frontiers in Applied Mathematics and Statistics

ORIGINAL RESEARCH
published: 19 January 2018
doi: 10.3389/fams.2017.00028

When Is Network Lasso Accurate?

Alexander Jung, Nguyen Tran and Alexandru Mara*

Department of Computer Science, Aalto University, Espoo, Finland

The "least absolute shrinkage and selection operator" (Lasso) method has been adapted recently for network-structured datasets. In particular, this network Lasso method allows to learn graph signals from a small number of noisy signal samples by using the total variation of a graph signal for regularization. While efficient and scalable implementations of the network Lasso are available, only little is known about the conditions on the underlying network structure which ensure network Lasso to be accurate. By leveraging concepts of compressed sensing, we address this gap and derive precise conditions on the underlying network topology and sampling set which guarantee the network Lasso for a particular loss function to deliver an accurate estimate of the entire underlying graph signal. We also quantify the error incurred by network Lasso in terms of two constants which reflect the connectivity of the sampled nodes.

Keywords: compressed sensing, big data, semi-supervised learning, complex networks, convex optimization, clustering

OPEN ACCESS

Edited by:
Juergen Prestin,
University of Lübeck, Germany

Reviewed by:
Katerina Hlavackova-Schindler,
University of Vienna, Austria
Valeriya Naumova,
Simula Research Laboratory, Norway

****Correspondence:***
Alexander Jung
alexander.jung@aalto.fi

Specialty section:
This article was submitted to Mathematics of Computation and Data Science, a section of the journal Frontiers in Applied Mathematics and Statistics

Received: *09 October 2017*
Accepted: *28 December 2017*
Published: *19 January 2018*

Citation:
Jung A, Tran N and Mara A (2018) When Is Network Lasso Accurate? Front. Appl. Math. Stat. 3:28. doi: 10.3389/fams.2017.00028

1. INTRODUCTION

In many applications ranging from image processing, social networks to bioinformatics, the observed datasets carry an intrinsic network structure. Such datasets can be represented conveniently by signals defined over a "data graph" which models the network structure inherent to the dataset [1, 2]. The nodes of this data graph represent individual data points which are labeled by some quantity of interest, e.g., the class membership in a classification problem. We represent this label information as a graph signal whose value for a particular node is given by its label [1, 3–8]. This graph signal representation of datasets allows to apply efficient methods from graph signal processing (GSP) which are obtained, in turn, by extending established methods (e.g., fast filtering and transforms) from discrete time signal processing (over chain graphs) to arbitrary graphs [9–11].

The resulting graph signals are typically clustered, i.e., these signals are nearly constant over well connected subset of nodes (clusters) in the data graph. Exploiting this clustering property enables the accurate recovery of graph signals from few noisy samples. In particular, using the total variation to measure how well a graph signal conforms with the underlying cluster structure, the authors of Hallac et al. [12] obtain the network Lasso (nLasso) by adapting the well-known Lasso estimator which is widely used for learning sparse models [13, 14]. The nLasso can be interpreted as an instance of the regularized empirical risk minimization principle, using total variation of a graph signal for the regularization. Some applications where the use of nLasso based methods has proven beneficial include housing price prediction and personalized medicine [12, 15].

A scalable implementation of the nLasso has been obtained via the alternating direction method of multipliers (ADMM) [16]. However, the authors of Boyd et al. [16] do not discuss conditions on the underlying network structure which ensure success of the network Lasso. We close this gap in the understanding of the performance of network Lasso, by deriving sufficient conditions on the data graph (cluster) structure and sampling set such that nLasso is accurate. To this end,

we introduce a simple model for clustered graph signals which are constant over well connected groups or clusters of nodes. We then define the notion of resolving sampling sets, which relates the cluster structure of the data graph to the sampling set. Our main contribution is an upper bound on the estimation error obtained from nLasso when applied to resolving sampling sets. This upper bound depends on two numerical parameters which quantify the connectivity between sampled nodes and cluster boundaries.

Much of the existing work on recovery conditions and methods for graph signal recovery (e.g., [17–22]), relies on spectral properties of the data graph Laplacian matrix. In contrast, our approach is based directly on the connectivity properties of the underlying network structure. The closest to our work is Sharpnack et al. [23] and Wang et al. [24], which provide sufficient conditions such that a special case of the nLasso (referred to as the "edge Lasso") accurately recovers piece-wise constant (or clustered) graph signals from noisy observations. However, these works require access to fully labeled datasets, while we consider datasets which are only partially labeled (as it is typical for machine learning applications where label information is costly).

1.1. Outline

The problem setting considered is formalized in section 2. In particular, we show how to formulate the problem of learning a clustered graph signal from a small amount of signal samples as a convex optimization problem, which is underlying the nLasso method. Our main result, i.e., an upper bound on the estimation error of nLasso is stated in section 3. Numerical experiments which illustrate our theoretical findings are discussed in section 4.

1.2. Notation

We will conform to standard notation of linear algebra as used, e.g., in Golub and Van Loan [25]. For a binary variable b, we denote its negation as $\bar{b}$.

2. PROBLEM FORMULATION

We consider datasets which are represented by a network model, i.e., a data graph $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathbf{W})$ with node set $\mathcal{V}=\{1,\ldots,N\}$, edge set $\mathcal{E}$ and weight matrix $\mathbf{W}\in\mathbb{R}_{+}^{N\times N}$. The nodes $\mathcal{V}$ of the data graph represent individual data points. For example, the node $i\in\mathcal{V}$ might represent a (super-)pixel in image processing, a neuron of a neural network [26] or a social network user profile [27].

Many applications naturally suggest a notion of similarity between individual data points, e.g., the profiles of befriended social network users or grayscale values of neighboring image pixels. These domain-specific notions of similarity are represented by the edges of the data graph $\mathcal{G}$, i.e., the nodes $i,j\in\mathcal{V}$ representing similar data points are connected by an undirected edge $\{i,j\}\in\mathcal{E}$. We denote the neighborhood of the node $i\in\mathcal{V}$ by $\mathcal{N}(i):=\{j\in\mathcal{V}:\{i,j\}\in\mathcal{E}\}$. It will be convenient to associate with each undirected edge $\{i,j\}$ a pair of directed edges, i.e., (i,j) and (j,i). With slight abuse of notation we will treat the elements of the edge set $\mathcal{E}$ either as undirected edges $\{i,j\}$ or as pairs of two directed edges (i,j) and (j,i).

In some applications it is possible to quantify the extent to which data points are similar, e.g., via the physical distance between neighboring sensors in a wireless sensor network application [28]. Given two similar data points $i,j\in\mathcal{V}$, which are connected by an edge $\{i,j\}\in\mathcal{E}$, we will quantify the strength of their connection by the edge weight $W_{i,j}>0$ which we collect in the symmetric weight matrix $\mathbf{W}\in\mathbb{R}_{+}^{N\times N}$. The absence of an edge between nodes $i,j\in\mathcal{V}$ is encoded by a zero weight $W_{i,j}=0$. Thus the edge structure of the data graph $\mathcal{G}$ is fully specified by the support (locations of the non-zero entries) of the weight matrix $\mathbf{W}$.

2.1. Graph Signals

Beside the network structure, encoded in the data graph $\mathcal{G}$, datasets typically also contain additional labeling information. We represent this additional label information by a graph signal defined over $\mathcal{G}$. A graph signal $x[\cdot]$ is a mapping $\mathcal{V}\rightarrow\mathbb{R}$, which associates every node $i\in\mathcal{V}$ with the signal value $x[i]\in\mathbb{R}$ (which might representing a label characterizing the data point). We denote the set of all graph signals defined over a graph $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathbf{W})$ by $\mathbb{R}^{\mathcal{V}}$.

Many machine learning methods for network structured data rely on a "cluster hypothesis" [4]. In particular, we assume the graph signals $x[\cdot]$ representing the label information of a dataset conforms with the cluster structure of the underlying data graph. Thus, any two nodes $i,j\in\mathcal{V}$ out of a well-connected region ("cluster") of the data graph tend to have similar signal values, i.e., $x[i]\approx x[j]$. Two important application domains where this cluster hypothesis has been applied successfully are digital signal processing where time samples at adjacent time instants are strongly correlated for sufficiently high sampling rate (cf. **Figure 1A**) as well as processing of natural images whose close-by pixels tend to be colored likely (cf. **Figure 1B**). The cluster hypothesis is verified also often in social networks where the clusters are cliques of individuals having similar properties (cf. **Figure 1C** and Newman [29, Chap. 3]).

In what follows, we quantify the extend to which a graph signal $x[\cdot]\in\mathbb{R}^{\mathcal{V}}$ conforms with the clustering structure of the data graph $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathbf{W})$ using its *total variation* (TV)

$$\|x[\cdot]\|_{\mathrm{TV}}:=\sum_{\{i,j\}\in\mathcal{E}}W_{i,j}|x[j]-x[i]|. \tag{1}$$

For a subset of edges $\mathcal{S}\subseteq\mathcal{E}$, we use the shorthand

$$\|x[\cdot]\|_{\mathcal{S}}:=\sum_{\{i,j\}\in\mathcal{S}}W_{i,j}|x[j]-x[i]|. \tag{2}$$

For a supervised machine learning application, the signal values $x[i]$ might represent class membership in a classification problem or the target (output) value in a regression problem. For the house price example considered in Hallac et al. [12], the vector-valued graph signal $\mathbf{x}[i]$ corresponds to a regression weight vector for a local pricing model (used for the house market in a limited geographical area represented by the node i).

