Localized Linear Regression in Networked Data

Alexander Jung and Nguyen Tran

***Abstract*—The network Lasso (nLasso) has been proposed recently as an efficient learning algorithm for massive networked data sets (big data over networks). It extends the well-known least least absolute shrinkage and selection operator (Lasso) from learning sparse (generalized) linear models to network models. Efficient implementations of the nLasso have been obtained using convex optimization methods. These implementations naturally lend to highly scalable message passing methods. In this paper, we analyze the performance of nLasso when applied to localized linear regression problems involving networked data. Our main result is a sufficient conditions on the network structure and available label information such that nLasso accurately learns a localized linear regression model from few labeled data points.**

I. Introduction

The data arising in many important application domains can be modelled efficiently using some network structure. Examples of such networked data are found in signal processing where signal samples can be arranged as a chain, in image processing with pixels arranged on a grid, in wireless sensor networks where measurements conform to sensor proximity [1]–[4]. Organising data using networks is also used in knowledge bases (graphs) whose items are linked by relations [5], [6].

In what follows, we will represent networked data using an undirected "empirical graph". The nodes of the empirical graph represent individual data points (e.g., one image out of a image collection) which are are connected by edges according to some notion of similarity. This similarity might be induced naturally by the application at hand (e.g., friendship relations in social networks) or obtained from statistical models (probabilistic graphical models) [7], [8].

Beside their network structure, data points are typically characterized by features and labels. The features of data points are quantities that can be measured or computed efficiently (in an automated fashion). In contrast, the labels of data points are costly to acquire, involving human expert labour.

We consider regression problems within which data points are characterized by features and a numeric label (or target). The goal is to learn an accurate predictor which maps the features of a data point to a predicted label. The learning of the predictor is based on the availability of few data points with known labels. Facing partially labeled data is common since acquisition of reliable label information is often costly (involving human expert labor).

Learning an accurate predictor is particularly challenging in the high-dimensional regime, where the number of labeled data points (the training size) is much smaller than the length of the feature vectors. A particular example of an application falling into this regime is social profiling in large social network. Here, each data point represents an individual which is described my a multitude of features but reliable information about the personality can be obtained only for few individuals [9].

The learning of an accurate predictor from a small number of labeled data points is enabled by exploiting the tendency of well-connected data points to have similar statistical properties. Such a clustering assumption, which underlies most (semi-) supervised machine learning methods [10], [11], requires any reasonable predictor to be nearly constant over well connected subsets (clusters) of data points. The clustering assumption motivates the network Lasso (nLasso) as a form of empirical risk minimization [12].

The main contribution of this paper is a sufficient condition on the network topology and available label information such that the nLasso accurately learns a predictor from a small number of labeled data points. To this end we extend the network compatibility condition (NCC) introduced in [13] to vector-valued graph signals representing predictors. We demonstrate theoretically and empirically, via numerical experiments, that the NCC guarantees that nLasso learns an accurate predictor which conforms with the clustering hypothesis. Our theoretical findings can be used to design sampling schemes which identify those data points whose labels would provide most information about the labels of the other data points [4], [14].

Notation. We denote matrices and vectors using boldface upper and lower case letters. The identity matrix of size $d \times d$ is denoted $\mathbf{I}_d$. The positive part of some real number $a \in \mathbb{R}$ is $(a)_+ = \max\{a, 0\}$. The Euclidean norm of a vector $\mathbf{x} = (x_1, \ldots, x_p)^T$ is $\|\mathbf{x}\| := \sqrt{\sum_{r=1}^{p} x_r^2}$. For a positive definite matrix $\mathbf{C}$, we define the induced norm $\|\mathbf{x}\|_{\mathbf{C}} := \sqrt{\mathbf{x}^T \mathbf{C} \mathbf{x}}$. We will need the vector-valued clipping function $\mathcal{T}^{(\lambda)}(\mathbf{x}) := \lambda \mathbf{x}/\|\mathbf{x}\|$ for $\|\mathbf{x}\| \geq \lambda$ and $\mathcal{T}^{(\lambda)}(\mathbf{x}) := \mathbf{x}$ otherwise. The soft-thresholding operator is $\mathcal{S}(x; \tau) := \mathrm{sign}(x)(|x| - \tau)_+$.

II. Problem Formulation

We consider networked data which is represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ whose nodes $\mathcal{V}$ represent individual data points. The undirected edges $\mathcal{E}$ encode some domain-specific notion of similarity between data points. The extend of similarity between similar nodes $i, j \in \mathcal{V}$, connected by the edge $\{i, j\} \in \mathcal{E}$, is quantified by a positive edge weight A_{ij}. We collect the weights of all edges (setting $A_{ij} = 0$ if nodes $i, j \in \mathcal{V}$ are not connected by an edge), into the weight matrix $\mathbf{A} \in \mathbb{R}_+^{n \times n}$.

