SPARSEST NETWORK SUPPORT ESTIMATION: A SUBMODULAR APPROACH

Mario Coutino, Sundeep Prabhakar Chepuri, Geert Leus

Delft University of Technology

ABSTRACT

In this work, we address the problem of identifying the underlying network structure of data. Different from other approaches, which are mainly based on convex relaxations of an integer problem, here we take a distinct route relying on algebraic properties of a matrix representation of the network. By describing what we call possible ambiguities on the network topology, we proceed to employ submodular analysis techniques for retrieving the network support, i.e., network edges. To achieve this we only make use of the network modes derived from the data. Numerical examples showcase the effectiveness of the proposed algorithm in recovering the support of sparse networks.

***Index Terms*—** Graph signal processing, graph learning, sparse graphs, network deconvolution, network topology inference.

1. INTRODUCTION

In recent years, large efforts have been made to understand how traditional tools from signal processing can be adapted for cases where the acquired data is not defined over a regular domain but over a network [1]. This surge of interest is due to the fact that such network-supported signals are able to model complex transport networks [2], the activity of the brain [3], epidemic dynamics and gene coexpression [4], to name a few. As a result, the field of graph signal processing (GSP) has emerged [5, 6].

Assuming the topology of the network which supports a signal directly influences the behaviour of it, GSP develops algorithms which exploit the network structure to perform classical signal processing tasks such as estimation, detection and filtering [7–9]. As a result, appropriate knowledge of the *relations* (edges) among the elements (nodes) of the network (graph) is required for any algorithm that requires connectivity information. Therefore, in this paper, we focus on the problem of *estimating the underlying network structure* in which the data is embedded.

Several works have been devoted to find the underlying network (graph) structure within the data [10–17]. Some of those works fit the observed data to particular structural models [12, 13], or find a network which renders the observed signal *smooth* in its topology [17]. In addition, similar to the approach presented here, there are works [10] that use the network modes to retrieve a sparse network support by means of a convex problem. However, despite the fact that those works successfully address the problem of network estimation, each of them making different structural assumptions, they do not completely address the characterization of general network matrices, i.e., networks with disconnected components, networks with self loops, etc. Moreover, even though that for simple graphs (networks), [10] provides theoretical guarantees for the recovery of the adjacency and normalized Laplacian matrix, in the general case, due to the convex formulation of the recovery problem, solutions that only retrieve the support approximately are obtained, i.e., thresholding operations are required.

Taking this issue into account, and following arguments closely related to the ones used for network topology estimation through spectral templates [10], the main aim of this work is twofold: (i) to provide a proper characterization of a particular set of matrices that are of high importance within graph signal processing: *fixed-support matrices with the same eigenbasis*, and (ii) to provide a pure combinatorial algorithm, based on the submodular machinery that has been proven useful in subset selection problems within signal processing [18–22], as an alternative to traditional convex relaxations.

2. PRELIMINARIES

Consider a network being represented by an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, which consists of a finite set of nodes $\mathcal{V}$ with $|\mathcal{V}| = N$ and a set of edges (connections) $\mathcal{E}$ with $|\mathcal{E}| = M$. If there is a connection in the network between nodes i and j, then $(i, j) \in \mathcal{E}$. A signal or function $f : \mathcal{V} \to \mathbb{R}$ defined on the nodes of the network can be collected in a length-N vector $\boldsymbol{x}$, where the nth element of $\boldsymbol{x}$ represents the function value at the nth node in $\mathcal{V}$. Since $\boldsymbol{x}$ resides on the graph (network), we refer to the function $\boldsymbol{x}$ as a *graph signal*.

Let us introduce a *network topology matrix* $\boldsymbol{S} \in \mathbb{R}^{N\times N}$, where the (i, j)th entry of $\boldsymbol{S}$ denoted by $s_{i,j}$ can only be nonzero if $(i, j) \in \mathcal{E}$. This matrix acts as a possible representation of the network connectivity. Within the field of GSP, this matrix is usually referred to as *graph shift operator* [6].

For undirected graphs, $\boldsymbol{S}$ is symmetric, and thus it admits the following eigenvalue decomposition

$$\boldsymbol{S} \;=\; \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T \tag{1}$$

where the eigenvectors $\boldsymbol{U}$ and the eigenvalues $\boldsymbol{\Lambda}$ of $\boldsymbol{S}$ provide a notion of frequency in the graph setting [5,6]. Specifically, $\boldsymbol{U}$ forms an orthonormal Fourier-like basis for graph signals with the graph frequencies denoted by $\{\lambda_n\}_{n=1}^N$.

