations, IEEE Trans. Signal Process. 64 (23) (2016) 6160–6173.
[15] P.A. Traganitis, Y. Shen, G.B. Giannakis, Network topology inference via elastic net structural equation models, in: Proc. European Signal Processing Conference, Kos island, Greece, Aug.-Sept., 2017, pp. 156–160.
[16] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, M. G. Rabbat, Characterization and inference of graph diffusion processes from observations of stationary signals, IEEE Trans. Signal Inf. Process. Netw. 4 (3) (2018) 481–496.
[17] S. Segarra, S. A. G. Marques, G. Mateos, A. Ribeiro, Network topology inference from spectral templates, IEEE Trans. Signal Inf. Process. Netw. 3 (3) (2017) 467–483.
[18] A. Ghosh, S. Boyd, Growing well-connected graphs, in: Proceedings of the 45th IEEE Conference on Decision and Control, 6605–6611, 2006.
[19] A. Sydney, C. Scoglio, D. Gruenbacher, Optimizing algebraic connectivity by edge rewiring, Appl. Math. Comput. 10 (219) (2013) 5465–5479.
[20] E. Ceci, S. Barbarossa, Graph signal processing in the presence of topology uncertainties, IEEE Trans. Signal Process. 68 (2020) 1558–1573.
[21] E. Ceci, Y. Shen, G.B. Giannakis, S. Barbarossa, Graph-based learning under perturbations via total least-squares, IEEE Trans. Signal Process. 68 (2020) 2870–2882.
[22] F. Gama, J. Bruna, A. Ribeiro, Stability properties of graph neural networks, 2019. arXiv:1905.04497.
[23] S. Segarra, A.G. Marques, A. Ribeiro, Optimal graph-filter design and applications to distributed linear network operators, IEEE Trans. Signal Process. 65 (15) (2017) 4117–4131.
[24] S. Kruzick, J.M. Moura, Optimal filter design for signal processing on random graphs: accelerated consensus, IEEE Trans. Signal Process. 66 (5) (2017) 1258–1272.
[25] S. Segarra, A. Ribeiro, Stability and continuity of centrality measures in weighted graphs, IEEE Trans. Signal Process. 64 (3) (2016) 543–555.
[26] E. Isufi, A. Loukas, A. Simonetto, G. Leus, Filtering random graph processes over random time-varying graphs, IEEE Trans. Signal Process. 65 (16) (2017) 4406–4421.
[27] E. Isufi, A. Loukas, A. Simonetto, G. Leus, On the transferability of spectral graph filters, in: Proc. Int. Conf. Sampling Theory and Applications, IEEE, Bordeaux, France, 2019, pp. 1–5.
[28] H. Kenlay, D. Thanou, X. Dong, On the stability of polynomial spectral graph filters, in: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., 2020, pp. 5350–5354.
[29] T. Cai, T. Liang, A. Rakhlin, On detection and structural reconstruction of small-world random networks, IEEE Trans. Netw. Sci. Eng. 4 (3) (2017) 165–176.
[30] E. Abbe, A.S. Bandeira, A. Bracher, A. Singer, Decoding binary node labels from censored edge measurements: phase transition and efficient recovery, IEEE Trans. Netw. Sci. Eng. 1 (1) (2014) 10–22.
[31] P. Balachandran, E.D. Kolaczyk, W.D. Viles, On the propagation of low-rate measurement error to subgraph counts in large networks, J. Mach. Learn. Res. 18 (61) (2017) 1–33.
[32] V.N. Ioannidis, A.G. Marques, G.B. Giannakis, Tensor graph convolutional networks for multi-relational and robust learning, IEEE Trans. Signal Process. 68 (2020) 6535–6546.
[33] D. Zou, G. Lerman, Graph convolutional neural networks via scattering, Appl. Comput. Harmon. Anal. 49 (3) (2020) 1046–1074.
[34] S.A. Vorobyov, A.B. Gershman, Z.Q. Luo, Robust adaptive beamforming using worst-case performance optimization: a solution to the signal mismatch problem, IEEE Trans. Signal Process. 51 (2) (2003) 313–324.
[35] P. Erdös, A. Rényi, On random graphs, Publ. Math. Debrecen 6 (1959) 290–297.
[36] P.W. Holland, K. Laskey, S. Leinhardt, Stochastic blockmodels: first steps, Soc Netw. 5 (2) (1983) 109–137.
[37] M.W. Morency, G. Leus, Signal processing on kernel-based random graphs, in: Proc. European Signal Processing Conference, Kos Island, Greece, 2017, pp. 385–389.
[38] L. Lovász, B. Szegedy, Limits of dense graph sequences, J. Comb. Theory B 96 (2006) 933–957.
[39] L. Lovász, Large Networks and Graph Limits, vol. 60, Providence: American Mathematical Society, 2012.
[40] B. Bollobás, S. Janson, O. Riordan, The phase transition in inhomogeneous random graphs, Random Struct. Algorithms 31 (1) (2007) 3–122.
[41] K. Avrachenkov, L. L. Cottatellucci, A. Kadavankandy, Spectral properties of random matrices for stochastic block model, in: Proc. IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2015, pp. 537–544.
[42] A. Sandryhaila, J.M.F. Moura, Discrete signal processing on graphs: frequency analysis, IEEE Trans. Signal Process. 62 (12) (2014) 3042–3054.
[43] National Climatic Data Center, 2016. ftp://ftp.ncdc.noaa.gov/pub/data/gsod/.
[44] E. Isufi, A. Loukas, A. Simonetto, G. Leus, Autoregressive moving average graph filtering, IEEE Trans. Signal Process. 65 (2) (2017) 274–288.
[45] J. Miettinen, E. Nitzan, S.A. Vorobyov, E. Ollila, Graph signal processing meets blind source separation, IEEE Trans. Signal Process. 69 (2021) 2585–2599.
[46] D. Clarkson, Remark AS R74: a least squares version of algorithm AS 211: the F-G diagonalization algorithm, Appl. Stat. 37 (1988) 317–321.
[47] A. Belouchrani, K. Abed-Meraim, J.F. Cardoso, E. Moulines, A blind source separation technique using second-order statistics, IEEE Trans. Signal Process. 45 (2) (1997) 434–444.
[48] J. Miettinen, K. Illner, K. Nordhausen, H. Oja, S. Taskinen, F. Theis, Separation of uncorrelated stationary time series using autocovariance matrices, J. Time Ser. Anal. 37 (3) (2016) 337–354.
[49] P. Ilmonen, K. Nordhausen, H. Oja, E. Ollila, A new performance index for ICA: properties computation and asymptotic analysis, in: V. Vigneron, V. Zarzoso, E. Moreau, R. Gribonval, E. Vincent (Eds.), Latent Variable Analysis and Signal Separation, LNCS, 63–65, Springer, Heidelberg, 2010, pp. 229–236.