Consider a partition $\mathcal{F}=\{\mathcal{C}_1,\ldots,\mathcal{C}_{|\mathcal{F}|}\}$ of the data graph $\mathcal{G}$ into $|\mathcal{F}|$ disjoint subsets $\mathcal{C}_l$ of nodes ("clusters") such that

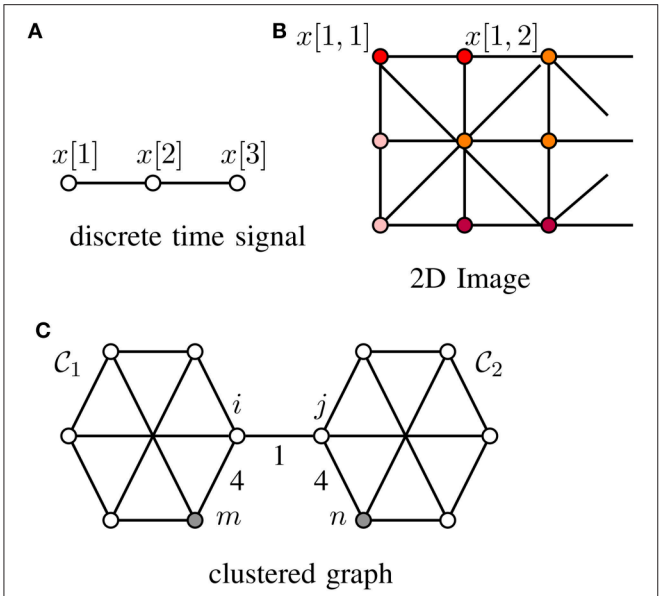

FIGURE 1 | Graph signals defined over **(A)** a chain graph (representing, e.g., discrete time signals), **(B)** grid graph (representing, e.g., 2D-images) and **(C)** a general graph (representing, e.g., social network data), whose edges $\{i,j\} \in \mathcal{E}$ are captioned by edge weights $W_{i,j}$.

$\mathcal{V} = \bigcup_{l=1}^{|\mathcal{F}|} \mathcal{C}_l$. We associate a subset $\mathcal{C} \subseteq \mathcal{V}$ of nodes with a particular "indicator" graph signal

$$\mathcal{I}_{\mathcal{C}}[i] := \begin{cases} 1 & \text{if } i \in \mathcal{C} \\ 0 & \text{else.} \end{cases} \tag{3}$$

A simple model of clustered graph signals is then obtained by piece-wise constant or clustered graph signals of the form

$$x[i] = \sum_{l=1}^{|\mathcal{F}|} a_l \mathcal{I}_{\mathcal{C}_l}[i]. \tag{4}$$

In **Figure 2**, we depict a clustered graph signal for a chain graph with 10 nodes which are partitioned into two clusters: $\mathcal{C}_1$ and $\mathcal{C}_2$.

It will be convenient to define, for a given partition $\mathcal{F}$, its boundary $\partial\mathcal{F} \subseteq \mathcal{E}$ as the set of edges $\{i,j\} \in \mathcal{E}$ which connect nodes $i \in \mathcal{C}_a$ and $j \in \mathcal{C}_b$ from different clusters, i.e., with $\mathcal{C}_a \neq \mathcal{C}_b$. With a slight abuse of notation, we will use the same symbol $\partial\mathcal{F}$ also to denote the set of nodes which are connected to a node from another cluster.

The TV of a clustered graph signal of the form (Equation 4) can be upper bounded as

$$\|x[\cdot]\|_{\mathrm{TV}} \leq 2 \max_{l \in \{1,\ldots,|\mathcal{F}|\}} |a_l| \sum_{\{i,j\} \in \partial\mathcal{F}} W_{i,j}. \tag{5}$$

Thus, for a partition $\mathcal{F}$ with small weighted boundary $\sum_{\{i,j\} \in \partial\mathcal{F}} W_{i,j}$, the associated clustered graph signals (Equation 4) have small TV $\|x[\cdot]\|_{\mathrm{TV}}$ due to Equation (5).

The signal model (Equation 4), which also has been used in Sharpnack et al. [23] and Wang et al. [24], is closely related to the stochastic block model (SBM) [30]. Indeed, the SBM is obtained from Equation (4) by choosing the coefficients $a_{\mathcal{C}}$ uniquely for each cluster, i.e., $a_{\mathcal{C}} \in \{1,\ldots,|\mathcal{F}|\}$. Moreover, the SBM provides a generative (stochastic) model for the edges within and between the clusters $\mathcal{C}_l$.

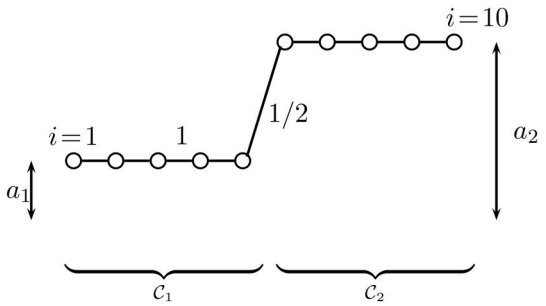

FIGURE 2 | A clustered graph signal $x[i] = a_1 \mathcal{I}_{\mathcal{C}_1}[i] + a_2 \mathcal{I}_{\mathcal{C}_2}[i]$ (cf. Equation 4) defined over a chain graph which is partitioned into two equal-size clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ which consist of consecutive nodes. The edges connecting nodes within the same cluster have weight 1, while the single edge connecting nodes from different clusters has weight 1/2.

We highlight that the clustered signal model (Equation 4) is somewhat dual to the model of band-limited graph signals [1, 4–7, 17, 19]. The model of band-limited graph signals is obtained by the subspaces spanned by the eigenvectors of the graph Laplacian corresponding to the smallest (in magnitude) eigenvalues, i.e., the low-frequency components. Such band-limited graph signals are smooth in the sense of small values of the Laplacian quadratic form [31]

$$\sum_{\{i,j\} \in \mathcal{E}} W_{i,j} (x[j] - x[i])^2 = \mathbf{x}^T \mathbf{L} \mathbf{x}. \tag{6}$$

Here, we used the vector representation $\mathbf{x} = (x[1],\ldots,x[N])^T$ of the graph signal $x[\cdot]$ and the graph Laplacian matrix $\mathbf{L} \in \mathbb{R}^{N \times N}$ defined element-wise as

$$L_{i,j} = \begin{cases} \sum_{k \in \mathcal{V}} W_{i,k} & \text{if } i = j \\ -W_{i,j} & \text{otherwise.} \end{cases} \tag{7}$$

A band-limited graph signal $x[\cdot]$ is characterized by a clustering (within a small bandwidth) of their graph Fourier transform (GFT) coefficients [22]

$$\tilde{x}[l] := \mathbf{u}_l^T \mathbf{x}, \text{ for } l = 1,\ldots,N, \tag{8}$$

with the orthonormal eigenvectors $\{\mathbf{u}_l\}_{l=1}^N$ of the graph Laplacian matrix $\mathbf{L}$. In particular, by the spectral decomposition of the psd graph Laplacian matrix $\mathbf{L}$ (cf. Equation 7), we have $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H$ with $\mathbf{U} = (\mathbf{u}_1,\ldots,\mathbf{u}_N)$ and the diagonal matrix $\mathbf{\Lambda}$ having (in decreasing order) the non-negative eigenvalues λ_l of $\mathbf{L}$ on its diagonal.

In contrast to band-limited graph signals, a clustered graph signal of the form (Equation 4) will typically have GFT coefficients which are spread out over the entire (graph)

frequency range. Moreover, while band-limited graph signals are characterized by having a sparse GFT, a clustered graph signal of the form (Equation 4) has a dense (non-sparse) GFT in general. On the other hand, while a clustered graph signal of the form (Equation 4) has sparse signal differences $\{x[i]-x[j]\}_{\{i,j\}\in\mathcal{E}}$, the signal differences of a band-limited graph signal are dense (non-sparse).

Let us illustrate the duality between the clustered graph signal model (Equation 4) and the model of band-limited graph signals (cf. [7, 17]) by considering a dataset representing a finite length segment of a time series. The data graph $\mathcal{G}_0$ underlying this time series data is chosen as a chain graph (cf. **Figure 2**), consisting of $N = 100$ nodes which represent the individual time samples. The time series is partitioned into two clusters $\mathcal{C}_1, \mathcal{C}_2$, each cluster consisting of 50 consecutive nodes (time samples). We model the correlations between successive time samples using edge weight $W_{i,j} = 1$ for data points i, j belonging to the same cluster and a smaller weight $W_{i,j} = 1/2$ for the single edge $\{i, j\}$ connecting the two clusters $\mathcal{C}_1$ and $\mathcal{C}_2$.