In addition to the graph structure $\mathcal{G}$, datasets typically convey additional information about the data points. Let us assume that each individual data point $i \in \mathcal{V}$ is characterized by a feature vectors $\mathbf{x}^{(i)} \in \mathbb{R}^p$ and a numeric label $y^{(i)} \in \mathbb{R}$. The features $\mathbf{x}^{(i)}$ can be determined easily for any data point. In contrast, acquisition of labels $y^{(i)}$ is difficult (requiring human

Authors are with the Department of Computer Science, Aalto University, Finland; firstname.lastname(at)aalto.fi

expert labour). Thus, we are interested in settings where we have access to the labels of a small subset, the training set, $\mathcal{M} = \{i_1, \ldots, i_m\} \subseteq \mathcal{V}$ of nodes only.

We relate the features $\mathbf{x}^{(i)}$ and labels $y^{(i)}$ of a data point using the linear model

$$y^{(i)} = \big(\overline{\mathbf{w}}^{(i)}\big)^T \mathbf{x}^{(i)} + \varepsilon^{(i)}, \tag{1}$$

with some true underlying weight vector $\overline{\mathbf{w}}^{(i)}$ for each node $i \in \mathcal{V}$. The noise component $\varepsilon^{(i)}$ in (1) summarizes any labelling our modelling errors. It is important to note that the weight vector $\overline{\mathbf{w}}^{(i)}$ in (1) can vary for different nodes $i \in \mathcal{V}$.

We will apply nLasso to the available labels $y^{(i)}$ for the training set to obtain an estimate $\widehat{\mathbf{w}}^{(i)}$ for the weight vector $\mathbf{w}^{(i)}$ at each node $i \in \mathcal{V}$. The estimates $\widehat{\mathbf{w}}^{(i)}$ define a predictor which maps the node $i \in \mathcal{V}$ to the predicted label

$$\hat{y}^{(i)} := \big(\widehat{\mathbf{w}}^{(i)}\big)^T \mathbf{x}^{(i)}. \tag{2}$$

The predictions $\hat{y}^{(i)}$ will be accurate, i.e., the prediction error $\hat{y}^{(i)} - y^{(i)}$ will be small, if the estimation error $\overline{\mathbf{w}}^{(i)} - \widehat{\mathbf{w}}^{(i)}$ is small. Our main result (see Theorem 2) provides a sufficient condition on the structure of the empirical graph $\mathcal{G}$ and the training set $\mathcal{M}$ such that the estimation error is small.

We find it convenient to interpret the weight vectors $\mathbf{w}^{(i)}$ as the values of a graph signal $\mathbf{w} : \mathcal{V} \to \mathbb{R}^p$ which assigns each node $i \in \mathcal{V}$ the vector $\mathbf{w}^{(i)} \in \mathbb{R}^p$. The set of all vector-valued graph signals is denoted

$$\mathcal{W} := \{\mathbf{w} : \mathcal{V} \to \mathbb{R}^p : i \mapsto \mathbf{w}^{(i)}\}. \tag{3}$$

Each graph signal $\widehat{\mathbf{w}} \in \mathcal{W}$ represents a predictor which maps a node with features $\mathbf{x}^{(i)}$ to the predicted label (2).

Given partially labeled networked data, we aim at leaning a classifier $\widehat{\mathbf{w}} \in \mathcal{W}$ whose predictions (2) agree with the labels $y^{(i)}$ of labeled data points in the training set $\mathcal{M}$. In particular, we aim at learning a classifier having a small training error

$$\widehat{E}(\widehat{\mathbf{w}}) := \sum_{i \in \mathcal{M}} \big|y^{(i)} - \hat{y}^{(i)}\big| \overset{(2)}{=} \sum_{i \in \mathcal{M}} \big|y^{(i)} - \big(\widehat{\mathbf{w}}^{(i)}\big)^T \mathbf{x}^{(i)}\big|. \tag{4}$$

We use the absolute value loss since it somewhat simplifies our analysis. However, we expect no big challenges in extending our analysis to nLasso using different loss functions, such as the squared error loss. The absolute value loss is actually preferred for learning linear regression models (1) when the noise $\varepsilon^{(i)}$ is expected to contain only few large values, known as "salt and pepper" noise in image processing [15, Example 2.3].

III. Network Lasso

The criterion (4) by itself is not enough for guiding the learning of a predictor $\mathbf{w}$ since (4) completely ignores the weights $\widehat{\mathbf{w}}^{(i)}$ at unlabeled nodes $i \in \mathcal{V} \setminus \mathcal{M}$. Therefore, we need to impose some additional structure on the predictor $\widehat{\mathbf{w}}$. To this end, we require the classifier $\widehat{\mathbf{w}}$ to conform with the *cluster structure* of the empirical graph $\mathcal{G}$ [16], [17].

