GRAPH-SIGNAL RECONSTRUCTION AND BLIND DECONVOLUTION FOR DIFFUSED SPARSE INPUTS

David Ramírez[*,⋆], *Antonio G. Marques*[†] *and Santiago Segarra*[‡]

[*]Dept. of Signal Theory and Comms., Universidad Carlos III de Madrid, Leganés, Spain
[⋆]Gregorio Marañón Health Research Institute, Madrid, Spain
[†]Dept. of Signal Theory and Comms., King Juan Carlos University, Madrid, Spain
[‡]Institute for Data, Systems, and Society, Massachusetts Institute of Technology, Cambridge, MA, USA

ABSTRACT

This paper investigates the problems of signal reconstruction and blind deconvolution for graph signals that have been generated by an originally sparse input diffused through the network via the application of a graph filter operator. Assuming that the support of the sparse input signal is unknown, and that the diffused signal is observed only at a subset of nodes, we address the related problems of: 1) identifying the input and 2) interpolating the values of the diffused signal at the non-sampled nodes. We first consider the more tractable case where the coefficients of the diffusing graph filter are known and then address the problem of joint input and filter identification. The corresponding blind identification problems are formulated, novel convex relaxations are discussed, and modifications to incorporate a priori information on the sparse inputs are provided.

***Index Terms*—** Blind signal reconstruction, blind system identification, graph signal processing, sampling and interpolation.

1. INTRODUCTION

Graph signal processing (GSP) generalizes traditional SP algorithms to deal with signals defined on irregular domains represented by graphs [1,2]. Depending on the application, the particular graph may correspond to an actual (social, electrical, sensor) network where the signal is observed, or encode (pairwise) statistical relationships between the signal values. Recent examples of GSP works dealing with a number of relevant problems include sampling and reconstruction of bandlimited signals [3–6], filter design [7, 8], and frequency analysis [9, 10], to name a few.

In this paper we investigate how to generalize blind deconvolution and signal reconstruction to a particular class of graph signals. Formally, consider a graph with N nodes, and suppose that the linear relation $\mathbf{y} = \mathbf{H}\mathbf{x}$ holds, where $\mathbf{y} \in \mathbb{R}^N$ is a *partially observed* graph signal, $\mathbf{H} \in \mathbb{R}^{N\times N}$ is a *linear graph filter*, and $\mathbf{x} \in \mathbb{R}^N$ is an unknown *sparse* input. We are then interested in solving the following problem: Given a) the values of $\mathbf{y}$ at a subset of nodes and b) side information on $\mathbf{H}$ and $\mathbf{x}$, find 1) the unknown sparse input $\mathbf{x}$ and 2) the values of $\mathbf{y}$ at the non-observed nodes. Since graph filters implement local diffusion dynamics [8, 11], such a model is of interest in applications such as opinion formation and source identification in social networks, inverse problems of biological signals supported on graphs, and modeling and estimation of diffusion processes in multi-agent networks. Signals adhering to this generative model will be referred to as *diffused sparse* inputs.

Contributions and related works. We first look at the more favorable setup where the filter $\mathbf{H}$ is known (Sec. 2). The goal there is to reconstruct $\mathbf{y}$ under the assumption that the input $\mathbf{x}$ is sparse. The support of $\mathbf{x}$ is not known and additional information on some of the values of $\mathbf{x}$ may be available or not. This problem falls into the class of sparse signal reconstruction. While a number of GSP works have investigated the reconstruction of bandlimited signals (i.e., assuming that $\mathbf{y}$ belongs to a subspace defined by some of the frequencies of the graph [3–6]), the *subspace* here is given by the filter $\mathbf{H}$. Moreover, our focus is on *blind* setups where the support is not known – see [5, 12] for exceptions dealing with unknown frequency support – and incorporate additional side information on the input. After analyzing that problem, we transition to setups where the coefficients that define the filter $\mathbf{H}$ are also unknown (Sec. 3). In this case we assume that the maximum degree of the filter is known and that some of the values of the input $\mathbf{x}$ may be available. The problem of joint filter and input identification for graph signals was addressed in [11]. The difference here is on the *algorithmic approach* (which yields better results), the incorporation of additional *side information* on the input $\mathbf{x}$, and the interest in reconstructing $\mathbf{y}$.

1.1. Fundamentals of graph signal processing

This section briefly reviews the main concepts in GSP. For more details, we refer the reader to, e.g., [1, 2]. Consider the directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ formed by the set $\mathcal{N}$ of N nodes and the set of links $\mathcal{E}$, such that the pair (i, j) belongs to $\mathcal{E}$ if there exists a link from node i to node j. Associated with a given $\mathcal{G}$, a graph signal is a mapping from $\mathcal{N}$ to $\mathbb{R}$ that can be conveniently represented as a vector $\mathbf{x} = [x_1, \ldots, x_N]^T \in \mathbb{R}^N$, where the ith component, x_i, represents the signal value at node i. To account for the network structure when operating on graph signals, define the graph shift operator $\mathbf{S}$ [9, 13] as a sparse matrix with nonzero values if $(i, j) \in \mathcal{E}$ or $i = j$, that is, $[\mathbf{S}]_{j,i} \neq 0$ for $(i, j) \in \mathcal{E}$ or $i = j$. The multiplication of a graph signal by a shift matrix results in a linear transformation that can be computed locally at the nodes of the graph. Mathematically, to compute the ith element of transformed signal, $y_i = [\mathbf{S}\mathbf{x}]_i$, only the signal values $\{x_j\}_{j\in\mathcal{N}_i}$ are required, where $\mathcal{N}_i = \{j|(j, i) \in \mathcal{E}\}$. Although multiple definitions for the shift matrix have been proposed, the most widely used are the adjacency matrix [9, 13] and the graph Laplacian [1]. Assuming that $\mathbf{S}$ is diagonalizable, the shift matrix can be decomposed as $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, where $\mathbf{\Lambda} \in \mathbb{R}^{N\times N}$ is a diagonal matrix. Based on the shift matrix $\mathbf{S}$, linear graph filters are defined as graph-signal operators of the form