A clustered graph signal (time series) $x_0[i] = a_1\mathcal{I}_{\mathcal{C}_1}[i] + a_2\mathcal{I}_{\mathcal{C}_2}[i]$ (cf. Equation 4) defined over $\mathcal{G}_0$ is characterized by very sparse signal differences $\{x_0[i] - x_0[j]\}_{\{i,j\}\in\mathcal{E}}$. Indeed the signal difference $x_0[i]-\mathbf{x}_0[j]$ of the clustered graph signal $\mathbf{x}_0[\cdot]$ is non-zero only for the single edge $\{i, j\}$ which connects $\mathcal{C}_1$ and $\mathcal{C}_2$. In stark contrast, the GFT of $x_0[\cdot]$ is spread out over the entire (graph) frequency range (cf. **Figure 3**), i.e., the graph signal $x_0[\cdot]$ does not conform with the band-limited signal model.

On the other hand, we illustrate in **Figure 4** a graph signal $x_{\mathrm{BL}}[\cdot]$ with GFT coefficients $\tilde{x}_{\mathrm{BL}}[l] = 1$ (cf. Equation 8) for $l = 1, 2$ and $\tilde{x}_{\mathrm{BL}}[l] = 0$ otherwise. Thus, the graph signal is clearly band-limited (it has only two non-zero GFT coefficients) but the signal differences $x_{\mathrm{BL}}[i] - x_{\mathrm{BL}}[j]$ across the edges $\{i, j\} \in \mathcal{E}$ are clearly non-sparse.

2.2. Recovery via nLasso

Given a dataset with data graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, we aim at recovering a graph signal $x[\cdot] \in \mathbb{R}^{\mathcal{V}}$ from its noisy values

$$y[i] = x[i] + e[i] \tag{9}$$

provided on a (small) sampling set

$$\mathcal{M} \coloneqq \{i_1, \ldots, i_M\} \subseteq \mathcal{V}. \tag{10}$$

Typically $M \ll N$, i.e., the sampling set is a small subset of all nodes in the data graph $\mathcal{G}$.

The recovered graph signal $\hat{x}[\cdot]$ should incur only a small empirical (or training) error

$$\widehat{E}(\hat{x}[\cdot]) \coloneqq \sum_{i\in\mathcal{M}} |\hat{x}[i]-y[i]|. \tag{11}$$

Note that the definition (Equation 11) of the empirical error involves the ℓ_1-norm of the deviation $\hat{x}[\cdot]i - y[i]$ between recovered and measured signal samples. This is different from the error criterion used in the ordinary Lasso, i.e., the squared-error loss $\sum_{i\in\mathcal{M}}(\hat{x}[i]-y[i])^2$ [32]. The definition (Equation 11) is beneficial for applications with measurement errors e_i (cf. Equation 9) having mainly small values except for a few large outliers [18, 33]. However, by contrast to plain Lasso, the error function in Equation (11) does not satisfy a restricted strong convexity property [34], which might be detrimental for the convergence speed of the resulting recovery methods (cf. Section 4).

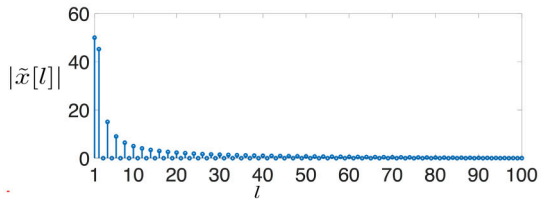

FIGURE 3 | The magnitudes of the GFT coefficients $\tilde{x}[l]$ (cf. Equation 8) of a clustered graph signal $x_0[\cdot]$ defined over a chain graph (cf. **Figure 2**).

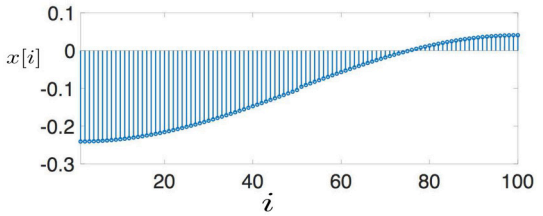

FIGURE 4 | A band-limited graph signal defined over a chain graph with $N = 100$.

In order to recover a clustered graph signal with a small TV $\|\hat{x}[\cdot]\|_{\mathrm{TV}}$ (cf. Equation 5) from the noisy signal samples $\{y[i]\}_{i\in\mathcal{M}}$ it is sensible to consider the recovery problem

$$\hat{x}[\cdot] \in \operatorname*{arg\,min}_{\tilde{x}[\cdot]\in\mathbb{R}^{\mathcal{V}}} \widehat{E}(\tilde{x}[\cdot]) + \lambda\|\tilde{x}[\cdot]\|_{\mathrm{TV}}. \tag{12}$$

This recovery problem amounts to a convex optimization problem [35], which, as the notation already indicates, might have multiple solutions $\hat{x}[\cdot]$ (which form a convex set). In what follows, we will derive conditions on the sampling set $\mathcal{M}$ such that any solution $\hat{x}[\cdot]$ of Equation (12) allows to accurately recover clustered a graph signal $x[\cdot]$ of the form (Equation 4).

Any graph signal obtained from Equation (12) balances the empirical error $\widehat{E}(\hat{x}[\cdot])$ with the TV $\|\hat{x}[\cdot]\|_{\mathrm{TV}}$ in an optimal manner. The parameter λ in Equation (12) allows to trade off a small empirical error against the amount to which the resulting signal is clustered, i.e., having a small TV. In particular, choosing a small value for λ enforces the solutions of Equation (12) to yield a small empirical error, whereas choosing a large value for λ enforces the solutions of Equation (12) to have small TV. Our analysis in section 3 provides a selection criterion for the parameter λ which is based on the location of the sampling set $\mathcal{M}$ (cf. Equation 10) and the partition $\mathcal{F}$ underlying the clustered graph signal model (Equation 4). Alternatively, for sufficiently large sampling sets one might choose λ using a cross-validation procedure [13].

Note that the recovery problem (Equation 12) is a particular instance of the generic nLasso problem studied in Hallac et al. [12]. There exist efficient convex optimization methods for solving the nLasso problem (Equation 12) (cf. [36] and the references therein). In particular, the alternating method of multipliers (ADMM) has been applied to the nLasso problem in Hallac et al. [12] to obtain a scalable learning algorithm which can cope with massive heterogeneous datasets.

3. WHEN IS NETWORK LASSO ACCURATE?

The accuracy of graph signal recovery methods based on the nLasso problem (Equation 12), depends on how close the solutions $\hat{x}[\cdot]$ of Equation (12) are to the true underlying graph signal $x[\cdot] \in \mathbb{R}^{\mathcal{V}}$. In what follows, we present a condition which guarantees any solution $\hat{x}[\cdot]$ of Equation (12) to be close to the underlying graph signal $x[\cdot]$ if it is clustered of the form (Equation 4).

A main contribution of this paper is the insight that the accuracy of nLasso methods, aiming at solving Equation (12), depends on the topology of the underlying data graph via the existence of certain *flows with demands* [37]. Given a data graph $\mathcal{G}$, we define a flow on it as a mapping $h[\cdot]: \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ which assigns each directed edge (i,j) the value $h[(i,j)]$, which can be interpreted as the amount of some quantity flowing through the edge (i,j) [37]. A flow with demands has to satisfy the conservation law

$$\sum_{j \in \mathcal{N}(i)} h(j,i) - h(i,j) = d[i], \text{ for any } i \in \mathcal{V} \tag{13}$$

with a prescribed demand $d[i]$ for each node $i \in \mathcal{V}$. Moreover, we require flows to satisfy the capacity constraints

$$|h(i,j)| \leq W_{i,j} \text{ for any edge } (i,j) \in \mathcal{E} \setminus \partial\mathcal{F}. \tag{14}$$

Note that the capacity constraint (Equation 14) applies only to intra-cluster edges and does not involve the boundary edges $\partial\mathcal{F}$. The flow values $h(i,j)$ at the boundary edges $(i,j) \in \partial\mathcal{F}$ take a special role in the following definition of the notion of resolving sampling sets.

Definition 1. *Consider a dataset with data graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ which contains the sampling set $\mathcal{M} \subseteq \mathcal{V}$. The sampling set $\mathcal{M}$ resolves a partition $\mathcal{F} = \{\mathcal{C}_1, \ldots, \mathcal{C}_{|\mathcal{F}|}\}$ with constants K and L if, for any $b_{i,j} \in \{0,1\}$ with $\{i,j\} \in \partial\mathcal{F}$, there exists a flow $h[\cdot]$ on $\mathcal{G}$ (cf. Equations 13, 14)* with

$$h(i,j) = b_{i,j} \cdot L \cdot W_{i,j}, h(j,i) = \bar{b}_{i,j} \cdot L \cdot W_{i,j} \tag{15}$$

for every boundary edge $\{i,j\} \in \partial\mathcal{F}$ and demands (cf. Equation 13) satisfying

$$|d[i]| \leq K \text{ for } i \in \mathcal{M}, \text{ and } d[i] = 0 \text{ for } i \in \mathcal{V} \setminus \mathcal{M}. \tag{16}$$

This definition requires nodes of a resolving sampling set to be sufficiently well connected with every boundary edge $\{i,j\} \in \partial\mathcal{F}$. In particular, we could think of injecting (absorbing) certain amounts of flow into (from) the data graph at the sampled nodes. At each sampled node $i \in \mathcal{M}$, we can inject (absorb) a flow of level at most K (cf. Equation 16). The injected (absorbed) flow has to be routed from the sampled nodes via the intra-cluster edges to each boundary edge such that it carries a flow value $L \cdot W_{i,j}$. Clearly, this is only possible if there are paths of sufficient capacity between sampled nodes and boundary edges available.