The extend by which a classifier $\widehat{\mathbf{w}} \in \mathcal{W}$ conforms with $\mathcal{G}$ by the total variation (TV)

$$\|\mathbf{w}\|_{\rm TV} := \sum_{\{i,j\} \in \mathcal{E}} A_{ij} \|\mathbf{w}^{(j)} - \mathbf{w}^{(i)}\|. \tag{5}$$

If the weights $\mathbf{w}^{(i)}$ are approximately constant over well connected subsets of nodes, the classifier $\mathbf{w} \in \mathcal{W}$ has small TV $\|\widehat{\mathbf{w}}\|_{\rm TV}$. The restriction of (5) to a subset $\mathcal{S} \subseteq \mathcal{E}$ of edges is denoted $\|\mathbf{w}\|_{\mathcal{S}} := \sum_{\{i,j\} \in \mathcal{S}} A_{ij} \|\mathbf{w}^{(j)} - \mathbf{w}^{(i)}\|$.

We are led naturally to learning a classifier $\widehat{\mathbf{w}}$ via the *regularized empirical risk minimization* (ERM)

$$\widehat{\mathbf{w}} \in \operatorname*{arg\,min}_{\mathbf{w} \in \mathcal{W}} \widehat{E}(\mathbf{w}) + \lambda \|\mathbf{w}\|_{\rm TV}, \tag{6}$$

which is a special case of nLasso [12]. The parameter $\lambda > 0$ in (6) allows to trade small TV $\|\widehat{\mathbf{w}}\|_{\rm TV}$ against small error $\widehat{E}(\widehat{\mathbf{w}})$ (4). The choice of λ can be guided by cross validation [18]. Alternatively the choice of λ can be guided by our analysis of the nLasso estimation error (see the discussion after Theorem 2).

Note that nLasso (6) does not enforce the labels $y^{(i)}$ themselves to be clustered. Instead, it requires the classifier $\widehat{\mathbf{w}}$, which is used to obtain predictions (2), to be clustered.

It will be convenient to reformulate (6) using vector notation. To this end, we represent a graph signal $\mathbf{w} \in \mathcal{W}$ as the vector

$$\mathbf{w} = ((\mathbf{w}^{(1)})^T, \ldots, (\mathbf{w}^{(n)})^T)^T \in \mathbb{R}^{pn}. \tag{7}$$

and define the partitioned matrix $\mathbf{D} \in \mathbb{R}^{pq \times pn}$

$$\mathbf{D}_{e,i} = \begin{cases} A_{ij} \mathbf{I}_p & e = \{i,j\}, i < j \\ -A_{ij} \mathbf{I}_p & e = \{i,j\}, i > j \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{8}$$

Applying the matrix $\mathbf{D}$ to a graph signal vector $\mathbf{w}$ (7) results in a partitioned vector $\mathbf{D}\mathbf{w}$ whose eth block is given by $A_{ij}(\mathbf{w}^{(i)} - \mathbf{w}^{(j)})$ (see (5)). Using (7) and (8), we can reformulate the nLasso (6) as

$$\widehat{\mathbf{w}} \in \operatorname*{arg\,min}_{\mathbf{w} \in \mathbb{R}^{pn}} h(\mathbf{w}) + g(\mathbf{D}\mathbf{w}). \tag{9}$$

Here,

$$h(\mathbf{w}) = \widehat{E}(\mathbf{w}) \text{ and } g(\mathbf{u}) := \lambda \sum_{e=1}^{q} \|\mathbf{u}^{(e)}\| \tag{10}$$

with $\mathbf{u} = \big(\big(\mathbf{u}^{(1)}\big)^T, \ldots, \big(\mathbf{u}^{(q)}\big)^T\big)^T \in \mathbb{R}^{pq}$.

IV. Primal-Dual Method

The nLasso (9) is a convex optimization problem with a non-smooth objective function which rules out the use of gradient descent methods [19]. However, the objective function is highly structured since it is the sum of two components $h(\mathbf{w})$ and $g(\mathbf{D}\mathbf{w})$, which can be optimized efficiently when considered separately. Such composite functions can be optimized efficienty using proximal splitting methods [20]–[22].

We solve (9) using a particular proximal splitting method which is based on reformulating the problem (9) as a saddle-point problem [23]

$$\min_{\mathbf{w} \in \mathbb{R}^{pn}} \max_{\mathbf{u} \in \mathbb{R}^{pq}} \mathbf{u}^T \mathbf{D}\mathbf{w} + h(\mathbf{w}) - g^*(\mathbf{u}), \tag{11}$$

with the convex conjugate g^* of g [22].

Solutions $(\widehat{\mathbf{w}}, \widehat{\mathbf{u}})$ of (11) are characterized by [24, Thm 31.3]

$$-\mathbf{D}^T \widehat{\mathbf{u}} \in \partial h(\widehat{\mathbf{w}}), \text{ and } \mathbf{D}\widehat{\mathbf{w}} \in \partial g^*(\widehat{\mathbf{u}}). \tag{12}$$