$$\mathbf{H} = \textstyle\sum_{l=0}^{L-1} h_l \mathbf{S}^l, \tag{1}$$

The work of the first author was partly supported by the Spanish MINECO grants OTOSIS (TEC2013-41718-R) and COMONSENS Network (TEC2015-69648-REDC); by the Spanish MINECO and the European Commission (ERDF) grant ADVENTURE (TEC2015-69868-C2-1-R); and by the Comunidad de Madrid grant CASI-CAM-CM (S2013/ICE-2845). The work of the second and third authors was suported by the Spanish MINECO grants OMICRON (TEC2013-41604-R) and KLINILYCS (TEC2016-75361-R).

i.e, polynomials on $\mathbf{S}$ [13]. The filtering operation is thus given by $\mathbf{y} = \mathbf{H}\mathbf{x}$, where $\mathbf{y}$ is the filtered signal, $\mathbf{x}$ the original input, and $\mathbf{h} = [h_0, \ldots, h_{L-1}]^T$ are the filter coefficients, with $L-1$ being the filter degree.

Frequency interpretation. As in classical SP, graph filters and signals may be represented in the frequency (or Fourier) domain. Defining the graph Fourier operator as $\mathbf{U} = \mathbf{V}^{-1}$, the graph Fourier transform (GFT) of the signal $\mathbf{x}$ is $\tilde{\mathbf{x}} = \mathbf{U}\mathbf{x}$. For graph filters, the definition of the GFT that maps $\mathbf{h}$, the filter coefficients, to $\tilde{\mathbf{h}}$, the response of the filter in the frequency domain, is given by $\tilde{\mathbf{h}} = \mathbf{\Psi}\mathbf{h}$, where $\mathbf{\Psi}$ is a $N \times L$ Vandermonde matrix whose elements are $[\mathbf{\Psi}]_{i,j} = [\mathbf{\Lambda}]_{i,i}^{j-1}$ [8]. Note that while in classical SP both Fourier transforms are the same, here $\mathbf{U} \neq \mathbf{\Psi}$. With $\circ$ denoting the Hadamard (elementwise) product, the definitions of the GFT of a signal and a filter allow us to rewrite the filtering operation in the Fourier domain as follows [9]

$$\tilde{\mathbf{y}} = \mathbf{U}\mathbf{y} = \text{diag}\big(\mathbf{\Psi}\mathbf{h}\big)\mathbf{U}\mathbf{x} = \text{diag}\big(\tilde{\mathbf{h}}\big)\tilde{\mathbf{x}} = \tilde{\mathbf{h}} \circ \tilde{\mathbf{x}}. \tag{2}$$

Finally, for the purpose of joint input and filter identification (Sec. 3), the dependence of $\tilde{\mathbf{y}}$ on $\mathbf{x}$ and $\mathbf{h}$ can be alternative written as

$$\tilde{\mathbf{y}} = (\mathbf{\Psi}^T \odot \mathbf{U}^T)^T \text{vec}(\mathbf{x}\mathbf{h}^T), \tag{3}$$

where $\odot$ is the Khatri-Rao, or columnwise Kronecker, product and $\text{vec}(\cdot)$ is the vectorization operator, i.e., the stack of the columns of the matrix input.

1.2. Generation of diffused sparse signals

This paper considers signals generated by the model $\mathbf{y} = \mathbf{H}\mathbf{x}$ with $\mathbf{H} = \sum_{l=0}^{L-1} h_l \mathbf{S}^l$ and $\mathbf{x}$ having at most S non-zero elements. Mathematically, this means that signal $\mathbf{y}$ belongs to a subspace of dimension (at most) S spanned by a subset of the columns of $\mathbf{H}$. The particular subspace depends on the network topology encoded in $\mathbf{S}$, the filter coefficients, and the positions of the non-zero values in $\mathbf{x}$.