The definition of resolving sampling sets is quantitive as it involves the numerical constants K and L. Our main result stated below is an upper bound on the estimation error of nLasso methods which depends on the value of these constants. It will turn out that resolving sampling sets with a small values of K and large values of L are beneficial for the ability of nLasso to recover the entire graph signal from noisy samples observed on the sampling set. However, the constants K and L are coupled via the flow $h[\cdot]$ used in Definition 1, e.g., the constant K always has to satisfy

$$K \geq \max_{\{i,j\} \in \partial\mathcal{F}} L W_{i,j}. \tag{17}$$

Thus, the minimum possible value for K depends on the values of the edge weights $W_{i,j}$ of the data graph. Moreover, the minimum possible value for L depends on the precise connectivity of sampled nodes with the boundary edges $\partial\mathcal{F}$. Indeed, Definition 1 requires to route (by satisfying the capacity constraints, Equation 14), an amount of flow given by $LW_{i,j}$ from a boundary edge $\{i,j\} \in \partial\mathcal{F}$ to the sampled nodes in $\mathcal{M}$.

In order to make (the somewhat abstract) Definition 1 more transparent, let us state an easy-to-check sufficient condition for a sampling set $\mathcal{M}$ such that it resolves a given partition $\mathcal{F}$.

Lemma 2. *Consider a partition $\mathcal{F} = \{\mathcal{C}_1, \ldots, \mathcal{C}_{|\mathcal{F}|}\}$ of the data graph $\mathcal{G}$ which contains the sampling set $\mathcal{M} \subseteq \mathcal{V}$. If each boundary edge $\{i,j\} \in \partial\mathcal{F}$ with $i \in \mathcal{C}_a$, $j \in \mathcal{C}_b$ is connected to sampled nodes, i.e., $\{m,i\} \in \mathcal{E}$ and $\{n,j\} \in \mathcal{E}$ with $m \in \mathcal{M} \cap \mathcal{C}_a$, $n \in \mathcal{M} \cap \mathcal{C}_b$, and weights $W_{m,i}, W_{n,j} \geq LW_{i,j}$, then the sampling set $\mathcal{M}$ resolves the partition $\mathcal{F}$ with constants L and*

$$K = L \cdot \max_{i \in \mathcal{V}} |\mathcal{N}(i) \cap \partial\mathcal{F}|. \tag{18}$$

In **Figure 1C** we depict a data graph consisting of two clusters $\mathcal{F} = \{\mathcal{C}_1, \mathcal{C}_2\}$. The data graph contains the sampling set $\mathcal{M} = \{m,n\}$ which resolves the partition $\mathcal{F}$ with constants $K = L = 4$ according to Lemma 2.

The sufficient condition provided by Lemma 2 can be used to guide the choice for the sampling set $\mathcal{M}$. In particular Lemma 2 suggests to sample more densely near the boundary edges $\partial\mathcal{F}$ which connect different clusters. This rationale allows to cope with applications where the underlying partition $\mathcal{F}$ is unknown. In particular, we could use highly scalable local clustering methods (cf. [38]) to find the cluster boundaries $\partial\mathcal{F}$ and then select the sampled nodes in their vicinity. Another approach to cope with lack of information about $\mathcal{F}$ is based on using random walks to identify the subset of nodes with a large boundary which are sampled more densely [39].

We now state our main result which is that solutions of the nLasso problem (Equation 12) allow to accurately recover the true underlying clustered graph signal $x[\cdot]$ (conforming with the partition $\mathcal{F}$ (cf. Equation 4) from the noisy measurements (Equation 9) whenever the sampling set $\mathcal{M}$ resolves the partition $\mathcal{F}$.

Theorem 3. *Consider a clustered graph signal $x[\cdot]$ of the form (Equation 4), with underlying partition $\mathcal{F} = \{\mathcal{C}_1, \ldots, \mathcal{C}_{|\mathcal{F}|}\}$ of the data graph into disjoint clusters $\mathcal{C}_l$. We observe the noisy signal values $y[i]$ at the samples nodes $\mathcal{M} \subseteq \mathcal{V}$ (cf. Equation 9). If the sampling set $\mathcal{M}$ resolves the partition $\mathcal{F}$ with parameters $K > 0, L > 1$, any solution $\hat{x}[\cdot]$ of the nLasso problem (Equation 12) with $\lambda := 1/K$ satisfies*

$$\|\hat{x}[\cdot] - x[\cdot]\|_{\mathrm{TV}} \leq (K + 4/(L-1)) \sum_{i \in \mathcal{M}} |e[i]|. \tag{19}$$

Thus, if the sampling set $\mathcal{M}$ is chosen such that it resolves the partition $\mathcal{F} = \{\mathcal{C}_1, \ldots, \mathcal{C}_{|\mathcal{F}|}\}$ (cf. Definition 1), nLasso methods (cf. Equation 12) recover a clustered graph signal $\mathbf{x}[\cdot]$ (cf. Equation 4) with an accuracy which is determined by the level of the measurement noise $e[i]$ (cf. Equation 9).

Let us highlight that the knowledge of the partition $\mathcal{F}$ underlying the clustered graph signal model (Equation 4) is only needed for the analysis of nLasso methods leading to Theorem 3. In contrast, the actual implementation methods of nLasso methods based on Equation (12) does not require any knowledge of the underlying partition. What is more, if the true underlying graph signal $\mathbf{x}[\cdot]$ is clustered according to Equation (4) with different signal values a_l for different clusters $\mathcal{C}_l$, the solutions of the nLasso Equation (12) could be used for determining the clusters $\mathcal{C}_l$ which constitute the partition $\mathcal{F}$.

We also note that the bound (Equation 19) characterizes the recovery error in terms of the semi-norm $\|\hat{x}[\cdot] - x[\cdot]\|_{\mathrm{TV}}$ which is agnostic toward a constant offset in the recovered graph signal $\hat{x}[\cdot]$. In particular, having a small value of $\|\hat{x}[\cdot] - x[\cdot]\|_{\mathrm{TV}}$ does in general not imply a small squared error $\sum_{i \in \mathcal{V}} (\hat{x}[i] - x[i])^2$ as there might be an arbitrarily large constant offset contained in the nLasso solution $\hat{x}[\cdot]$.

However, if the error $\|\hat{x}[\cdot] - x[\cdot]\|_{\mathrm{TV}}$ is sufficiently small, we might be able to identify the boundary edges $\{i,j\} \in \partial\mathcal{F}$ of the partition $\mathcal{F}$ underlying a clustered graph signal of the form (Equation 4).

Indeed, for a clustered graph signal of the form (Equation 4), the signal difference $x[i] - x[j]$ across edges is non-zero only for boundary edges $\{i,j\} \in \partial\mathcal{F}$. Lets assume the signal differences of $x[\cdot]$ across boundary edges $\{i,j\} \in \mathcal{F}$ are lower bounded by some positive constant $\eta > 0$ and the nLasso error satisfies $\|\hat{x}[\cdot] - x[\cdot]\|_{\mathrm{TV}} < \eta/2$. As can be verified easily, we can then perfectly recover the boundary $\partial\mathcal{F}$ of the partition $\mathcal{F} = \{\mathcal{C}_1, \ldots, \mathcal{C}_{|\mathcal{F}|}\}$ as precisely those edges $\{i,j\} \in \mathcal{E}$ for which $|\hat{x}[i] - \hat{x}[j]| \geq \eta/2$. Given the boundary $\partial\mathcal{F}$, we can recover the partition $\mathcal{F}$ and, in turn, average the noisy observations $y[i]$ over all sampled nodes $i \in \mathcal{M}$ belonging to the same cluster. This simple post-processing of the nLasso estimate $\hat{x}[i]$ is summarized in Algorithm 1.