For a general network matrix, $\boldsymbol{S}$, that does not admit an orthonormal eigenvalue decomposition, we might consider a decomposition through its singular values. Although, the SVD decomposition does not carry a graph frequency interpretation as in the case of the eigenvalue decomposition, the theory of network support estimation presented in this work holds for both decompositions.

In this work, we aim to reconstruct the support of the network $\boldsymbol{S}$ by means of a set of output data, i.e., $\{\boldsymbol{y}_i\}_{i=1}^Q$, under the assumption that the matrix representation of the network, $\boldsymbol{S}$, and the covariance matrix of the data, $\boldsymbol{R}_y \triangleq \mathbb{E}\{\boldsymbol{y}\boldsymbol{y}^T\}$, share the same eigenspace [23, 24]. Therefore, the only side information used for this purpose are the so called *modes of the network*.

This work is part of the ASPIRE project (project 14926 within the STW OTP program), which is financed by the Netherlands Organization for Scientific Research (NWO). Mario Coutino is partially supported by CONACYT.

3. THE AMBIGUOUS CLASS OF NETWORK MATRICES

Following the assumption that $\boldsymbol{R_y}$ and $\boldsymbol{S}$ share the same eigenbasis, we have the following property:

$$\boldsymbol{R_y S} = \boldsymbol{S R_y}. \tag{2}$$

Now, let us assume that from the set of available data, $\{\boldsymbol{y}_i\}_{i=1}^{Q}$, both the covariance matrix, $\boldsymbol{R_y}$, and its eigenvalue decomposition, $\boldsymbol{U\Omega U}^T$, are obtained. Similar to the spectral templates approach [10], we know that $\boldsymbol{S} = \boldsymbol{U\Lambda U}^T$, for some diagonal matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{N\times N}$. Hence, the support identification problem is reduced to finding a matrix $\boldsymbol{\Lambda}$ which meets the desired properties for a particular choice of network matrix, e.g., integer-valued matrix, nonnegativity, zero diagonal, etc. That is,

$$\underset{\boldsymbol{\Lambda}\in\mathcal{D}}{\text{minimize}} \quad \|\boldsymbol{S}\|_0 \qquad \text{subject to } \boldsymbol{S} = \boldsymbol{U\Lambda U}^T \ , \tag{3}$$

where $\mathcal{D}$ is the set of permissible matrices, i.e., constraint set.

Differently from [10], here we are interested in a *generative* solution based on *algebraic* properties of fixed-support sparse matrices. This approach is motivated from the fact that for a particular problem instance (nature of the data) there is no clear distinction between network matrices with the same support sharing the same eigenbasis. That is, given

$$\boldsymbol{S}_i = \boldsymbol{U\Lambda}_i\boldsymbol{U}^T \text{ and } \boldsymbol{S}_j = \boldsymbol{U\Lambda}_j\boldsymbol{U}^T \neq \boldsymbol{S}_i, \tag{4}$$

where $\boldsymbol{\Lambda}_i$ and $\boldsymbol{\Lambda}_j$ are diagonal matrices, and $\boldsymbol{S}_i$ and $\boldsymbol{S}_j$ have the same support, *which network matrix should be employed?*

To tackle this problem, first we have to understand the class of matrices that falls within this *ambiguous* family, i.e., the family of jointly diagonalizable matrices sharing the same sparsity support. This family of matrices is formally defined as the set

$$\mathcal{J}_{\boldsymbol{U}}^{\mathcal{A}} = \{\boldsymbol{S} : \boldsymbol{S} = \boldsymbol{U\Lambda U}^T \text{ and } [\text{vec}(\boldsymbol{S})]_i = 0,\ \forall\ i \in \mathcal{A}\}, \tag{5}$$

where $\mathcal{A}$ and $\boldsymbol{U}$ are the index set defining the zero entries of the matrix and the eigenbasis, respectively. In the following section, we characterize this set of matrices to leverage their properties for support identification.