To see why this model bears practical relevance, note that graph filters can be used to represent linear diffusion dynamics that depend on the network topology [8, 11]. Formally, define the lth shifted version of the input as $\mathbf{x}^{[l+1]} = \mathbf{S}\mathbf{x}^{[l]}$ and use this to write the output $\mathbf{y}$ of a graph filter as

$$\mathbf{y} = \textstyle\sum_{l=0}^{L-1} h_l \mathbf{x}^{[l]}, \;\; \text{with} \;\; x_i^{[l+1]} = [\mathbf{S}]_{ii} x_i^{[l]} + \sum_{j \in \mathcal{N}_i} [\mathbf{S}]_{ij} x_j^{[l]}. \tag{4}$$

Since each of the shifted inputs can be computed locally, (4) reveals that $\mathbf{y}$ can be viewed as a steady-state signal generated by a *seeding* signal $\mathbf{x}^{[0]} = \mathbf{x}$ that is *diffused locally* by means of the successive application of the network dynamics captured by $\mathbf{S}$. Under this interpretation, the assumption of the input $\mathbf{x}$ being sparse implies that, at the initial state $\mathbf{x}^{[0]}$, only a few nodes have a non-zero value. However, each application of the shift (say the lth one) spreads the information in $\mathbf{x}^{[l-1]}$ across the one-hop neighborhood of the nodes in the (non-zero) support of $\mathbf{x}^{[l-1]}$ [cf. (4)]. Hence, for L sufficiently high (larger than the diameter of the graph) the seeding values in $\mathbf{x}^{[0]}$ will have percolated across the entire network. Potential applications range from social networks where a rumor originated by a small group of people is spread across the network via local opinion exchanges, to brain networks where an epileptic seizure emanating from few regions is later diffused across the entire brain.

2. RECOVERY WITH KNOWN DIFFUSING FILTERS

Provided that the generative model $\mathbf{y} = \mathbf{H}\mathbf{x}$ with $\mathbf{x}$ sparse holds true, the goal in this section is to recover $\mathbf{y}$ and $\mathbf{x}$ from a few samples of $\mathbf{y}$ under the assumption that the shift $\mathbf{S}$ and the filter taps $\mathbf{h}$ are known. Note that this implies that the full matrix $\mathbf{H}$ is known [cf. (1)]. Depending on the application, the interest is in using the sampled values of $\mathbf{y}$ to recover the values of $\mathbf{x}$, identify the support of $\mathbf{x}$, recover the values of $\mathbf{y}$ in the non-observed nodes (interpolation), or any combination thereof.

To describe the problem rigorously, let $\mathbf{C}_{\mathcal{M}} \in \{0,1\}^{M \times N}$ be a *sampling matrix* whose rows correspond to canonical vectors identifying the elements $\mathcal{M} = \{i_1, \ldots, i_M\}$ of the signal $\mathbf{y}$ that are observed, which is then used to define the observed signal as $\bar{\mathbf{y}} = \mathbf{C}_{\mathcal{M}}\mathbf{y} = \mathbf{C}_{\mathcal{M}}\mathbf{H}\mathbf{x}$. To encode the sparsity in $\mathbf{x}$, consider the set $\mathcal{S} = \{j_1, \ldots, j_S\}$ containing the indices of the *unknown* support of $\mathbf{x}$ and the binary matrix $\mathbf{C}_{\mathcal{S}} \in \{0,1\}^{S \times N}$ that, when applied to $\mathbf{x}$, yields the non-zero values associated with the S nodes in $\mathcal{S}$. The blind recovery problem can then be formulated as

$$\begin{aligned} \hat{\mathbf{x}} = \text{find} \quad & \{\mathbf{x}\}, \\ \text{s. to:} \quad & \bar{\mathbf{y}} = \mathbf{C}_{\mathcal{M}}\mathbf{H}\mathbf{x}, \quad \|\mathbf{x}\|_0 \leq S, \end{aligned} \tag{5}$$

which falls into the class of sparse reconstruction; see, e.g., [14]. Once $\hat{\mathbf{x}}$ is identified, the value of the signal $\mathbf{y}$ for the nodes $m \notin \mathcal{M}$ can be found as $[\hat{\mathbf{y}}]_m = [\mathbf{H}\hat{\mathbf{x}}]_m$.

If $M < S$, the recovery is ill posed. This is true, even if the support of $\mathbf{x}$ is known. With known support, the success of the recovery depends on the *rank* (invertibility) of the submatrix $(\mathbf{C}_{\mathcal{M}}\mathbf{H}\mathbf{C}_{\mathcal{S}}^T) \in \mathbb{R}^{M \times S}$, which should be at least S and, hence, requires $M \geq S$ as a necessary condition. When the support is not known, the recovery performance depends on the *spark* of the submatrix $(\mathbf{C}_{\mathcal{M}}\mathbf{H}) \in \mathbb{R}^{M \times N}$. In particular, it holds that if $\mathbf{C}_{\mathcal{M}}\mathbf{H}$ is full spark, then (5) is guaranteed to identify the ground-truth input $\mathbf{x}$ provided that $M \geq 2S$ [5]. Since (5) is not convex, a typical approach to address the optimization is to replace the ℓ_0 norm with the convex ℓ_1 norm [15]. Recovery in this case depends on the *coherence* of the matrix $(\mathbf{C}_{\mathcal{M}}\mathbf{H})$ [5, 15].