The coupled conditions (12) are, in turn, equivalent to

$$\widehat{\mathbf{w}}-\mathbf{T}\mathbf{D}^T\widehat{\mathbf{u}}\in(\mathbf{I}+\mathbf{T}\partial h)(\widehat{\mathbf{w}}),\ \widehat{\mathbf{u}}+\mathbf{\Sigma}\mathbf{D}\widehat{\mathbf{w}}\in(\mathbf{I}+\mathbf{\Sigma}\partial g^*)(\widehat{\mathbf{u}}),\ (13)$$

with positive definite matrices $\mathbf{\Sigma}\in\mathbb{R}^{pq\times pq},\mathbf{T}\in\mathbb{R}^{pn\times pn}$. In principle, the matrices $\mathbf{\Sigma},\mathbf{T}$ in (13) can be chosen arbitrarily. However, it will prove convenient to choose them as

$$\mathbf{\Sigma}=\mathrm{diag}\{\sigma^{(e)}\mathbf{I}_p\}_{e=1}^q \text{ and } \mathbf{T}=\mathrm{diag}\{\tau^{(i)}\mathbf{I}_p\}_{i=1}^n \quad (14)$$

with scalars $\{\sigma^{(e)}\}_{e=1}^q$ and $\{\tau^{(i)}\}_{i\in\mathcal{V}}$ as specified below.

The optimality condition (13) for nLasso (9) lends naturally to the following coupled fixed point iterations [23]

$$\mathbf{w}_{k+1}=(\mathbf{I}+\mathbf{T}\partial h)^{-1}(\mathbf{w}_k-\mathbf{T}\mathbf{D}^T\mathbf{u}_k) \quad (15)$$

$$\mathbf{u}_{k+1}=(\mathbf{I}+\mathbf{\Sigma}\partial g^*)^{-1}(\mathbf{u}_k+\mathbf{\Sigma}\mathbf{D}(2\mathbf{w}_{k+1}-\mathbf{w}_k)). \quad (16)$$

The update (16) involves the resolvent operator

$$(\mathbf{I}+\mathbf{\Sigma}\partial g^*)^{-1}(\mathbf{u})=\operatorname*{arg\,min}_{\mathbf{u}'\in\mathbb{R}^{pq}} g^*(\mathbf{u}')+(1/2)\|\mathbf{u}'-\mathbf{u}\|^2_{\mathbf{\Sigma}^{-1}}. \quad (17)$$

The convex conjugate g^* of g (see (10)) can be decomposed as $g^*(\mathbf{v})=\sum_{e=1}^q g_2^*(\mathbf{v}^{(e)})$ with the convex conjugate g_2^* of $g_2(\mathbf{z}):=\lambda\|\mathbf{z}\|$. Combining the fact that $\mathbf{\Sigma}$ is a block diagonal matrix with the Moreau decomposition [25, Sec. 6.5], it can be shown that $\mathbf{c}=(\mathbf{I}_{pq}+\mathbf{\Sigma}\partial g^*)^{-1}(\mathbf{u})$ (see (17)) with

$$\mathbf{c}=\big(\big(\mathbf{c}^{(1)}\big)^T,\ldots,\big(\mathbf{c}^{(q)}\big)^T\big)^T,\ \mathbf{c}^{(e)}=\mathcal{T}^{(\lambda)}\big(\mathbf{u}^{(e)}\big). \quad (18)$$

Similar to the update (16), also the update (15) decomposes into independent updates of the weight vectors

$$\mathbf{w}^{(i)}=\mathbf{w}_k^{(i)}-\sum_{j\succ i}\tau^{(j)}A_{i,j}\mathbf{u}_k^{(j)}+\sum_{i\succ j}\tau^{(j)}A_{i,j}\mathbf{u}_k^{(j)}$$

yielding the updated weight vectors $\mathbf{w}_{k+1}^{(i)}=\mathbf{v}^{(i)}$ for each node $i\in\mathcal{V}$. In particular, for unlabeled nodes $i\notin\mathcal{M}$, the update (15) reduces to $\mathbf{v}^{(i)}=\mathbf{w}^{(i)}$. For labeled nodes $i\in\mathcal{M}$, using elementary sub-gradient calculus, we obtain

$$\begin{aligned}\mathbf{v}^{(i)}&=\mathbf{x}^{(i)}(\tilde{y}+\mathcal{S}(\tilde{w}-\tilde{y};\tau^{(i)}))\\&\quad+(\mathbf{I}-(1/\|\mathbf{x}^{(i)}\|^2)\mathbf{x}^{(i)}\big(\mathbf{x}^{(i)}\big)^T)\mathbf{w}^{(i)}\end{aligned} \quad (19)$$

with $\tilde{y}:=y^{(i)}/\|\mathbf{x}^{(i)}\|^2$ and $\tilde{w}:=\big(\mathbf{w}^{(i)}\big)^T\mathbf{x}^{(i)}/\|\mathbf{x}^{(i)}\|^2$. Inserting (19) and (18) into the fixed point iteration (16) results in Algorithm 1 for solving the nLasso (9).

If the matrices $\mathbf{\Sigma}$ and $\mathbf{T}$ using in (16) satisfy

$$\|\mathbf{\Sigma}^{1/2}\mathbf{D}\mathbf{T}^{1/2}\|^2<1, \quad (20)$$

the sequences obtained from iterating (15) and (16) converge to a saddle point of the problem (11) [23, Thm. 1]. The condition (20) is ensured by choosing $\mathbf{\Sigma}$ and $\mathbf{T}$ according to (14) using $\sigma^{(e)}=1/(2A_e)$ and $\tau^{(i)}:=\eta/d^{(i)}$, with (weighted) node degree $d^{(i)}=\sum_{j\neq i}A_{i,j}$ and some constant $\eta<1$ [23, Lem. 2].