4. NETWOK SUPPORT NULLSPACE PROPERTY

Recall that from the definition of the set $\mathcal{J}_{\boldsymbol{U}}^{\mathcal{A}}$ [cf. (5)] we have the following property:

$$[\text{vec}(\boldsymbol{S})]_i = 0,\ \forall i \in \mathcal{A}. \tag{6}$$

This condition can be expressed through a linear equality of the form

$$\boldsymbol{\Phi}_{\mathcal{A}}\text{vec}(\boldsymbol{S}) = \boldsymbol{0}, \tag{7}$$

where $\boldsymbol{\Phi}_{\mathcal{A}} \in \{0,1\}^{|\mathcal{A}|\times N^2}$ is the selection matrix whose rows are the rows of an $N^2 \times N^2$ identity matrix indexed by the set $\mathcal{A} \subseteq \mathcal{N}$, with $\mathcal{N}$ denoting the set of indices of the entries of an $N \times N$ matrix. Furthermore, using the vectorization operation we obtain the relation

$$\boldsymbol{\Phi}_{\mathcal{A}}\text{vec}(\boldsymbol{S}) = \boldsymbol{\Phi}_{\mathcal{A}}(\boldsymbol{U}\otimes\boldsymbol{U})\text{vec}(\boldsymbol{\Lambda}) = \boldsymbol{0}, \tag{8}$$

where $\otimes$ denotes the Kronecker product.

As the matrix $\boldsymbol{\Lambda}$ is a diagonal matrix, we can equivalently write this condition using the Khatri-Rao product ($*$), i.e.,

$$\boldsymbol{\Phi}_{\mathcal{A}}\text{vec}(\boldsymbol{S}) = \boldsymbol{\Phi}_{\mathcal{A}}(\boldsymbol{U} * \boldsymbol{U})\boldsymbol{\lambda} = \boldsymbol{0}, \tag{9}$$

where $\boldsymbol{\lambda} = \text{diag}\{\boldsymbol{\Lambda}\}$ is the eigenvalue vector. From this relation, it is seen that the eigenvalue vector should lie in the intersection of the nullspace of $\boldsymbol{\Phi}_{\mathcal{A}}$ and the range of $\boldsymbol{U} * \boldsymbol{U}$. This statement is equivalent to the following condition:

$$\boldsymbol{\lambda} \in \text{null}\{\boldsymbol{T}_{\boldsymbol{U}}^{\mathcal{A}}\}, \tag{10}$$

where we have defined $\boldsymbol{T}_{\boldsymbol{U}}^{\mathcal{A}} = \boldsymbol{\Phi}_{\mathcal{A}}(\boldsymbol{U} * \boldsymbol{U}) \in \mathbb{R}^{|\mathcal{A}|\times N}$.

Alternatively, we can consider another formulation for the condition expressed in (8) in terms of the covariance matrix $\boldsymbol{R_y}$ and its inverse. That is, considering [cf. (2)]

$$\boldsymbol{S} = \boldsymbol{R}^{-1}\boldsymbol{SR} \tag{11}$$

and applying the vectorization operation to (11), we obtain

$$[\boldsymbol{I} - (\boldsymbol{R_y}\otimes\boldsymbol{R_y}^{-1})]\boldsymbol{\Phi}_{\mathcal{A}^c}^T\boldsymbol{\Phi}_{\mathcal{A}^c}\text{vec}(\boldsymbol{S}) = \boldsymbol{0}. \tag{12}$$

where $\mathcal{A}^c$ is the set that indexes the nonzero entries of $\text{vec}(\boldsymbol{S})$. Note that while condition (10) characterizes the matrices in terms of *missing links* and leverages the spectral decomposition of $\boldsymbol{S}$, the condition (12) characterizes more general kind of matrices, i.e., not eigendecomposable matrices, in terms of *connections between nodes*.

Using the previous results [cf. (6)-(12)], we can summarize the main result of this section in the following theorem:

Theorem 1. (Nullspace Property)
Given an orthonormal basis $\boldsymbol{U}$, and a sparsity pattern defined by a set $\mathcal{A}$, the matrices within the set $\mathcal{J}_{\boldsymbol{U}}^{\mathcal{A}}$ are the matrices of the form $\boldsymbol{S} = \boldsymbol{U\Lambda U}^T$ whose eigenvalues are given by

$$\boldsymbol{\Lambda} = \text{diag}\{\boldsymbol{B}_{\boldsymbol{U}}^{\mathcal{A}}\boldsymbol{\alpha}\}. \tag{13}$$

Here, the matrix $\boldsymbol{B}_{\boldsymbol{U}}^{\mathcal{A}} \in \mathbb{R}^{N\times d}$ is a basis for the nullspace of $\boldsymbol{T}_{\boldsymbol{U}}^{\mathcal{A}}$, i.e.,

$$\text{span}\{\boldsymbol{B}_{\boldsymbol{U}}^{\mathcal{A}}\} = \text{null}\{\boldsymbol{T}_{\boldsymbol{U}}^{\mathcal{A}}\},$$

and $\boldsymbol{\alpha} \in \mathbb{R}^d$ is the expansion coefficient vector.