In the context of sparse reconstruction, a (typically) better alternative to the minimization of the ℓ_1 norm $\sum_{n=1}^N |x_n|$ is to use $\sum_{n=1}^N \log(|x_n| + \epsilon_0)$, for a small positive constant ϵ_0 [15]. Since minimizing this *concave* surrogate is challenging, the solution to the optimization problem may be found using a majorization-minimization (MM) approach [16]. The main idea is to solve a series of problems, where the concave function (or any nonconvex function) is replaced by a convex approximation. Using a first order Taylor series approximation and with $i = 1, ..., I$ being an iteration index, applying this approach to (5) yields the convex optimization problem

$$\begin{aligned} \hat{\mathbf{x}}^{(i)} := \underset{\mathbf{x}}{\text{argmin}} \quad & \textstyle\sum_{n=1}^N (|\hat{x}_n^{(i-1)}| + \epsilon_0)^{-1} |x_n|, \\ \text{s. to} \quad & \bar{\mathbf{y}} = \mathbf{C}_{\mathcal{M}}\mathbf{H}\mathbf{x}. \end{aligned} \tag{6}$$

We then set the estimated input to $\hat{\mathbf{x}} = \hat{\mathbf{x}}^{(I)}$ and the interpolated output to $\hat{\mathbf{y}} = \mathbf{H}\hat{\mathbf{x}}^{(I)}$. To account for noise in the observations and small model mismatches, the linear constraint in (6) can be relaxed and written as $\|\bar{\mathbf{y}} - \mathbf{C}_{\mathcal{M}}\mathbf{H}\mathbf{x}\|_2 \leq \varepsilon$.

The next step is to address the reconstruction problem when some of the values of the input are known. To that end, let $\mathcal{K} = \{k_1, k_2, \ldots, k_K\}$ represent the set containing the indices of the K nodes with known input values and let $\mathbf{C}_{\mathcal{K}} \in \{0,1\}^{K \times N}$ be a binary matrix that, when applied to $\mathbf{x}$, results in the vector $\mathbf{x}_{\mathcal{K}} = \mathbf{C}_{\mathcal{K}}\mathbf{x}$ containing the K known values. Moreover, with $\mathcal{K}^c$ denoting the complement set of $\mathcal{K}$, $\mathbf{x}_{\mathcal{K}^c}$ is the vector containing the $N-K$ values that are not known, and $\mathbf{C}_{\mathcal{K}^c} \in \{0,1\}^{(N-K) \times N}$ the corresponding selection matrix. Knowledge of $\mathbf{x}_{\mathcal{K}}$ can be based on structural properties of the application at hand (e.g., physiological constraints for particular nodes of a brain network). Alternatively, in multi-agent networks, it is conceivable that the sampling nodes have access not

only to the value of the diffused signal $\mathbf{y}$, but also to their own value of the initial sparse input $\mathbf{x}$ (note that in this case one could have that $\mathcal{K} = \mathcal{S}$). Mathematically, the incorporation of this knowledge into (5) or (6) is straightforward. The simplest option is to augment the problems with a constraint forcing $\mathbf{x}_{\mathcal{K}} = \mathbf{C}_{\mathcal{K}}\mathbf{x}$. If computational complexity is a concern, an alternative is to replace the optimization variable $\mathbf{x}$ with $\mathbf{x}_{\mathcal{K}^c}$, matrix $\mathbf{H}$ with $\mathbf{H}\mathbf{C}_{\mathcal{K}^c}^T$, and the observation $\bar{\mathbf{y}}$ with $\bar{\mathbf{y}} - \mathbf{C}_{\mathcal{M}}\mathbf{H}\mathbf{C}_{\mathcal{K}}^T\mathbf{x}_{\mathcal{K}}$.

Reconstruction of bandlimited signals. The problem presented in this section is closely related to that of recovering a bandlimited graph signal from a limited number of nodal observations [3–6, 12]. In the aforementioned works, the observed signal is assumed to have a *sparse frequency* representation $\|\tilde{\mathbf{y}}\|_0 = \|\mathbf{U}\mathbf{y}\|_0 \leq S$. In other words, the observed signal $\mathbf{y} = \mathbf{V}\tilde{\mathbf{y}}$ is assumed to belong to a subspace spanned by a subset of the eigenvectors of $\mathbf{S}$. Although most works assume that the set of active frequencies is known beforehand, some of them have also investigated the reconstruction under the assumption of unknown support [5, 12]. For bandlimited signals, the M observations in $\bar{\mathbf{y}} = \mathbf{C}_{\mathcal{M}}\mathbf{V}\tilde{\mathbf{y}}$ are used first to estimate the S non-zero frequency coefficients in $\tilde{\mathbf{y}}$. The estimated coefficients $\hat{\tilde{\mathbf{y}}}$ are then used to recover the full signal as $\hat{\mathbf{y}} = \mathbf{V}\hat{\tilde{\mathbf{y}}}$.

The main differences between the observation models given by $\mathbf{y} = \mathbf{V}\tilde{\mathbf{y}}$ with $\|\tilde{\mathbf{y}}\|_0 \leq S$ and $\mathbf{y} = \mathbf{H}\mathbf{x}$ with $\|\mathbf{x}\|_0 \leq S$ are summarized next. First, while for bandlimited signals the subspace is spanned by a subset of the columns of $\mathbf{V}$, for diffused sparse signals the subspace is spanned by a subset of the columns of $\mathbf{H} = \sum_{l=0}^{L-1} h_l\mathbf{S}^l$. Note that although different, both depend on the topology of the graph encoded in $\mathbf{S}$. Second, except for the *smooth* signals associated with a Laplacian shift [1, 3], mechanisms that generate bandlimited graph signals are not yet well understood. Differently, diffused sparse signals have a neat physical interpretation. Third, while for bandlimited signals the estimation of $\tilde{\mathbf{y}}$ is just an intermediate step to reconstruct the full $\mathbf{y}$, in our case finding the sparse signal $\mathbf{x}$ can have practical interest too. Finally, while for diffused sparse signals having access to some values x_k of the input can be reasonable in practice, knowledge of particular non-zero frequency coefficients $\tilde{y}_k \neq 0$ may be more difficult to motivate.