V. Error Analysis for nLasso

In order to analyze the statistical properties of Algorithm 1 we need to understand the structure of the solutions to the nLasso problem (9). To this end, will use a simple but useful model of piece-wise constant weight vectors [26], [27]

$$\overline{\mathbf{w}}^{(i)}=\sum_{l=1}^F\mathbf{a}^{(l)}\mathcal{I}_{\mathcal{C}^{(l)}}[i]. \quad (21)$$

with fixed vectors $\mathbf{a}^{(l)}\in\mathbb{R}^p$, for $l=1,\ldots,F$, and the indicator function $\mathcal{I}_{\mathcal{C}}[i]\in\{0,1\}$ with $\mathcal{I}_{\mathcal{C}}[i]=1$ if and only if $i\in\mathcal{C}\subseteq\mathcal{V}$. Here, we use a partition $\mathcal{F}=\{\mathcal{C}^{(1)},\ldots,\mathcal{C}^{(F)}\}$ of the nodes $\mathcal{V}$ in the empirical graph into disjoint subsets (clusters) $\mathcal{C}^{(l)}$.

The model (21), which generalizes the piece-wise constant signal model [], embodies a clustering assumption that well-connected nodes in the empirical graph should have similar relations between features and labels [16], [17].

While our analysis allows for an arbitrary choice of clusters $\mathcal{C}^{(l)}$ in (21), our results are most useful when the node sets $\mathcal{C}^{(l)}$ reflect the intrinsic cluster structure of the empirical graph $\mathcal{G}$. In particular, we expect the clusters $\mathcal{C}^{(l)}$ to be such that the TV $\|\overline{\mathbf{w}}\|_{\mathrm{TV}}$ (see (5)) is small.

We now introduce the network compatibility condition (NCC), which generalizes the compatibility conditions for Lasso type estimators [28] of ordinary sparse signals. Our main contribution is to show that the NCC guarantees accuracy of the nLasso (9) solutions, as obtained using Algorithm 1.

Algorithm 1 nLasso via primal-dual method

Input: $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathbf{A})$, $\{\mathbf{x}^{(i)}\}_{i\in\mathcal{V}}$, $\mathcal{M}$, $\{y^{(i)}\}_{i\in\mathcal{M}}$, λ,
Initialize: $k:=0$, $\widehat{\mathbf{w}}_0:=0$, $\widehat{\mathbf{u}}_0:=0$; $\mathbf{\Sigma}$ and $\mathbf{T}$ using (14) with $\sigma^{(e)}=1/(2A_e)$, $\tau^{(i)}=0.9/d^{(i)}$; $\beta_i:=\tau^{(i)}/|\mathcal{M}|$; indicence matrix $\mathbf{D}$ according to (8)

```
repeat
    w_{k+1} := w_k - T D^T u_k
    for each labeled node i ∈ M do
        w^{(i)}_{k+1} := v^{(i)} using (19)
    end for
    ū := u_k + Σ D(2 w_{k+1} - w_k)
    u^{(e)}_{k+1} = T^{(λ)}(ū^{(e)}) for every edge e ∈ E
    k := k+1
until stopping criterion is satisfied
```

Output: predictor $\widehat{\mathbf{w}}:=\widehat{\mathbf{w}}_k$

Definition 1. *Consider a networked dataset with empirical graph $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathbf{A})$. The nodes are characterized by feature vectors $\mathbf{x}^{(i)}\in\mathbb{R}^p$ and grouped according to a fixed partition $\mathcal{F}=\{\mathcal{C}^{(1)},\ldots,\mathcal{C}^{(F)}\}$. The labels $y^{(i)}$ of nodes are observed only on the training set $\mathcal{M}\subseteq\mathcal{V}$. The training set is said to satisfy NCC, with constants $K,L>0$, if*

$$K\sum_{i\in\mathcal{M}}\big|\big(\mathbf{x}^{(i)}\big)^T\mathbf{w}^{(i)}\big|+\|\mathbf{w}\|_{\overline{\partial\mathcal{F}}}\geq(L/\sqrt{p})\|\mathbf{w}\|_{\partial\mathcal{F}} \quad (22)$$

for any graph signal $\mathbf{w}\in\mathcal{W}$ (see (3)).

It turns out that, if the sampling set satisfies the NCC, any solution of (6) provides an accurate estimate of the true underlying weight vectors (see (1) and (21)).