Proof. The proof directly follows from the expressions in (6)-(10). ■

The result of the theorem provides a certificate of uniqueness of the network matrix in the following sense: the network matrix is considered *unique*, for a fixed sparsity pattern $\mathcal{A}$, if the set $\mathcal{J}_{\boldsymbol{U}}^{\mathcal{A}}$ has elements given by

$$\boldsymbol{S} = \alpha\boldsymbol{S}_0, \tag{14}$$

for some matrix $\boldsymbol{S}_0 = \boldsymbol{U\Lambda}_0\boldsymbol{U}^T$. That is, the elements of the set are scaled versions of each other. This property can be stated formally through the following proposition:

Proposition 1. *(*Uniqueness of the Symmetric Network Matrix*)*
*Given an orthogonal basis $\boldsymbol{U}$ and a sparsity pattern defined by set $\mathcal{A}$, we say that the network matrix is **unique** (up to a scaling ambiguity [cf.* (14)*]) if and only if*

$$d \triangleq \dim(\text{null}\{\boldsymbol{T}_{\boldsymbol{U}}^{\mathcal{A}}\}) = 1.$$

*Otherwise, for $d > 1$ we say that there are **infinite** network matrices sharing the same support and being diagonalizable by the same basis. For cases with $d = 0$, we say that no matrix with support $\mathcal{A}$ and diagonalizable by $\boldsymbol{U}$ exists.*

Proof. The proof is due to (13). ■

A similar statement can be made with respect to condition (12). In the following proposition, we make this result more precise.

Proposition 2. *(*Uniqueness of the General Network Matrix*)*
*Given a covariance matrix $\boldsymbol{R_y}$ and a sparsity pattern defined by set $\mathcal{A}$, we say that the network matrix is **unique** if and only if*

$$\dim(\text{null}\{[\boldsymbol{I} - \boldsymbol{R_y}\otimes\boldsymbol{R_y}^{-1}]\boldsymbol{\Phi}_{\mathcal{A}^c}^T\} = 1. \tag{15}$$

Proof. The proof follows from (12). ■

Algorithm 1: ITERATED GREEDY ALGORITHM [26]

Data: $\mathcal{A}_0, \mathcal{N}, B$ and submodular functions $f(\cdot),\ g(\cdot)$
Result: $\mathcal{A}$
initialization $\mathcal{A} = \mathcal{A}_0$;
for $t = 1, 2, \cdots, T$ **do**
  Choose surrogate modular functions $\hat{f}_t$ and $\hat{g}_t$ for f and g respectively, tight at $\mathcal{A}_{t-1}$;
  $\mathcal{A}_t \leftarrow$ Greedy optimizer of (18) with $\hat{f}_t$ and $\hat{g}_t$ instead of f and g.
end
$\mathcal{A} \leftarrow \mathcal{A}_t$

As in this work we restrict ourselves to symmetric network matrices, the task of retrieving the support of the network from knowledge of the eigenbasis $\boldsymbol{U}$ reduces to finding a set $\mathcal{A}$ with the largest cardinality such that the condition $d > 0$ is met. In the following, we introduce a greedy approach based on submodularity to retrieve the support of the sparsest network with eigenmodes $\boldsymbol{U}$.

5. NETWORK SUPPORT ESTIMATOR

Using the results from the previous section, we can rewrite the optimization problem in (3) using the parametrization in terms of the nullspace of $\boldsymbol{T}_{\boldsymbol{U}}^{\mathcal{A}}$ as follows

$$(\mathcal{A}^*,\ \boldsymbol{\alpha}^*) \in \arg\max_{\boldsymbol{\alpha}} |\mathcal{A}| \qquad \text{subject to } \ \boldsymbol{\lambda} = \boldsymbol{B}_{\boldsymbol{U}}^{\mathcal{A}}\boldsymbol{\alpha}, \tag{16}$$

where $|\mathcal{A}|$ denotes the cardinality of the set $\mathcal{A}$. Note that under this formulation without assuming any constraints on $\mathcal{D}$ it can be seen that there is no unique solution to the topology identification problem if it is not regularized in a particular way. Therefore, instead of trying to estimate the *value of the entries* of the non-zero elements, we focus on another reasonable problem: *the estimation of a set of matrices $\mathcal{J}_{\boldsymbol{U}}^{\mathcal{A}}$, whose sparsity pattern indexed by $\mathcal{A}$ has the largest cardinality and guarantees $d > 0$.*

Mathematically, this can written as

$$\mathcal{A}^* \in \max_{\mathcal{A} \subseteq \mathcal{N}} |\mathcal{A}| \qquad \text{subject to } \ \operatorname{rank}(\boldsymbol{T}_{\boldsymbol{U}}^{\mathcal{A}}) \leq N - 1, \tag{17}$$

Differently from typical convex methods for solving the support identification problem, here the feasible set is described by a constraint involving the rank of $\boldsymbol{T}_{\boldsymbol{U}}^{\mathcal{A}}$.