3. RECOVERY WITH UNKNOWN DIFFUSING FILTERS

Given the models $\mathbf{H} = \sum_{l=0}^{L-1} h_l\mathbf{S}^l$ and $\mathbf{y} = \mathbf{H}\mathbf{x}$ with $\mathbf{x}$ sparse, the goal in this section is to recover $\mathbf{y}$ and $\mathbf{x}$ from a few samples of $\mathbf{y}$ under the assumption that only the shift $\mathbf{S}$ and the order of the filter $L-1$ are known. Since the filter coefficients $\mathbf{h}$ (hence, the matrix $\mathbf{H}$) are *not known*, successful recovery requires estimating $\mathbf{h}$ too.

We begin by formulating the problem without considering the side information $\mathbf{x}_{\mathcal{K}}$ and discuss a convex relaxation. Recall first that the sampled signal is defined as $\bar{\mathbf{y}} = \mathbf{C}_{\mathcal{M}}\mathbf{y}$. Moreover, using the results in Sec. 1.1, the dependence of $\mathbf{y}$ on $\mathbf{x}$ and $\mathbf{h}$ given by $\mathbf{y} = \sum_{l=0}^{L-1} h_l\mathbf{S}^l\mathbf{x}$ can be rewritten as $\mathbf{y} = \mathbf{V}\tilde{\mathbf{y}} = \mathbf{V}(\mathbf{\Psi}^T \odot \mathbf{U}^T)^T\text{vec}(\mathbf{x}\mathbf{h}^T)$, where for the first equality we have used the definition of the GFT and for the second the expression in (3). Writing $\mathbf{y}$ in this form is more convenient because it reveals a linear relation between the output $\mathbf{y}$ and the crossproducts in $\text{vec}(\mathbf{x}\mathbf{h}^T)$. With these conventions, the joint blind recovery of the generating filter and sparse input is formulated as

$$\{\hat{\mathbf{x}}, \hat{\mathbf{h}}\} = \text{find} \quad \{\mathbf{x}, \mathbf{h}\}, \tag{7}$$
$$\text{s. to} \quad \bar{\mathbf{y}} = \mathbf{C}_{\mathcal{M}}\mathbf{V}(\mathbf{\Psi}^T \odot \mathbf{U}^T)^T\text{vec}(\mathbf{x}\mathbf{h}^T), \;\; \|\mathbf{x}\|_0 \leq S.$$

This problem requires taking more samples than those for (5), since the number of unknowns here is larger ($L + S$ vs. S). Even for high values of M the problem is hard due to: 1) the non-convexity of the l_0 norm and 2) the *bilinear* constraint in $\mathbf{x}$ and $\mathbf{h}$ (which is non-convex and reveals an inherent scaling ambiguity). In order to devise a tractable relaxation, we first *lift* the problem by defining the $N \times L$ rank-one matrix $\mathbf{Z} = \mathbf{x}\mathbf{h}^T$. Defining the $M \times NL$ observation matrix $\mathbf{M} := \mathbf{C}_{\mathcal{M}}\mathbf{V}\left(\mathbf{\Psi}^T \odot \mathbf{U}^T\right)^T$, the problem in (7) becomes

$$\hat{\mathbf{Z}} = \text{find} \quad \{\mathbf{Z}\}, \tag{8}$$
$$\text{s. to} \quad \bar{\mathbf{y}} = \mathbf{M}\text{vec}\left(\mathbf{Z}\right), \;\; \text{rank}(\mathbf{Z}) = 1, \;\; \|\mathbf{Z}\|_{2,0} \leq S,$$

where $\|\mathbf{Z}\|_{2,0}$, defined as the number of non-zero rows of $\mathbf{Z}$, is equivalent to $\|\mathbf{x}\|_0$ in (7). The vectors $\hat{\mathbf{x}}$ and $\hat{\mathbf{h}}$ are found as the scaled versions of the right and left principal *singular vectors* of the rank-one matrix $\hat{\mathbf{Z}}$, while the output for nodes $m \notin \mathcal{M}$ is found as $[\hat{\mathbf{y}}]_m = [\sum_{l=0}^{L-1} \hat{h}_l\mathbf{S}^l\hat{\mathbf{x}}]_m$.