Algorithm 1 Post-Processing for nLasso

Input: data graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, noisy signal samples $y[i]$ (cf. Equation 9), nLasso estimate $\hat{x}[\cdot]$ (cf. Equation 12) and threshold $\eta > 0$

construct candidate boundary $\mathcal{S} = \{\{i,j\} \in \mathcal{E} : |\hat{x}[i] - \hat{x}[j]| \geq \eta/2\}$

find partition $\widehat{\mathcal{F}} = \{\mathcal{C}_1, \ldots, \mathcal{C}_{|\widehat{\mathcal{F}}|}\}$ with $\partial\widehat{\mathcal{F}} = \mathcal{S}$

if no such partition exists return "ERR"

for each cluster $\mathcal{C}_l \in \widehat{\mathcal{F}}$

  construct set $\mathcal{A} = \mathcal{C}_l \cap \mathcal{M}$

  if set $\mathcal{A}$ is empty return "ERR"

  for every $i \in \mathcal{C}_l$ set $\tilde{x}[i] = (1/|\mathcal{A}|) \sum_{j \in \mathcal{A}} y[j]$

Output: new estimate $\tilde{x}[\cdot]$ or "ERR"

Lemma 4. *Consider the setting of Theorem 3 involving a clustered graph signal $x[\cdot]$ of the form (Equation 4) with coefficients a_l satisfying $|a_l - a_{l'}| > \eta$ for $l \neq l'$ with a known positive threshold $\eta > 0$. We observe noisy signal samples $y[i]$ (cf. Equation 9) over the sampling set $\mathcal{M}$ with a bounded error $e[i] \leq \epsilon$. If the sampling set $\mathcal{M}$ resolves the partition $\mathcal{F}$ with parameters $K > 0, L > 1$ such that*

$$(K + 4/(L-1)) \sum_{i \in \mathcal{M}} |e[i]| < \eta/2, \tag{20}$$

then the signal $\tilde{x}[\cdot]$ delivered by Algorithm 1 satisfies

$$\sum_{i \in \mathcal{V}} (\tilde{x}[i] - x[i])^2 \leq N\varepsilon^2. \tag{21}$$

4. NUMERICAL EXPERIMENTS

In order to illustrate the theoretical findings of section 3 we report the results of some illustrative numerical experiments involving the recovery of clustered graph signals of the form (Equation 4) from a small number of noisy measurements (Equation 9). To this end, we implemented the iterative method ADMM [16] to solve the nLasso (Equation 12) problem. We applied the resulting semi-supervised learning algorithm to two synthetically generated data sets. The first data set represents a time series, which can be represented as a graph signal over a chain graph. The nodes of the chain graph, which represent the discrete time instants are partitioned evenly into clusters of consecutive nodes. A second experiment is based on data sets generated using a recently proposed generative model for complex networks.

4.1. Chain Graph

Our first experiment, is based on a graph signal defined over a chain graph $\mathcal{G}_{\mathrm{chain}}$ (cf. **Figure 2**) with $N = 10^5$ nodes $\mathcal{V} = \{1, 2, \ldots, N\}$, connected by $N - 1$ undirected edges. The nodes of the data graph $\mathcal{G}_{\mathrm{chain}}$ are partitioned into $N/10$ equal-sized clusters $\mathcal{C}_l$, $l = 1, \ldots, N/10$, each constituted by 10 consecutive nodes. The intrinsic clustering structure of the chain graph $\mathcal{G}_{\mathrm{chain}}$

matches the partition $\mathcal{F}_{\rm chain} = \{\mathcal{C}_l\}_{l=1}^{N/10}$ via the edge weights $W_{i,j}$. In particular, the weights of the edges connecting nodes within the same cluster are chosen i.i.d. according to $W_{i,j} \sim |\mathcal{N}(2, 1/4)|$ (i.e., the absolute value of a Gaussian random variable with mean 2 and variance 1/4). The weights of the edges connecting nodes from different clusters are chosen i.i.d. according to $W_{i,j} \sim |\mathcal{N}(1, 1/4)|$.

We then generate a clustered graph signal $x[\cdot]$ of the form (Equation 4) with coefficients $a_l \in \{1, 5\}$, where the coefficients a_l and $a_{l'}$ of consecutive clusters $\mathcal{C}_l$ and $\mathcal{C}_{l'}$ are different. The graph signal $x[\cdot]$ is observed via noisy samples $y[i]$ (cf. Equation 9 with $e[i] \sim \mathcal{N}(0, 1/4)$) obtained for the nodes $i \in \mathcal{V}$ belonging to a sampling set $\mathcal{M}$. We consider two different choices for the sampling set, i.e., $\mathcal{M} = \mathcal{M}_1$ and $\mathcal{M} = \mathcal{M}_2$. Both choices contain the same number of nodes, i.e., $|\mathcal{M}_1| = |\mathcal{M}_2| = 2 \cdot 10^4$. The sampling set $\mathcal{M}_1$ contains neighbors of cluster boundaries $\partial\mathcal{F}_{\rm chain}$ and conforms to Lemma 2 with constants $K = 5.39$ and $L = 2$ (which have been determined numerically). In contrast, the sampling set $\mathcal{M}_2$ is obtained by selecting nodes uniformly at random from $\mathcal{V}$ and thereby completely ignoring the cluster structure $\mathcal{F}_{\rm chain}$ of $\mathcal{G}_{\rm chain}$.

The noisy measurements $y[i]$ are then input to an ADMM implementation for solving the nLasso problem (Equation 12) with $\lambda = 1/K$. We run ADMM for a fixed number of 300 iterations and using ADMM-parameter $\rho = 0.01$ [16]. In **Figure 5** we illustrate the recovered graph signals (over the first 100 nodes of the chain graph) $\hat{x}[\cdot]$, obtained from noisy signal samples over either sampling set $\mathcal{M}_1$ or $\mathcal{M}_2$.

As evident from **Figure 5**, the recovered signal obtained when using the sampling set $\mathcal{M}_1$, which takes the partition $\mathcal{F}_{\rm chain}$ into account, better resembles the original graph signal $x[\cdot]$ than when using the randomly selected sampling set $\mathcal{M}_2$. The favorable performance of $\mathcal{M}_1$ is also reflected in the empirical normalized mean squared errors (NMSE) between the real and recovered graph signals, which are $\text{NMSE}_{\mathcal{M}_1} = 3.3 \cdot 10^{-2}$ and $\text{NMSE}_{\mathcal{M}_2} = 2.192 \cdot 10^{-1}$, respectively.

We have repeated the above experiment with the same parameters but considering noiseless initial samples $y[i]$ for both sampling sets $\mathcal{M}_1$ and $\mathcal{M}_2$. The recovered graph signals $\hat{x}[\cdot]$ for the first 100 nodes of the chain are presented in **Figure 6**. It can be observed that the recovery starting from the sampling set $\mathcal{M}_1$ (conforming to the partition $\mathcal{F}_{\rm chain}$) perfectly resembles the original graph signal $x[\cdot]$, as expected according to our upper bound in Equation (19). The NMSE obtained after running ADMM for 300 iterations for solving the nLasso problem (Equation 12) are $\text{NMSE}_{\mathcal{M}_1} = 7.5 \cdot 10^{-6}$ and $\text{NMSE}_{\mathcal{M}_2} = 1.475 \cdot 10^{-1}$, respectively.

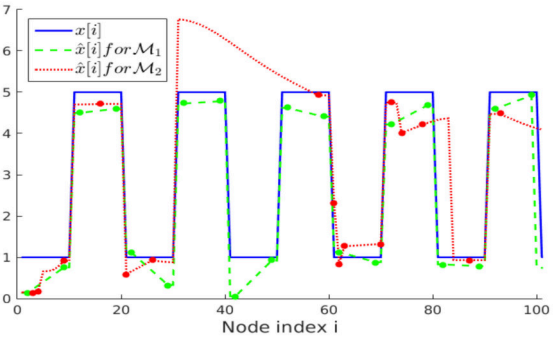

FIGURE 5 | Clustered graph signal $x[\cdot]$ along with the recovered graph signals obtained from noisy signal samples set $\mathcal{M}_1$ (Lemma 2) and $\mathcal{M}_2$ (random).

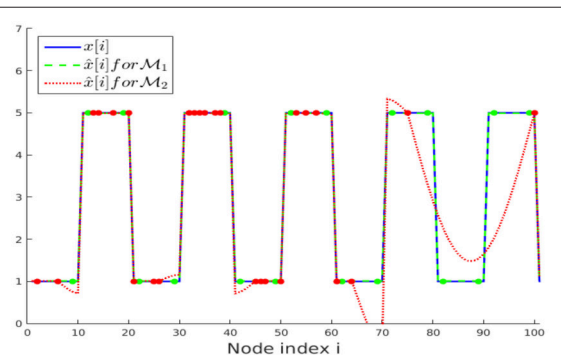

FIGURE 6 | Clustered graph signal $x[\cdot]$ along with the recovered graph signals obtained from noiseless signal samples over sampling set $\mathcal{M}_1$ (Lemma 2) and $\mathcal{M}_2$ (random). The noiseless signal samples $y[i] = x[i]$ are marked with dots.

4.2. Complex Network

In this second experiment, we generate a data graph $\mathcal{G}_{\rm lfr}$ using the generative model introduced by Lancichinetti et al. [40], in what follows referred to as LFR model. The LFR model aims at imitating some key characteristics of real-world networks such as power law distributions of node degrees and community sizes. The data graph $\mathcal{G}_{\rm lfr}$ contains a total of $N = 10^5$ nodes which are partitioned into 1,399 clusters, $\mathcal{F}_{\rm lfr} = \{\mathcal{C}_1, \ldots, \mathcal{C}_{1399}\}$. The nodes $\mathcal{V}$ of $\mathcal{G}_{\rm lfr}$ are connected by a total of $9.45 \cdot 10^5$ undirected edges $\mathcal{E}$.