5.1. Submodular Optimization

Observing (17) we notice that it has a special structure. That is, it is the maximization of a submodular set function, with a submodular constraint, particularly a *budget constraint*, i.e.,

$$\max_{\mathcal{A} \subseteq \mathcal{N}} \quad g(\mathcal{A}) \qquad \text{subject to } \ f(\mathcal{A}) \leq B \ , \tag{18}$$

where $g(\mathcal{A})$ and $f(\mathcal{A})$ are normalized submodular set functions, i.e., $f(\emptyset) = g(\emptyset) = 0$, and B is a *knapsack* budget. This is due to the fact that the cost set function is the cardinality of a set which is a modular set function, and the rank of $\boldsymbol{T}_{\boldsymbol{U}}^{\mathcal{A}}$ is the rank function of a *linear matroid* [25], which is a submodular function.

This kind of optimization problems are known in the literature as submodular constrained submodular knapsack (SCSK) problems. This family of problems has near-to-optimal guarantees through a series of bi-approximation algorithms [26]. Furthermore, they share characteristics with other kinds of common submodular optimization problems such as the submodular constrained submodular coverage (SCSC) problem, and the minimization of the difference of submodular functions [27]. A general form of a bi-approximation algorithm to solve problem (18) is given in Algorithm 1 [26]. This algorithm

Algorithm 2: NETWORK SUPPORT KNAPSACK PURSUIT

Data: $\mathcal{A}_0, \mathcal{N}, B$ and submodular functions $f(\cdot),\ g(\cdot)$
Result: $\mathcal{A}$
initialization $\mathcal{A} = \mathcal{A}_0$;
while feasible **do**
  $\mathcal{I} = \arg\min_{i \in \mathcal{N}} f(\mathcal{A} \cup i)$;
  if $\exists\, i^* \in \mathcal{I} : f(\mathcal{A} \cup i^*) \leq B$ **then**
    $\mathcal{A} \leftarrow \mathcal{A} \cup i^*$;
    $\mathcal{N} \leftarrow \mathcal{N} \setminus i^*$;
  else
    return $\mathcal{A}$;
  end
end

as input only requires: the ground set $\mathcal{N}$, the maximum budget B and an initial set $\mathcal{A}_0$ for obtaining the local tight bounds. In many instances, $\mathcal{A}_0$ can even be the empty set.

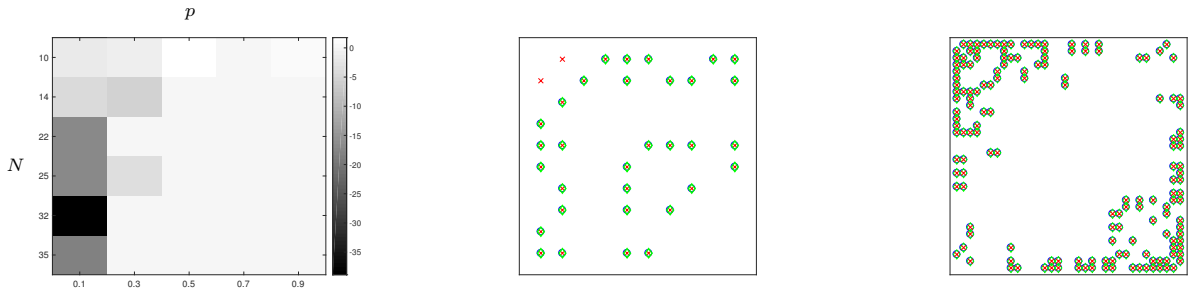
Typically, the common surrogate functions employed for Algorithm 1 are obtained through modular upper and lower bounds [27] that are analogous to the bounds used in majorization/minimization algorithms within the convex optimization literature [28]. Alternatively, for the particular case of our problem: modular cost and a saturated submodular set function (submodular set function with a total curvature equal to one [26]) as a constraint, instead of using a tight modular upper bound as suggested in Algorithm 1, we can use a property of our constraint to devise a greedy algorithm that generates a *feasible path*.