Since (7) and (8) are equivalent, (8) is still difficult to solve, but as we will see next it leads naturally to a convex relaxation.[1]

3.1. Algorithmic approach

The first step is to reformulate the feasibility problem in (8) as a minimization. Moreover, since S is not always known, the $l_{2,0}$ norm constraint is added to the objective as a penalty term, which yields

$$\hat{\mathbf{Z}} = \underset{\mathbf{Z}}{\text{argmin}} \;\; J(\mathbf{Z}) := \text{rank}(\mathbf{Z}) + \tau\|\mathbf{Z}\|_{2,0},$$
$$\text{s. to} \;\; \bar{\mathbf{y}} = \mathbf{M}\text{vec}\left(\mathbf{Z}\right).$$

The above optimization is non-convex due to the rank and the $l_{2,0}$ norm functions. Instead of using the well-known nuclear norm [18] and l_1 norm [19] convex approximations, as done in [11,17], we rely on the logarithm function to yield a better surrogate [20]. Thus, we may approximate the cost function as

$$J(\mathbf{Z}) \approx \textstyle\sum_{n=1}^{\min(N,L)} \log\left(\sigma_n + \epsilon_1\right) + \tau \sum_{n=1}^{N} \log\left(\|\mathbf{z}_n^T\|_2 + \epsilon_2\right),$$

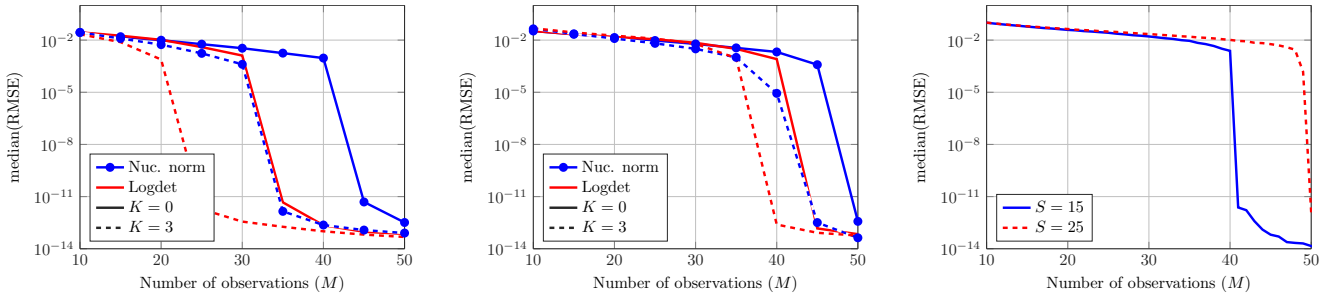
where σ_n is the nth singular value of $\mathbf{Z}$, $\mathbf{z}_n^T$ is the nth row of $\mathbf{Z}$, and ϵ_1 and ϵ_2 are small positive constants. However, the presence of the singular values renders the above minimization not straightforward. Replacing $\sum_{n=1}^{\min(N,L)} \log\left(\sigma_n + \epsilon_1\right)$ with $\log\det(\mathbf{Z} + \epsilon_1\mathbf{I})$ is not possible because $\mathbf{Z}$ is not a square positive definite matrix. To overcome this issue, we rely on the semidefinite embedding lemma [20], which yields

$$\min_{\mathbf{Z}, \mathbf{\Theta}_1, \mathbf{\Theta}_2} \;\; \sum_{j=1}^{2} \log\det\left(\mathbf{\Theta}_j + \epsilon_1\mathbf{I}\right) + \tau \sum_{n=1}^{N} \log\left(\|\mathbf{z}_n^T\|_2 + \epsilon_2\right),$$
$$\text{s. to} \quad \bar{\mathbf{y}} = \mathbf{M}\text{vec}\left(\mathbf{Z}\right), \quad \begin{bmatrix} \mathbf{\Theta}_1 & \mathbf{Z} \\ \mathbf{Z}^T & \mathbf{\Theta}_2 \end{bmatrix} \succeq \mathbf{0}.$$

As in the case of (6), an MM approach is used to deal with the fact of the cost being concave and not convex. Relying again on a first order Taylor series approximation, the problem at the ith iteration is

$$\{\hat{\mathbf{Z}}^{(i)}, \hat{\mathbf{\Theta}}_1^{(i)}, \hat{\mathbf{\Theta}}_2^{(i)}\} := \underset{\mathbf{Z}, \mathbf{\Theta}_1, \mathbf{\Theta}_2}{\text{argmin}} \sum_{j=1}^{2} \text{Tr}\left[\left(\hat{\mathbf{\Theta}}_j^{(i-1)} + \epsilon_1\mathbf{I}\right)^{-1}\mathbf{\Theta}_j\right] \tag{9}$$
$$+ \tau \sum_{n=1}^{N} \frac{\|\mathbf{z}_n^T\|_2}{\|\hat{\mathbf{z}}_n^{(i-1)T}\|_2 + \epsilon_2},$$
$$\text{s. to} \quad \bar{\mathbf{y}} = \mathbf{M}\text{vec}\left(\mathbf{Z}\right), \quad \begin{bmatrix} \mathbf{\Theta}_1 & \mathbf{Z} \\ \mathbf{Z}^T & \mathbf{\Theta}_2 \end{bmatrix} \succeq \mathbf{0}.$$

[1] The problems in (7) and (8) have been recently investigated in [11, 17]. The difference here is on the approach to relax the rank function (logdet vs. nuclear norm), which yields a better recovery performance; the incorporation of a priori information on the input; and the focus on sampling and signal reconstruction vs. system identification.



(a) (b) (c)

Fig. 1: Recovery performance (RMSE) for different number of observations (M) for a setup with: (a) $L = 3$, $S = 8$, and $N = M$; (b) $L = 3$, $S = 8$, and $N = 50$; and (c) $L = 3$, $K = 0$, $N = 50$ and known $\mathbf{H}$.