The edge weights $W_{i,j}$, which are also provided by the LFR model, conform to the cluster structure of $\mathcal{G}_{\rm lfr}$, i.e., inter-cluster edges $\{i, j\} \in \mathcal{E}$ with $i, j \in \mathcal{C}_l$ have larger weights compared to intra-cluster edges $\{i, j\} \in \mathcal{E}$ with $i \in \mathcal{C}_l$ and $j \in \mathcal{C}_{l'}$. Given the data graph $\mathcal{G}_{\rm lfr}$ and partition $\mathcal{F}_{\rm lfr}$ we generate a clustered graph signal according to Equation (4) as $x[i] = \sum_{j=1}^{1399} a_j \mathcal{I}_{\mathcal{C}_j}[i]$ with coefficients a_j randomly chosen i.i.d. according to a uniform distribution $\mathcal{U}(1, 50)$.

We then try to recover the entire graph signal $x[\cdot]$ by solving the nLasso problem (Equation 12) using noisy measurements $y[i]$, according to Equation (9) with i.i.d. measurement noise $e[i] \sim \mathcal{N}(0, 1/4)$, obtained at the nodes in a sampling set $\mathcal{M}$. As in section 4.1, we consider two different choices $\mathcal{M}_1$ and $\mathcal{M}_2$ for the sampling set which both contain the same number of nodes, i.e., $|\mathcal{M}_1| = |\mathcal{M}_2| = 10^4$. The nodes in sampling set $\mathcal{M}_1$ are selected according to Lemma 2, i.e., by choosing nodes which are well connected (close) to boundary edges $\partial\mathcal{F}_{\rm lfr}$ which connect different clusters of the partition $\mathcal{F}_{\rm lfr}$. In contrast, the sampling set $\mathcal{M}_2$ is constructed by selecting nodes uniformly at random, i.e., the partition $\mathcal{F}_{\rm lfr}$ is not taken into account.

In order to construct the sampling set $\mathcal{M}_1$, we first sorted the edges $\{i, j\} \in \mathcal{E}$ of the data graph $\mathcal{G}_{\rm lfr}$ in ascending order according to their edge weight $W_{i,j}$. We then iterate over the the

edges according to the list, starting with the edge having smallest weight, and for each edge $\{i,j\} \in \mathcal{E}$ we select the neighboring nodes of i and j with highest degree and add them to $\mathcal{M}_1$, if they are not already included there. This process continues until the sampling set $\mathcal{M}_1$ has reached the prescribed size of 10^4. Using Lemma 2, we then verified numerically that the sampling set $\mathcal{M}_1$ resolves $\mathcal{F}_{\rm lfr}$ with constants $K = 142.6$ and $L = 2$ (cf. Definition 1).

The measurements $y[i]$ collected for each sampling sets $\mathcal{M}_1$ and $\mathcal{M}_2$ are fed into the ADMM algorithm (using parameters $\rho = 1/100$) for solving the nLasso problem (Equation 12) with $\lambda = 1/K$. The evolution of the NMSE achieved by the ADMM output for an increasing number the iterations is shown in **Figure 7**. According to **Figure 7** the signal recovered from the sampling set $\mathcal{M}_1$ approximates the true graph signal $x[\cdot]$ more closely compared to when using the sampling set $\mathcal{M}_2$. The NMSE achieved after 300 iterations of ADMM is $\text{NMSE}_{\mathcal{M}_1} = 1.56 \cdot 10^{-2}$ and $\text{NMSE}_{\mathcal{M}_2} = 4.25 \cdot 10^{-2}$, respectively.

Finally, we compare the recovery accuracy of nLasso to that of plain label propagation (LP) [41], which relies on a band-limited signal model (cf. section 2.1). In particular, LP quantifies signal smoothness by the Laplacian quadratic form (Equation 6) instead of the total variation (Equation 1), which underlies nLasso (Equation 12). The signals recovered after running the LP algorithm for 300 iterations for the two sampling sets $\mathcal{M}_1$ and $\mathcal{M}_2$ incur an NMSE of $\text{NMSE}_{\mathcal{M}_1} = 3.1 \cdot 10^{-2}$ and $\text{NMSE}_{\mathcal{M}_2} = 7.43 \cdot 10^{-2}$, respectively. Thus, the signals recovered using nLasso are more accurate compared to LP, as illustrated in **Figure 8**. However, our results indicate that LP also benefits by using the sampling set $\mathcal{M}_1$ whose construction is guided by our theoretical findings (cf. Lemma 2).

5. PROOFS

The high-level idea behind the proof of Theorem 3 is to adapt the concept of compatibility conditions for Lasso type estimators [32]. This concept has been championed for analyzing Lasso type methods [32]. Our main technical contribution is to verify the compatibility condition for a sampling set $\mathcal{M}$ which resolves the partition $\mathcal{F}$ underlying the signal model (Equation 4) (cf. Lemma 6 below).

5.1. The Network Compatibility Condition

As an intermediate step toward proving Theorem 3, we adopt the compatibility condition [42], which has been introduced to analyze Lasso methods for learning sparse signals, to the clustered graph signal model (Equation 4). In particular, we define the network compatibility condition for sampling graph signals with small total variation (cf. Equation 1).

Definition 5. *Consider a data graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ whose nodes $\mathcal{V}$ are partitioned into disjoint clusters $\mathcal{F} = \{\mathcal{C}_1, \ldots, \mathcal{C}_{|\mathcal{F}|}\}$. A sampling set $\mathcal{M} \subseteq \mathcal{V}$ is said to satisfy the network compatibility condition, with constants $K, L > 0$, if*

$$K \sum_{i \in \mathcal{M}} |z[i]| + \|z[\cdot]\|_{\mathcal{E}\setminus\partial\mathcal{F}} \geq L \|z[\cdot]\|_{\partial\mathcal{F}} \tag{22}$$

for any graph signal $z[\cdot] \in \mathbb{R}^{\mathcal{V}}$.

It turns out that any sampling set $\mathcal{M}$ which resolves the partition $\mathcal{F} = \{\mathcal{C}_1, \ldots, \mathcal{C}_{|\mathcal{F}|}\}$ with constants K and L (cf. Definition 1) also satisfies the network compatibility condition (Equation 22) with the same constants.

Lemma 6. *Any sampling set $\mathcal{M}$ which resolves the partition $\mathcal{F}$ with parameters $K, L > 0$ satisfies the network compatibility condition with parameters K, L.*

Proof: Let us consider an arbitrary but fixed graph signal $z[\cdot] \in \mathbb{R}^{\mathcal{V}}$. Since the sampling set $\mathcal{M}$ resolves the partition $\mathcal{F}$ there exists a flow $h[e]$ on $\mathcal{G}$ with (cf. Definition 1)

$$\begin{aligned}
&\sum_{j \in \mathcal{N}(i)} h(j,i) - \sum_{j \in \mathcal{N}(i)} h(i,j) = 0 \text{ for all } i \notin \mathcal{M} \\
&\Big| \sum_{j \in \mathcal{N}(i)} h(j,i) - \sum_{j \in \mathcal{N}(i)} h(i,j) \Big| \leq K \text{ for all } i \in \mathcal{M} \\
&|h(i,j)| \leq W_{i,j} \text{ for } (i,j) \notin \partial\mathcal{F} \\
&h(i,j) \cdot h(j,i) = 0 \text{ for } \{i,j\} \in \partial\mathcal{F}
\end{aligned} \tag{23}$$

Moreover, due to Equation (15), we have the important identity

$$(h(i,j) - h(j,i))(z[i] - z[j]) = L W_{i,j} |z[i] - z[j]| \tag{24}$$

which holds for all boundary edges $\{i,j\} \in \partial\mathcal{F}$. This yields, in turn,

$$\begin{aligned}
L\|z[\cdot]\|_{\partial\mathcal{F}} &\overset{(2)}{=} \sum_{\{i,j\} \in \partial\mathcal{F}} |z[i] - z[j]| L W_{i,j} \\
&\overset{(24)}{=} \sum_{(i,j) \in \partial\mathcal{F}} (z[i] - z[j]) h(i,j).
\end{aligned} \tag{25}$$

Since $\mathcal{E} = \partial\mathcal{F} \cup \big(\mathcal{E} \setminus \partial\mathcal{F}\big)$, we can develop (Equation 25) as

$$\begin{aligned}
&L\|z[\cdot]\|_{\partial\mathcal{F}} \\
&= \sum_{(i,j) \in \mathcal{E}} (z[i] - z[j]) h(i,j) - \sum_{(i,j) \in \mathcal{E}\setminus\partial\mathcal{F}} (z[i] - z[j]) h(i,j) \\
&= \sum_{i \in \mathcal{V}} z[i] \sum_{j \in \mathcal{N}(i)} (h(j,i) - h(i,j)) \\
&\qquad - \sum_{(i,j) \in \mathcal{E}\setminus\partial\mathcal{F}} (z[i] - z[j]) h(i,j) \\
&\overset{(23)}{\leq} K \sum_{i \in \mathcal{M}} |z[i]| + \sum_{\{i,j\} \in \mathcal{E}\setminus\partial\mathcal{F}} |z[i] - z[j]| W_{i,j} \\
&= K \sum_{i \in \mathcal{M}} |z[i]| + \|z[\cdot]\|_{\mathcal{E}\setminus\partial\mathcal{F}}
\end{aligned} \tag{26}$$

which verifies (Equation 22). □

The next result shows that if the sampling set satisfies the network compatibility condition, any solution of the nLasso (Equation 12) allows to accurately recover a clustered graph signal (cf. Equation 4).