As the rank function is an integer saturated set function, we notice that any upper bound $\hat{f}_t$ will lead to an integer set function which either increases by one or not increases at all when a new element is added to a fixed set ($\hat{f}_t$ is a modular set function). Therefore, when solving the inner optimization problem of Algorithm 1 in most of the instances only one new element is added to the set $\mathcal{A}_{t-1}$ due to the possibility of constraint violation. Notice, that due to the fact that the set $\mathcal{A}_{t-1}$ does not increase the knapsack *weight*, this set is always chosen as part of the solution $\mathcal{A}_t$, i.e., its elements increase the cost without adding any cost. Hence, using this two properties we can obtain a simplified version of Algorithm 1 which at every step is guaranteed to increase the cost set function value while keeping the solution set feasible. Here, by feasible path we refer to a set of incremental solutions, defined by inclusion-wise closure, such that each element of these set is a feasible element, i.e., sparse fixed-support matrix diagonalizable with the selected basis $\boldsymbol{U}$. This procedure is summarized in Algorithm 2. As the network matrix is assumed to be a symmetric matrix, note that the elements of $\mathcal{A}$ can be treated as pairs (off diagonal entries) and singlenton elements (diagonal entries). The greedy approach here presented is akin to the simplification that can be obtained when an ellipsoidal approximation [29] is used to obtain a surrogate function for f when its total curvature is equal to one, however there is no overhead due to computing the approximation $\hat{f}$.

5.2. Adding Extra Constraints

In many cases, we have some information with respect to our network connections, e.g., some edges are known, there are no self-loops, etc., or some properties of the matrix, e.g., the constant eigenvector has a zero as eigenvalue, hollow matrix, etc. Hence, in the following we present examples of how to include such constraints in the proposed framework for network identification.

First, we discuss the most straightforward constraint: *edge*



(a) Erdõs-Rényi (b) Small Network Example (c) Zachary's Karate Network

Fig. 1: (a) Difference of nnz elements between the recovered sparsity pattern and the nnz elements of the true adjacency matrix. (b) Recovered network support comparison. (c) Recovered Zachary's Karate club network comparison.

knowledge. This constraint is added by either subtracting from the ground set $\mathcal{N}$ a fixed set $\mathcal{M}$ (known nonzero entries of $\boldsymbol{S}$) making them ineligible for the algorithm, or by adding a fixed set of elements to the starting set $\mathcal{A}_0$ (knowledge of disconnected nodes). This last case is applicable when we are identifying a hollow matrix for example (adjacency matrix).

Now, let us consider another common constraint: *bandlimitness of the network matrix*. Mathematically, this constraint implies that a subset of the eigenvectors of the network matrix are linked to eigenvalues with zero values. A typical instance for constraints of this kind is, for example, when the desired network matrix has its constant eigenvector linked to a zero eigenvalue, e.g., Laplacian matrix. For this case, the following condition must hold:

$$(\boldsymbol{a} * \boldsymbol{U})\boldsymbol{\lambda} = \boldsymbol{0}, \tag{19}$$

where $\boldsymbol{a} = \boldsymbol{1}^T\boldsymbol{U} = [N, 0, \ldots, 0]^T$ as it is assumed that $\boldsymbol{1}$ is an eigenvector of the network matrix. Condition (19) implies that $\boldsymbol{\lambda}$ lies in the intersection of $\text{null}\{\boldsymbol{T}_{\boldsymbol{U}}^{\mathcal{A}}\}$ and $\text{null}\{\boldsymbol{a} * \boldsymbol{U}\}$ which is tantamount to saying that

$$\boldsymbol{\lambda} \in \text{null}\{\tilde{\boldsymbol{T}}_{\boldsymbol{U}}^{\mathcal{A}}\}. \tag{20}$$

Here, $\tilde{\boldsymbol{T}}_{\boldsymbol{U}}^{\mathcal{A}}$ is the matrix resulting from stacking the matrices $\boldsymbol{T}_{\boldsymbol{U}}^{\mathcal{A}}$ and $(\boldsymbol{a} * \boldsymbol{U})$ as $[(\boldsymbol{T}_{\boldsymbol{U}}^{\mathcal{A}})^T, (\boldsymbol{a} * \boldsymbol{U})^T]$. As a result, we can substitute the matrix $\boldsymbol{T}_{\boldsymbol{U}}^{\mathcal{A}}$ in the rank constraint by $\tilde{\boldsymbol{T}}_{\boldsymbol{U}}^{\mathcal{A}}$ and apply Algorithm 2.

In general, other types of constraints that are expressible as matroids [30], e.g., node partitions, node degree constraints, to name a few, can be included in the presented framework. However, different from the previous constraints, the introduction of such structures comes with a degradation in the approximation quality and computational complexity (an oracle for the independent sets is needed).