Theorem 2. *Consider a partially labeled networked dataset with empirical graph $\mathcal{G}$ with features $\mathbf{x}^{(i)}$ known for all nodes and labels $y^{(i)}$ which are known only for the nodes $i\in\mathcal{M}$.*

We assume a linear model (1) *with true weights* $\overline{\mathbf{w}}^{(i)}$ *piece-wise constant* (21)*. If the sampling set* $\mathcal{M}$ *satisfies NCC with parameters* $L > \sqrt{p}$ *and* $K > 0$*, then any solution* $\widehat{\mathbf{w}}$ *of nLasso* (9) *with the choice* $\lambda := 1/K$ *satisfies*

$$\|\widehat{\mathbf{w}} - \overline{\mathbf{w}}\|_{\rm TV} \leq K(1+4\sqrt{p}/(L-\sqrt{p})) \sum_{i \in \mathcal{M}} |\boldsymbol{\varepsilon}^{(i)}|. \quad (23)$$

According to Theorem 2, the choice for the nLasso parameter λ in (9) can be based on the NCC constant K (see (22)) via setting $\lambda = 1/K$. For this choice, given the training set $\mathcal{M}$ satisfies the NCC with parameters K and L, the nLasso error $\widehat{\mathbf{w}} - \overline{\mathbf{w}}$ is bounded according to (23).

We highlight that the nLasso (6) does not require the partition $\mathcal{F}$ used for our signal model (21). This partition is only used for the analysis of nLasso (6). Moreover, if the true underlying graph signal is of the form (21) and nLasso accurately learns this signal (Theorem 2), we can obtain the partition $\mathcal{F}$ by thresholding the edge-wise differences $\|\mathbf{w}^{(i)} - \mathbf{w}^{(j)}\|$ for $\{i,j\} \in \mathcal{E}$ [29].

VI. Numerical Experiments

We generate an empirical graph $\mathcal{G}$ of size $n = 80$ by first randomly generating two clusters $\mathcal{C}^{(1)}$ and $\mathcal{C}_2$ of size $n/2$ with average node degree 10. We then connected the two clusters $\mathcal{C}^{(1)}$ and $\mathcal{C}^{(2)}$ by randomly placing edges between them.

The nodes of the empirical graph $\mathcal{G}$ are assigned feature vectors $\mathbf{x}^{(i)} \in \mathbb{R}^2$ obtained by i.i.d. random vectors uniformly distributed on the unit sphere $\{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x}\| = 1\}$. The labels $y^{(i)}$ of the nodes $i \in \mathcal{V}$ are generated according to the linear model (1) with zero noise $\varepsilon^{(i)} = 0$ and piecewise constant weight vectors $\mathbf{w}^{(i)}$ (see (21)). We assume that the labels $y^{(i)}$ are known for the nodes in the training set which includes three data points of each cluster, i.e., $|\mathcal{M} \cap \mathcal{C}^{(1)}| = |\mathcal{M} \cap \mathcal{C}^{(2)}| = 3$.

Using [13, Lemma 6] it can be shown that the training set $\mathcal{M}$ satisfies NCC with $L > \sqrt{p} = \sqrt{2}$ if there exists a sufficiently large network flow between the labeled node $i \in \mathcal{C}^{(l)} \cap \mathcal{M}$ and the boundary edges $\partial := \{\{i,j\} \in \mathcal{E} : i \in \mathcal{C}^{(1)}, j \in \mathcal{C}^{(2)}\}$ between the two clusters. In particular, let $\rho^{(l)}$ denote the normalized flow value from the labeled nodes in cluster $\mathcal{C}^{(l)}$ and the cluster boundary, normalized by the boundary size $|\partial|$. The NCC is satisfied with $L > \sqrt{2}$ if $\rho^{(l)} > 2$ for $l = 1, 2$.

In Figure 1, we depict the normalized mean squared error (NMSE) $\varepsilon := \|\overline{\mathbf{w}} - \widehat{\mathbf{w}}\|_2^2 / \|\overline{\mathbf{w}}\|_2^2$ incurred by Algorithm 1 (averaged over 10 i.i.d. simulation runs) for varying connectivity, as measured by the empirical average $\bar{\rho}$ of $\rho^{(1)}$ and $\rho^{(2)}$ (which have the same distribution due to the symmetric graph construction). Figure 1 agrees with our analysis which predicts that nLasso Algorithm 1 is accurate if clusters $\mathcal{C}^{(1)}$, $\mathcal{C}^{(2)}$ are well connected such that $\rho^{(1)}, \rho^{(2)} > \sqrt{2}$.

VII. Proof of Theorem 2

In order to proof Theorem 2, we consider an arbitrary but fixed nLasso solution $\widehat{\mathbf{w}} = \big(\big(\widehat{\mathbf{w}}^{(1)}\big)^T, \ldots, \big(\widehat{\mathbf{w}}^{(n)}\big)^T\big)^T$ (see (9)) and denote the estimation error between $\widehat{\mathbf{w}}^{(i)}$ and the true underlying weights $\overline{\mathbf{w}}^{(i)}$ (see (1)) as $\widetilde{\mathbf{w}}^{(i)} := \widehat{\mathbf{w}}^{(i)} - \overline{\mathbf{w}}^{(i)}$.