Once the algorithm has finalized, we need to find the best rank-one approximation of $\hat{\mathbf{Z}} = \hat{\mathbf{Z}}^{(I)}$. This is accomplished by selecting $\hat{\mathbf{x}}$ and $\hat{\mathbf{h}}$ as the scaled *principal singular vectors* of $\hat{\mathbf{Z}}$. As in (6), the linear observation constraint can be relaxed as $\|\bar{\mathbf{y}}-\mathbf{M}\text{vec}(\mathbf{Z})\|_2 \leq \varepsilon$ to account for noise in the observations and model mismatches. The recovery performance of (9) depends on the properties of $\mathbf{M}$, which, we recall, was defined as $\mathbf{M} := \mathbf{C}_{\mathcal{M}}\mathbf{V}\left(\mathbf{\Psi}^T \odot \mathbf{U}^T\right)^T$. Matrix $\mathbf{M}$ depends on both the sampling set $\mathcal{M}$ and the topology of the graph encoded via the eigenvectors (in $\mathbf{V}$ and $\mathbf{U}$) and eigenvalues (in $\mathbf{\Psi}$) of $\mathbf{S}$. As expected, this implies that there are particular graph topologies that result in more amenable recovery problems. An analytical characterization of the recovery performance together with design of optimal sampling schemes is out of the scope of the manuscript and will be reported elsewhere.

3.2. Known input values

This section discusses the changes in (9) to incorporate the fact that some values of the input $\mathbf{x}$ are known. Recall that $\mathbf{x}_{\mathcal{K}}$ collects the known values and $\mathcal{K}$ their associated node indices. If the known values are zero, it suffices to set to zero the corresponding rows of $\mathbf{Z}$ and remove those from the optimization. Hence, in the following we will assume that $\mathbf{x}_{\mathcal{K}}$ collects the known non-zero values. Since $\mathbf{Z} = \mathbf{x}\mathbf{h}^T$, it follows that the rows of the rank-one matrix $\mathbf{Z}$ are proportional, that is, $\mathbf{z}_i^T/x_i = \mathbf{z}_j^T/x_j = \mathbf{h}^T$ for all (i,j). Exploiting this, the optimization in (9) can be augmented with the following set of constraints

$$\mathbf{z}_{k_i}^T x_{k_{i+1}} = \mathbf{z}_{k_{i+1}}^T x_{k_i}, \quad i = 1, \ldots, K-1, \tag{10}$$

which still yields a convex problem. Note that this approach requires $K \geq 2$, i.e, at least two non-zero input values must be known.

As in Sec. 3.1, the last step is to find the best rank-one approximation of the solution $\hat{\mathbf{Z}} = \mathbf{Z}^{(I)}$ generated by the augmented version of (9). The information in $\mathbf{x}_{\mathcal{K}}$ must also be incorporated into this step. To achieve this, write the estimate of the filter coefficients as $\hat{\mathbf{h}} = c_h\check{\mathbf{h}}$, where $\check{\mathbf{h}}$ is a unit-norm vector obtained as the principal eigenvector of $\hat{\mathbf{Z}}^T\hat{\mathbf{Z}}$. The signal $\mathbf{x}$ and the scaling factor c_h are obtained then as the solution to

$$\{\hat{c}_h, \hat{\mathbf{x}}_{\mathcal{K}^c}\} = \underset{c_h, \mathbf{x}_{\mathcal{K}^c}}{\text{argmin}} \left\| \begin{bmatrix} \hat{\mathbf{Z}}_{\mathcal{K}} \\ \hat{\mathbf{Z}}_{\mathcal{K}^c} \end{bmatrix} - c_h \begin{bmatrix} \mathbf{x}_{\mathcal{K}}\check{\mathbf{h}}^T \\ \mathbf{x}_{\mathcal{K}^c}\check{\mathbf{h}}^T \end{bmatrix} \right\|_F^2, \tag{11}$$

where $\hat{\mathbf{Z}}_{\mathcal{K}} := \mathbf{C}_{\mathcal{K}}\hat{\mathbf{Z}}$ and $\hat{\mathbf{Z}}_{\mathcal{K}^c} := \mathbf{C}_{\mathcal{K}^c}\hat{\mathbf{Z}}$. After some algebra, it follows that the solution to (11) is

$$\hat{c}_h = \frac{\mathbf{x}_{\mathcal{K}}^T\hat{\mathbf{Z}}_{\mathcal{K}}\check{\mathbf{h}}}{\|\mathbf{x}_{\mathcal{K}}\|_2^2}, \qquad \hat{\mathbf{x}}_{\mathcal{K}^c} = \frac{1}{\hat{c}_h}\hat{\mathbf{Z}}_{\mathcal{K}^c}\check{\mathbf{h}}. \tag{12}$$

Finally, we must point out that the incorporation of $\mathbf{x}_{\mathcal{K}}$ into the best rank-one approximation of $\hat{\mathbf{Z}}$ can be implemented even for $K = 1$.