6. NUMERICAL RESULTS

In this section, we present a series of numerical simulations in order to demonstrate the developed theory and the proposed method for network support estimation. First, we illustrate the performance of the proposed method for recovering the sparsest symmetric matrix sharing a particular eigenbasis. To do so, we generate a series of Monte Carlo simulations involving Erdõs-Rényi [31] graphs of varying size. For these simulations we select graphs of sizes $N \in \{10, 14, 22, 25, 32, 35\}$ and with different edge-formation probabilities $p \in \{0.1, 0.3, \ldots, 0.9\}$. For each pair (N, p) we generate 100 Erdõs-Rényi graphs and aim to recover the sparsest matrix which shares the eigenbasis with the adjacency matrix, $\boldsymbol{A}$, of the generated graphs. The goal is not to necessarily obtain the original adjacency matrix but the matrix with the largest possible quantity of zero entries such that the eigenspace is given by the one of the true adjacency matrix. Here, the results of this experiment, shown in Fig. 1a, are reported in terms of the difference of the number of nonzero (nnz) elements between the sparsity pattern recovered by the proposed method, i.e., $\text{nnz}(\boldsymbol{S}^*) = N^2 - |\mathcal{A}|$ and the number of nnz elements of the adjacency matrix $\boldsymbol{A}$. From Fig. 1a, we notice that for low p-values the proposed method is able to retrieve *valid network matrices* with a sparsity level greater than the one of the original adjacency matrix $\boldsymbol{A}$. However, as the p-value increases (higher connectivity in the graphs), the eigenbasis becomes *more selective* leading to the recovery of sparsity patterns with the same number of nonzeros as the true $\boldsymbol{A}$. In these instances, as the sparsity pattern results to be unique, the basis that is recovered, $\boldsymbol{B}_{\boldsymbol{U}}^{\mathcal{A}^*}$, has right dimension one. Hence, this gives the certificate that the adjacency matrix is the *unique network matrix*. These results are consistent with the results shown in [10], where they show that through their convex formulation it is possible to retrieve with higher accuracy the adjacency matrix for intermediate p-values. Note that our method can recover the sparsity pattern of the sparsest network matrix for small p-values . Now, we consider two particular instances of unique network matrix recovery. For this, we use an Erdõs-Rényi network with parameters $(N, 0.9)$ and Zachary's Karate club network [32] for showing the performance of our method. The latter network consists of 34 nodes representing members of the club and 78 undirected edges representing the relationships among members. For this case, we consider that the network modes, i.e., eigenvectors of the network matrix, are known. For this example, we obtain the modes of the network from the eigendecomposition of the adjacency matrix. The results of applying Algorithm 2 (green diamonds) and the method in [10] (red crosses) are shown in Figs. 1b-1c. In Fig. 1b, it can be seen that perfect recovery of the support is obtained for the knapsack pursuit method where [10] overestimates the support. This is due to the fact that for this network its associate network matrix is not unique, and the convex method fails to find the appropriate support. In contrast, for Zachary's network the dimension of the nullspace of the matrix $\boldsymbol{T}_{\boldsymbol{U}}^{\mathcal{A}}$ is $d = 1$, which implies the uniqueness of the associate network matrix. Hence, in this case both methods are able to exactly recover the support of the original network (blue circles).

7. CONCLUSIONS

In this paper, we investigated the problem of identifying the underlying network structure based on algebraic properties of fixed-support jointly diagonalizable matrices. Diverging from typical approaches that rely on convex methods, we leverage these algebraic properties of the matrix network representation to devise a submodular method for network support identification. We show that a greedy variant of the submodular analogue of the majorization/minimization technique can be employed to retrieve the unknown support of the network from knowledge of its modes. Numerical examples showcased the proposed algorithm in recovering the support of sparse networks.

8. REFERENCES

[1] E. D. Kolaczyk, *Statistical analysis of network data: methods and models*. Springer Science & Business Media, 2009.

[2] J. A. Deri and J. M. Moura, "New york city taxi analysis with graph signal processing," in *Proc. of the IEEE Conf. on Sig. and Inf. Process. (GLOBALSIP)*, 2016, pp. 1275–1279.

[3] O. Sporns, *Networks of the Brain*. MIT press, 2010.

[4] F. Mittler, et al., "Reactive oxygen gene network of plants," *Trends in Plant Science*, vol. 9, no. 10, pp. 490–498, 2004.

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

[6] D. I. Shuman, et al., "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Sig. Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[7] S. P. Chepuri and G. Leus, "Subgraph detection using graph signals," in *Proc. of the 50th Asilomar Conf. Sig., Syst. and Comp.*, 2016, pp. 532–534.