By the definition of nLasso (6),

$$\sum_{i \in \mathcal{M}} |\hat{y}^{(i)} - y^{(i)}| + \lambda \|\widehat{\mathbf{w}}\|_{\rm TV} \leq \sum_{i \in \mathcal{M}} |\boldsymbol{\varepsilon}^{(i)}|_1 + \lambda \|\overline{\mathbf{w}}\|_{\rm TV}. \quad (24)$$

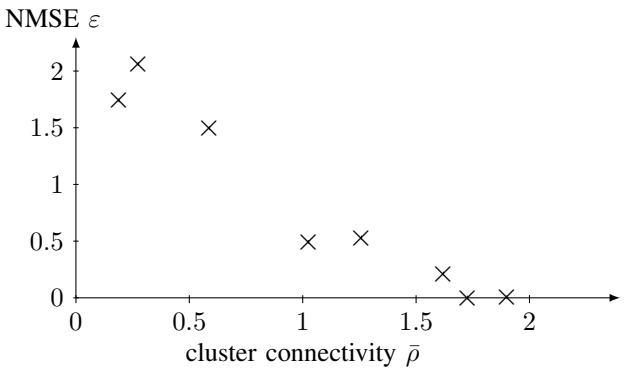

Fig. 1. NMSE achieved by Algorithm 1 for a two-cluster graph with varying connectivity $\bar{\rho}$ between the clusters.

Since the true weight vectors $\overline{\mathbf{w}}^{(i)}$ are piece-wise constant (see (21)), $\|\overline{\mathbf{w}}\|_{\overline{\partial \mathcal{F}}} = 0$ and $\|\widetilde{\mathbf{w}}\|_{\overline{\partial \mathcal{F}}} = \|\widehat{\mathbf{w}}\|_{\overline{\partial \mathcal{F}}}$. Using the decomposition property and triangle inequality for the TV in (24),

$$\sum_{i \in \mathcal{M}} |\hat{y}^{(i)} - y^{(i)}| + \lambda \|\widehat{\mathbf{w}}\|_{\overline{\partial \mathcal{F}}} \leq \sum_{i \in \mathcal{M}} |\boldsymbol{\varepsilon}^{(i)}| + \lambda \|\overline{\mathbf{w}}\|_{\partial \mathcal{F}} - \lambda \|\widehat{\mathbf{w}}\|_{\partial \mathcal{F}}$$

and, in turn,

$$\sum_{i \in \mathcal{M}} |\hat{y}^{(i)} - y^{(i)}| + \lambda \|\widetilde{\mathbf{w}}\|_{\overline{\partial \mathcal{F}}} \leq \sum_{i \in \mathcal{M}} |\boldsymbol{\varepsilon}^{(i)}| + \lambda \|\widetilde{\mathbf{w}}\|_{\partial \mathcal{F}}. \quad (25)$$

We conclude from (25) that

$$\|\widetilde{\mathbf{w}}\|_{\overline{\partial \mathcal{F}}} \leq (1/\lambda) \sum_{i \in \mathcal{M}} \big|\boldsymbol{\varepsilon}^{(i)}\big| + \|\widetilde{\mathbf{w}}\|_{\partial \mathcal{F}}, \quad (26)$$

Thus, for small noise $\boldsymbol{\varepsilon}^{(i)}$ (see (1)), the nLasso estimation error $\widetilde{\mathbf{w}}$ is piece-wise constant. However, it remains to control the size of the error for which we will invoke the NCC 22.

We can develop the LHS of (25) as

$$\begin{aligned} \sum_{i \in \mathcal{M}} |\hat{y}^{(i)} - y^{(i)}| &\overset{(1),(2)}{=} \sum_{i \in \mathcal{M}} \big|\big(\mathbf{x}^{(i)}\big)^T \widetilde{\mathbf{w}}^{(i)} - \boldsymbol{\varepsilon}^{(i)}\big| \\ &\geq \sum_{i \in \mathcal{M}} \big|\big(\mathbf{x}^{(i)}\big)^T \widetilde{\mathbf{w}}^{(i)}\big| - \sum_{i \in \mathcal{M}} |\boldsymbol{\varepsilon}^{(i)}|, \quad (27) \end{aligned}$$

where we have used the triangle inequality in the last step. Combining (27) with (25),

$$\sum_{i \in \mathcal{M}} \big|\big(\mathbf{x}^{(i)}\big)^T \widetilde{\mathbf{w}}^{(i)}\big| + \lambda \|\widetilde{\mathbf{w}}\|_{\overline{\partial \mathcal{F}}} \leq 2 \sum_{i \in \mathcal{M}} |\boldsymbol{\varepsilon}^{(i)}| + \lambda \|\widetilde{\mathbf{w}}\|_{\partial \mathcal{F}}. \quad (28)$$

Since we assume NNC holds for $\mathcal{M}$, (22) yields

$$(L/\sqrt{p}) \|\widetilde{\mathbf{w}}\|_{\partial \mathcal{F}} \leq K \sum_{i \in \mathcal{M}} \big|\big(\mathbf{x}^{(i)}\big)^T \widetilde{\mathbf{w}}^{(i)}\big| + \|\widetilde{\mathbf{w}}\|_{\overline{\partial \mathcal{F}}}. \quad (29)$$