4. NUMERICAL RESULTS

Here we illustrate numerically the recovery performance of the proposed algorithms and compare it to that of existing alternatives. The default test case considers an Erdős-Rényi random graph [21] with edge-presence probability $p = 0.1$; uses as shift the adjacency matrix of the obtained graph; and draws the non-zero values of $\mathbf{x}$ and the coefficients $\mathbf{h}$ from a multivariate Gaussian distribution, which are normalized to have unit norm. Three scenarios are simulated, for each of them 1000 realizations of the default case $(\mathcal{G}, \mathbf{x}, \mathbf{h})$ are generated, the recovery performance for each realization is evaluated using the normalized root-mean-squared error defined as $\text{RMSE} = (\|\hat{\mathbf{x}}\check{\mathbf{h}}^T - \mathbf{x}\mathbf{h}^T\|_F)/((N-K)L)$, and the median value across the 1000 realizations is reported. In the first scenario, the sparsity of the input is set to $S = 8$, the number of filter coefficients to $L = 3$, the number M of observations varies from 10 to 50, and the size of the network is set to $N = M$. Recovery was performed using the algorithm in (9)-(12) and that in [11], which is based on the nuclear norm surrogate. As can be seen in Fig. 1a, for the recovery problem considered in this paper, the logdet surrogate outperforms the alternative based on the nuclear norm with equivalent computational complexity. This figure shows that the performance gap is very small when the number of observations is small ($M < 15$) and then it grows for higher values of M. We also observe that the additional information provided by as few as $K = 3$ known input values provides a substantial advantage, one of the reasons being that the matrices $\hat{\mathbf{Z}}$ generated by (9) have smaller ranks. Fig. 1b shows the results for a scenario like the one in Fig. 1a, but with a fixed number of nodes $N = 50$ and varying M. While the results validate the previous findings, the error grows larger due to the changes in the structure of the (subsampled) observation matrix. Finally, Fig. 1c shows the recovery performance of (6) when $\mathbf{H}$ is known, and S is set to $S = 15$ or $S = 25$. As expected, with known $\mathbf{h}$ and enough observations we achieve perfect recovery of $\mathbf{x}$ even for large values of S.

5. CONCLUSIONS

This paper presented algorithms for blind recovery of graph signals from partial observations of diffused sparse inputs. The proposed techniques incorporated a priori information on the sparse input and addressed two scenarios of interest: one where the filter was known and another one where only the order of this filter was available. The resultant non-convex recovery problems were relaxed using the log as surrogate of the zero-norm and the logdet as surrogate of the rank. Preliminary numerical results demonstrated the superiority of the proposed approach relative to existing alternatives, and illustrated that incorporating additional side information improves the performance substantially. Ongoing work includes analytical characterization of the recovery performance for particular types of filters, as well as optimal design of schemes to select the sampling nodes.

6. REFERENCES

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

[2] A. Sandryhaila and J. M. F. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 80–90, 2014.

[3] A. Anis, A. Gadde, and A. Ortega, "Towards a sampling theorem for signals on arbitrary graphs," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, 2014, pp. 3864–3868.

[4] S. Chen, R. Varma, A. Sandryhaila, and J. Kovacević, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6510–6523, Dec. 2015.

[5] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Sampling of graph signals with successive local aggregations," *IEEE Trans. Signal Process.*, vol. 64, no. 7, pp. 1832 – 1843, Apr. 2016.

[6] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, "Signals on graphs: Uncertainty principle and sampling," *arXiv preprint arXiv:1507.08822*, 2015.

[7] A. Loukas, A. Simonetto, and G. Leus, "Distributed autoregressive moving average graph filters," *IEEE Signal Process. Lett.*, vol. 22, no. 11, pp. 1931–1935, 2015.

[8] S. Segarra, A. G. Marques, and A. Ribeiro, "Distributed network operators using graph filters," *arXiv preprint arXiv:1510.03947*, 2015.

[9] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, Jun. 2014.

[10] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *arXiv preprint arXiv:1603.04667*, 2016.

[11] S. Segarra, G. Mateos, A. G. Marques, and A. Ribeiro, "Blind identification of graph filters," *arXiv preprint arXiv:1604.07234*, 2016.

[12] R. Varma, S. Chen, and J. Kovacević, "Spectrum-blind signal recovery on graphs," in *IEEE Intl. Wrksp. Computat. Advances Multi-Sensor Adaptive Process. (CAMSAP)*, Cancun, Mexico, Dec. 2015, pp. 81–84.

[13] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[14] R. G. Baraniuk, "Compressive sensing," *IEEE Signal Process. Mag.*, vol. 24, no. Jul., pp. 118–121, 2007.

[15] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted l_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, pp. 877–905, 2008.

[16] B. R. Marks and G. P. Wright, "A general inner approximation algorithm for nonconvex mathematical programs," *Operations Research*, vol. 26, no. 4, pp. 681–683, 1978.

[17] S. Segarra, G. Mateos, A. G. Marques, and A. Ribeiro, "Blind identification of graph filters with sparse inputs," in *IEEE Intl. Wrksp. Computat. Advances Multi-Sensor Adaptive Process. (CAMSAP)*, 2015, pp. 449–452.

[18] M. Fazel, H. Hindi, and S. P. Boyd, "A rank minimization heuristic with application to minimum order system approximation," in *American Control Conf.*, 2001, pp. 4734–4739.

[19] J. A. Tropp, "Just relax: Convex programming methods for identifying sparse signals in noise," *IEEE Trans. Info. Theory*, vol. 52, no. 3, pp. 1030–1051, 2006.

[20] M. Fazel, H. Hindi, and S. P. Boyd, "Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices," in *American Control Conf.*, 2003.

[21] B. Bollobás, *Random Graphs*, Springer, 1998.