[8] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Autoregressive moving average graph filtering," IEEE Trans. Sig. Process., vol. 65, no. 2, pp. 274–288, 2017.

[9] M. Coutino, E. Isufi, and G. Leus, "Distributed edge-variant graph filters," in *2017 IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2017.

[10] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Trans. on Sig. and Inf. Process. over Netws.*, vol. 3, no. 3, pp. 467–483, 2017.

[11] R. Shafipour, S. Segarra, A. G. Marques, and G. Mateos, "Network topology inference from non-stationary graph signals," in *Proc. of the IEEE Int. Conf. on Acoust. Speech and Sig. Process. (ICASSP)*, 2017, pp. 5870–5874.

[12] G. Karanikolas, G. B. Giannakis, K. Slavakis, and R. M. Leahy, "Multi-kernel based nonlinear models for connectivity identification of brain networks," in *Proc. of the IEEE Int. Conf. on Acoust. Speech and Sig. Process. (ICASSP)*, 2016, pp. 6315–6319.

[13] Y. Shen, B. Baingana, and G. B. Giannakis, "Kernel-based structural equation models for topology identification of directed networks," *IEEE Trans. on Sig. Process.*, vol. 65, no. 10, pp. 2503–2516, 2017.

[14] K.-S. Lu and A. Ortega, "Closed form solutions of combinatorial graph laplacian estimation under acyclic topology constraints," *arXiv preprint arXiv:1711.00213*, 2017.

[15] Y. Shen, B. Baingana, and G. B. Giannakis, "Tensor decompositions for identifying directed graph topologies and tracking dynamic networks," *IEEE Trans. on Sig. Process.*, vol. 65, no. 14, pp. 3675–3687, 2017.

[16] J. Zhou and J.-a. Lu, "Topology identification of weighted complex dynamical networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 386, no. 1, pp. 481–491, 2007.

[17] S. P. Chepuri, S. Liu, G. Leus, and A. O. Hero, "Learning sparse graphs under smoothness prior," in *Proc. of the IEEE Int. Conf. on Acoust. Speech and Sig. Process. (ICASSP)*, 2017, pp. 6508–6512.

[18] A. Krause and C. Guestrin, "Near-optimal observation selection using submodular functions," in *AAAI*, vol. 7, 2007, pp. 1650–1654.

[19] C. Guestrin, A. Krause, and A. P. Singh, "Near-optimal sensor placements in gaussian processes," in *Proc. of the 22nd Int. Conf. on Mach. Learn.* ACM, 2005, pp. 265–272.

[20] M. Coutino, S. P. Chepuri, and G. Leus, "Near-optimal sparse sensing for gaussian detection with correlated observations," *arXiv preprint arXiv:1710.09676*, 2017.

[21] M. Coutino, S. Chepuri, and G. Leus, "Near-optimal greedy sensor selection for mvdr beamforming with modular budget constraint," in *Proc. of the 25th European Sig. Process. Conf. (EUSIPCO)*, 2017, pp. 1981–1985.

[22] L. F. Chamon and A. Ribeiro, "Greedy sampling of graph signals," *IEEE Trans. on Sig. Process.*, vol. 66, no. 1, pp. 34–47, 2018.

[23] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *IEEE Transactions on Signal Processing*, vol. 65, no. 22, pp. 5911–5926, 2016.

[24] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3462–3477, 2017.

[25] S. Fujishige, *Submodular functions and optimization*. Elsevier, 2005, vol. 58.

[26] R. K. Iyer and J. A. Bilmes, "Submodular optimization with submodular cover and submodular knapsack constraints," in *Advances in Neural Information Processing Systems*, 2013, pp. 2436–2444.

[27] R. Iyer and J. Bilmes, "Algorithms for approximate minimization of the difference between submodular functions, with applications," *arXiv preprint arXiv:1207.0560*, 2012.

[28] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Trans. on Sig. Process.*, vol. 65, no. 3, pp. 794–816, 2017.

[29] M. X. Goemans, N. J. Harvey, S. Iwata, and V. Mirrokni, "Approximating submodular functions everywhere," in *Proc. of the 20th annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2009, pp. 535–544.

[30] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák, "Maximizing a submodular set function subject to a matroid constraint," in *Int. Conf. on Integer Prog. and Combinatorial Opt.* Springer, 2007, pp. 182–196.

[31] R. Diestel, *Graph theory*. Springer-Verlag Berlin and Heidelberg GmbH & amp, 2000.

[32] W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, no. 4, pp. 452–473, 1977.