Inserting (29) into (28) and using $\lambda := 1/K$, yields

$$\lambda (L/\sqrt{p} - 1) \|\widetilde{\mathbf{w}}\|_{\partial \mathcal{F}} \leq 2 \sum_{i \in \mathcal{M}} |\boldsymbol{\varepsilon}^{(i)}|. \quad (30)$$

Combining (26) with (30) yields

$$\begin{aligned} \|\widetilde{\mathbf{w}}\|_{\rm TV} &= \|\widetilde{\mathbf{w}}\|_{\overline{\partial \mathcal{F}}} + \|\widetilde{\mathbf{w}}\|_{\partial \mathcal{F}} \\ &\overset{(26)}{\leq} (1/\lambda) \sum_{i \in \mathcal{M}} |\boldsymbol{\varepsilon}^{(i)}|_1 + 2 \|\widetilde{\mathbf{w}}\|_{\partial \mathcal{F}} \\ &\overset{(30)}{\leq} ((1/\lambda) + \frac{4\sqrt{p}/\lambda}{(L - \sqrt{p})}) \sum_{i \in \mathcal{M}} |\boldsymbol{\varepsilon}^{(i)}|. \end{aligned}$$

REFERENCES

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

[2] S. G. Mallat, *A Wavelet Tour of Signal Processing – The Sparse Way*, 3rd ed. San Diego, CA: Academic Press, 2009.

[3] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 1998.

[4] L. F. O. Chamon and A. Ribeiro, "Greedy sampling of graph signals," *2018*, vol. 66, no. 1, pp. 34–47, 2018.

[5] D. Vrandečić and M. Krötzsch, "Wikidata: A free collaborative knowledgebase," *Commun. ACM*, vol. 57, no. 10, pp. 78–85, Sep. 2014.

[6] A. Sadeghi, C. Lange, M. Vidal, and S. Auer, "Communication metadata using knowledge graphs," in *Lecture Notes in Computer Science*. Springer, 2017.

[7] N. Q. Tran and A. Jung, "Learning conditional independence structure for high-dimensional uncorrelated vector processes," New Orleans (LA), 2017, pp. 5920–5924.

[8] D. Koller, N., and Friedman, *Probabilistic Graphical Models: Principles and Techniques*, ser. Adaptive computation and machine learning. MIT Press, 2009.

[9] R. Gross and A.Acquisti, "Information revelation and privacy in online social networks," in *Workshop on Privacy in the Electronic Society (WPES)*, 2005.

[10] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[11] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. Cambridge, Massachusetts: The MIT Press, 2006.

[12] D. Hallac, J. Leskovec, and S. Boyd, "Network lasso: Clustering and optimization in large graphs," in *Proc. SIGKDD*, 2015, pp. 387–396.

[13] A. Jung, N. Quang, and A. Mara, "When is Network Lasso Accurate?" *Front. Appl. Math. Stat.*, vol. 3, Jan. 2018.

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

[15] A. Chambolle and T. Pock, "An introduction to continuous optimization for imaging," *Acta Numer.*, vol. 25, pp. 161–319, 2016. [Online]. Available: http://dx.doi.org/10.1017/S096249291600009X

[16] M. E. J. Newman, *Networks: An Introduction*. Oxford Univ. Press, 2010.

[17] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, "Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications," *Phys. Rev. E*, vol. 84, 2011.

[18] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY, USA: Springer, 2001.

[19] A. Jung, "A fixed-point of view on gradient methods for big data," *Frontiers in Applied Mathematics and Statistics*, vol. 3, 2017. [Online]. Available: https://www.frontiersin.org/article/10.3389/fams.2017.00018

[20] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," *ArXiv e-prints*, 2009.

[21] D. O'Connor and L. Vandenberghe, "Primal-dual decomposition by operator splitting and applications to image deblurring," *SIAM Journal on Imaging Sciences*, vol. 7, no. 3, pp. 1724–1754, 2014.

[22] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imag. Vis.*, vol. 40, no. 1, 2011. [Online]. Available: http://dx.doi.org/10.1007/s10851-010-0251-1

[23] T. Pock and A. Chambolle, "Diagonal preconditioning for first order primal-dual algorithms in convex optimization," in *IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, Nov. 2011.

[24] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ: Princeton Univ. Press, 1970.

[25] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2013.

[26] Z. Fan and L. Guan, "Approximate l0-penalized estimation of piecewise-constant signals on graphs," *arxiv*, 2017.

[27] S. Chen, R. Varma, A. Singh, and J. Kovačević, "Representations of piecewise smooth signals on graphs," in *Proc. IEEE ICASSP 2016*, Shanghai, CN, March 2016, pp. 6370–6374.

[28] P. Bühlmann and S. van de Geer, *Statistics for High-Dimensional Data*. New York: Springer, 2011.

[29] Y.-X. Wang, J. Sharpnack, A. J. Smola, and R. J. Tibshirani, "Trend filtering on graphs," *J. Mach. Lear. Research*, vol. 17, 2016.