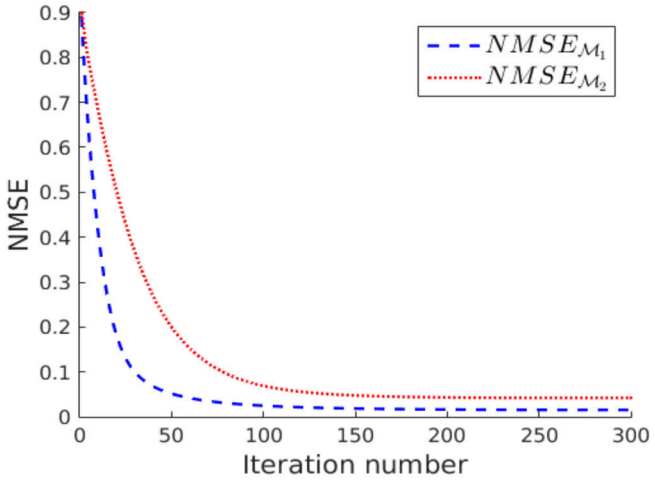

FIGURE 7 | Evolution of the NMSE achieved by increasing number of nLasso-ADMM iterations when using sampling set $\mathcal{M}_1$ or $\mathcal{M}_2$, respectively.

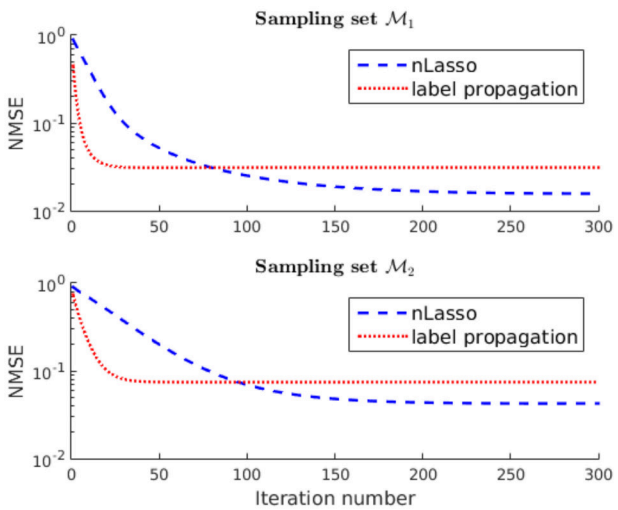

FIGURE 8 | Evolution of the NMSE achieved by increasing number of nLasso-ADMM iterations and LP iterations. Both algorithms are fed with the same signal samples obtained either over sampling set $\mathcal{M}_1$ or $\mathcal{M}_2$, respectively.

Lemma 7. *Consider a clustered graph signal $x[\cdot]$ of the form (Equation 4) defined on the data graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ whose nodes $\mathcal{V}$ are partitioned into the clusters $\mathcal{F} = \{\mathcal{C}_1, \ldots, \mathcal{C}_{|\mathcal{F}|}\}$. We observe the noisy signal values $y[i]$ at the sampled nodes $\mathcal{M} \subseteq \mathcal{V}$ (cf. Equation 9). If the sampling set $\mathcal{M}$ satisfies the network compatibility condition with constants $L > 1, K > 0$, then any solution of the nLasso problem (Equation 12), for the choice $\lambda := 1/K$, satisfies*

$$\|\hat{x}[\cdot] - x[\cdot]\|_{\mathrm{TV}} \leq (K+4/(L-1)) \sum_{i \in \mathcal{M}} |e[i]|. \tag{27}$$

Proof: Consider a solution $\hat{x}[\cdot]$ of the nLasso problem (Equation 12) which is different from the true underlying clustered signal $x[\cdot]$ (cf. Equation 4). We must have (cf. Equation 9)

$$\sum_{i \in \mathcal{M}} |\hat{x}[i] - y[i]| + \lambda \|\hat{x}[\cdot]\|_{\mathrm{TV}} \leq \sum_{i \in \mathcal{M}} |e[i]| + \lambda \|x[\cdot]\|_{\mathrm{TV}} \tag{28}$$

since otherwise the true underlying signal $x[\cdot]$ would achieve a smaller objective value in Equation (12) which, in turn, would contradict the premise that $\hat{x}[\cdot]$ is optimal for the problem (Equation 12).

Let us denote the difference between the solution $\hat{x}[\cdot]$ of Equation (12) and the true underlying clustered signal $x[\cdot]$ by $\tilde{x}[\cdot] := \hat{x}[\cdot] - x[\cdot]$. Since $x[\cdot]$ satisfies Equation (4),

$$\|x[\cdot]\|_{\mathcal{E} \setminus \partial \mathcal{F}} = 0, \text{ and } \|\tilde{x}[\cdot]\|_{\mathcal{E} \setminus \partial \mathcal{F}} = \|\hat{x}[\cdot]\|_{\mathcal{E} \setminus \partial \mathcal{F}}. \tag{29}$$

Applying the decomposition property of the semi-norm $\|\cdot\|_{\mathrm{TV}}$ to Equation (28) yields

$$\sum_{i \in \mathcal{M}} |\hat{x}[i] - y[i]| + \lambda \|\hat{x}[\cdot]\|_{\mathcal{E} \setminus \partial \mathcal{F}} \leq \sum_{i \in \mathcal{M}} |e[i]| + \lambda \|x[\cdot]\|_{\partial \mathcal{F}} - \lambda \|\hat{x}[\cdot]\|_{\partial \mathcal{F}}. \tag{30}$$

Therefore, using Equation (29) and the triangle inequality,

$$\sum_{i \in \mathcal{M}} |\hat{x}[i] - y[i]| + \lambda \|\tilde{x}[\cdot]\|_{\mathcal{E} \setminus \partial \mathcal{F}} \leq \lambda \|\tilde{x}[\cdot]\|_{\partial \mathcal{F}} + \sum_{i \in \mathcal{M}} |e[i]|. \tag{31}$$

Since $\sum_{i \in \mathcal{M}} |\hat{x}[i] - y[i]| \geq 0$, Equation (31) yields

$$\lambda \|\tilde{x}[\cdot]\|_{\mathcal{E} \setminus \partial \mathcal{F}} \leq \lambda \|\tilde{x}[\cdot]\|_{\partial \mathcal{F}} + \sum_{i \in \mathcal{M}} |e[i]|, \tag{32}$$

i.e., for sufficiently small measurement noise $e[i]$, the signal differences of the recovery error $\tilde{x}[\cdot] = \hat{x}[\cdot] - x[\cdot]$ cannot be concentrated across the edges within the clusters $\mathcal{C}_l$. Moreover, using

$$\sum_{i\in\mathcal{M}} |\hat{x}[i] - y[i]| \overset{(9)}{=} \sum_{i\in\mathcal{M}} |\hat{x}[i] - x[i] - e[i]| \geq \sum_{i\in\mathcal{M}} |\tilde{x}[i]| - \sum_{i\in\mathcal{M}} |e[i]|, \quad (33)$$

the inequality Equation (31) becomes

$$\sum_{i\in\mathcal{M}} |\tilde{x}[i]| + \lambda\|\tilde{x}[\cdot]\|_{\mathcal{E}\setminus\partial\mathcal{F}} \leq \lambda\|\tilde{x}[\cdot]\|_{\partial\mathcal{F}} + 2\sum_{i\in\mathcal{M}} |e[i]|. \quad (34)$$

Thus, since the sampling set $\mathcal{M}$ satisfies the network compatibility condition, we can apply Equation (22) to $\tilde{\mathbf{x}}[\cdot]$ yielding

$$\sum_{i\in\mathcal{M}} |\tilde{x}[i]| + (1/K)\|\tilde{x}[\cdot]\|_{\mathcal{E}\setminus\partial\mathcal{F}} \geq (1/K)L\|\tilde{x}[\cdot]\|_{\partial\mathcal{F}}. \quad (35)$$

Inserting Equation (35) into Equation (34), with $\lambda = 1/K$, yields

$$\lambda(L-1)\|\tilde{x}[\cdot]\|_{\partial\mathcal{F}} \leq 2\sum_{i\in\mathcal{M}} |e[i]|. \quad (36)$$

Combining Equations (32) and (36) yields

$$\|\tilde{x}[\cdot]\|_{\mathrm{TV}} = \|\tilde{x}[\cdot]\|_{\mathcal{E}\setminus\partial\mathcal{F}} + \|\tilde{x}[\cdot]\|_{\partial\mathcal{F}} \overset{(32)}{\leq} 2\|\tilde{x}[\cdot]\|_{\partial\mathcal{F}} + (1/\lambda)\sum_{i\in\mathcal{M}} |e[i]| \overset{(36)}{\leq} \frac{1+4\lambda/(L-1)}{\lambda}\sum_{i\in\mathcal{M}} |e[i]|. \quad (37)$$

□

5.2. Proof of Theorem 3

Combine Lemma 6 with Lemma 7.

6. CONCLUSIONS

Given a known cluster structure of the data graph, we introduced the notion of resolving sampling sets. A sampling set resolves a cluster structure if there exists a sufficiently large network flow between the sampled nodes, with prescribed flow values over boundary edges which connect different clusters. Loosely speaking, this requires to choose the sampling set mainly in the boundary regions between different clusters in the data graph. Thus, we can leverage efficient clustering methods for identifying the cluster boundary regions in order to find sampling sets which resolve the intrinsic cluster structure of the network structure underlying a dataset.

The verification if a particular sampling set resolves a given partition requires to consider all possible sign patterns for the boundary edges, which is intractable for large graphs. An important avenue for follow-up work is the investigation if resolving sampling sets can be characterized easily using probabilistic models for the underlying network structure and sampling sets. Moreover, we plan to extend our analysis to nLasso methods using other loss functions, e.g., the squared error loss and also the logistic loss function in the context of classification problems.

AUTHOR NOTE

Parts of the work underlying this paper have been presented in Mara and Jung [43]. A preprint of this manuscript is available under https://arxiv.org/abs/1704.02107 [44].

AUTHOR CONTRIBUTIONS

AJ initiated the research and provided first proofs for the main results. NT helped with proof reading and pointing to some typos in the proofs. AM took care of the numerical experiments.

ACKNOWLEDGMENTS

The authors are grateful to Madelon Hulsebos for a careful proof-reading of an early manuscript. Moreover, the constructive comments of reviewers are appreciated sincerely. This manuscript is available as a pre-print at the following address: https://arxiv.org/abs/1704.02107.

REFERENCES

1. Sandryhaila A, Moura JMF. Classification via regularization on graphs. In *2013 IEEE Global Conference on Signal and Information Processing*. Austin, TX (2013). p. 495–8. doi: 10.1109/GlobalSIP.2013.6736923
2. Chen S, Sandryhaila A, Moura JMF, Kovačević J. Signal recovery on graphs: variation minimization. *IEEE Trans Signal Process.* (2015) **63**:4609–24. doi: 10.1109/TSP.2015.2441042
3. Bishop CM. *Pattern Recognition and Machine Learning*. New York, NY: Springer (2006).
4. Chapelle O, Schölkopf B, Zien A, (eds.). *Semi-Supervised Learning*. Cambridge, MA: The MIT Press (2006). doi: 10.7551/mitpress/9780262033589.001.0001
5. Zhou D, Schölkopf B. A regularization framework for learning from graph data. In: *ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields*, vol. 15. Banff (2004). p. 67–8.
6. Gadde A, Anis A, Ortega A. Active semi-supervised learning using sampling theory for graph signals. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '14* (2014). p. 492–501. doi: 10.1145/2623330.2623760
7. Ando RK, Zhang T. Learning on graph with laplacian regularization. In: *Advances in Neural Information Processing Systems*. Vancouver, BC (2007).
8. Belkin M, Niyogi P, Sindhwani V. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *J Mach Lear Res.* (2006) **7**:2399–434.

9. Sandryhaila A, Moura JMF. Big data analysis with signal processing on graphs: representation and processing of massive data sets with irregular structure. *IEEE Signal Process Mag.* (2014) **31**:80–90. doi: 10.1109/MSP.2014.2329213
10. Shuman DI, Narang SK, Frossard P, Ortega A, Vandergheynst P. The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process Mag*. (2013) **30**:83–98. doi: 10.1109/MSP.2012.2235192
11. Narang SK, Gadde A, Ortega A. Signal processing techniques for interpolation in graph structured data. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (2013). p. 5445–9. doi: 10.1109/ICASSP.2013.6638704
12. Hallac D, Leskovec J, Boyd S. Network Lasso: clustering and optimization in large graphs. In: *Proceedings of SIGKDD* (2015). p. 387–96. doi: 10.1145/2783258.2783313
13. Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY: Springer (2001).
14. Hastie T, Tibshirani R, Wainwright M. *Statistical Learning with Sparsity. The Lasso and its Generalizations*. Boca Raton FL: CRC Press (2015).
15. Yamada M, Koh T, Iwata T, Shawe-Taylor J, Kaski S. Localized lasso for high-dimensional regression. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Fort Laud-erdale, FL (2017). p. 325–33.
16. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J. *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. vol. 3 of Foundations and Trends in Machine Learning*. Hanover, MA: Now Publishers (2010).
17. Romero D, Ma M, Giannakis GB. Kernel-based reconstruction of graph signals. *IEEE Trans Signal Process.* (2017) **65**:764–78. doi: 10.1109/TSP.2016.2620116
18. Tsitsvero M, Barbarossa S, Lorenzo PD. Signals on graphs: uncertainty principle and sampling. *IEEE Trans Signal Process.* (2016) **64**:4845–60. doi: 10.1109/TSP.2016.2573748
19. Chen S, Varma R, Sandryhaila A, Kovačević J. Discrete signal processing on graphs: sampling theory. *IEEE Trans Signal Process.* (2015) **63**:6510–23. doi: 10.1109/TSP.2015.2469645
20. Chen S, Varma R, Singh A, Kovačević J. Signal recovery on graphs: fundamental limits of sampling strategies. *IEEE Trans Signal Inform Process Over Netw*. (2016) **2**:539–54. doi: 10.1109/TSIPN.2016.2614903
21. Segarra S, Marques AG, Leus G, Ribeiro A. Reconstruction of graph signals through percolation from seeding nodes. *IEEE Trans Signal Process.* (2016) **64**:4363–78. doi: 10.1109/TSP.2016.2552510
22. Wang X, Liu P, Gu Y. Local-set-based graph signal reconstruction. *IEEE Trans Signal Process.* (2015) **63**:2432–44. doi: 10.1109/TSP.2015.2411217
23. Sharpnack J, Rinaldo A, Singh A. *Sparsistency of the Edge Lasso over Graphs*. AIStats (JMLR WCP). La Palma (2012).
24. Wang YX, Sharpnack J, Smola AJ, Tibshirani RJ. Trend filtering on graphs. *J Mach Lear Res.* (2016) **17**:1–41. Available online at: http://jmlr.org/papers/v17/15-147.html
25. Golub GH, Van Loan CF. *Matrix Computations. 3rd Edn.* Baltimore, MD: Johns Hopkins University Press (1996).
26. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. Cambridge, MA: MIT Press (2016).
27. Cui S, Hero A, Luo ZQ, Moura JMF, (eds.). *Big Data Over Networks*. Cambridge, UK: Cambridge University Press (2016).
28. Zhu X, Rabbat M. Graph spectral compressed sensing for sensor networks. In: *Proceedings of IEEE ICASSP 2012*. Kyoto (2012). p. 2865–8.
29. Newman MEJ. *Networks: An Introduction*. New York, NY: Oxford University Press (2010).
30. Mossel E, Neeman J, Sly A. Stochastic block models and reconstruction. ArXiv e-prints. (2012).
31. Bapat RB. *Graphs and Matrices*. London: Springer-Verlag (2014).
32. Bühlmann P, van de Geer S. *Statistics for High-Dimensional Data*. New York, NY: Springer (2011).
33. Chambolle A, Pock T. An introduction to continuous optimization for imaging. *Acta Numer.* (2016) **25**:161–319. doi: 10.1017/S096249291600009X
34. Agarwal A, Negahban S, Wainwright MJ. Fast global convergence of gradient methods for high-dimensional statistical recovery. *Ann Stat.* (2012) **40**:2452–82. doi: 10.1214/12-AOS1032
35. Boyd S, Vandenberghe L. *Convex Optimization*. Cambridge: Cambridge University Press (2004).
36. Zhu Y. An augmented ADMM algorithm with application to the generalized lasso problem. *J Comput Graph Stat*. (2017) **26**:195–204. doi: 10.1080/10618600.2015.1114491
37. Kleinberg J, Tardos E. *Algorithm Design*. New York, NY: Addison Wesley (2006).
38. Spielman DA, hua Teng S. A local clustering algorithm for massive graphs and its application to nearly-linear time graph partitioning. arXiv:0809.3232 (2008).
39. Basirian S, Jung A. Random walk sampling for big data over networks. In: *Proceedings of International Conference on Sampling Theory and Applications*. Tallinn (2017).
40. Lancichinetti A, Fortunato S, Radicchi F. Benchmark graphs for testing community detection algorithms. *Phys Rev E* (2008) **78**:046110. doi: 10.1103/PhysRevE.78.046110
41. Zhu X, Ghahramani Z. *Learning from Labeled and Unlabeled Data with Label Propagation*. Technical Report CMU-CALD-02-107, Carnegie Mellon University (2002).
42. van de Geer SA, Bühlmann P. On the conditions used to prove oracle results for the Lasso. *Electron J Stat.* (2009) **3**:1360–92. doi: 10.1214/09-EJS506
43. Mara A, Jung A. Recovery conditions and sampling strategies for network lasso. In: *Proceedings of 51st Asilomar Conference Signals, Systems, Computers*. Pacific Grove, CA (2017).
44. Jung A, Tran NQ, Mara A. When is network lasso accurate? arXiv:170402107 (2017).

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.