Jung, Alexander

Networked Exponential Families for Big Data over Networks

Published in:
IEEE Access

DOI:
10.1109/ACCESS.2020.3033817

Published: 01/01/2020

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

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 20, 2020, accepted October 18, 2020, date of publication October 26, 2020,
date of current version November 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3033817

Networked Exponential Families for Big Data Over Networks

ALEXANDER JUNG, (Member, IEEE)
Department of Computer Science, Aalto University, FI-00076 Aalto, Finland
e-mail: alex.jung@aalto.fi

ABSTRACT The data generated in many application domains can be modeled as big data over networks, i.e., massive collections of high-dimensional local datasets related via an intrinsic network structure. Machine learning for big data over networks must jointly leverage the information contained in the local datasets and their network structure. We propose networked exponential families as a novel probabilistic modeling framework for machine learning from big data over networks. We interpret the high-dimensional local datasets as the realizations of a random process distributed according to some exponential family. Networked exponential families allow us to jointly leverage the information contained in local datasets and their network structure in order to learn a tailored model for each local dataset. We formulate the task of learning the parameters of networked exponential families as a convex optimization problem. This optimization problem is an instance of the network Lasso and enforces a data-driven pooling (or clustering) of the local datasets according to their corresponding parameters for the exponential family. We derive an upper bound on the estimation error of network Lasso. This upper bound depends on the network structure and the information geometry of the node-wise exponential families. These insights provided by this bound can be used for determining how much data needs to be collected or observed to ensure network Lasso to be accurate. We also provide a scalable implementation of the network Lasso as a message-passing between adjacent local datasets. Such message passing is appealing for federated machine learning relying on edge computing. We finally note that the proposed method is also privacy-preserving because no raw data but only parameter (estimates) are shared among different nodes.

INDEX TERMS Big data, networks, statistical machine learning, federated learning, privacy-preserving machine learning, lasso.

I. INTRODUCTION

The data generated in many important application domains have an intrinsic network structure. Networked data arises in the study of social networks, natural language processing and personalized medicine [3], [9], [51]. Most existing network science provides powerful tools for the analysis of such data based solely on its intrinsic network structure [14], [38].

We consider networked data where each node of the network represents a local dataset or high-dimensional data point. To study disease spread, we represent individuals by nodes in a network whose links indicate physical proximity as relevant for disease transmission [38, Ch. 17]. Existing compartment models consider all individuals to behave (structurally) similar, somewhat like an ''i.i.d. assumption'' in statistical learning theory. However, the tendency to get infected might strongly depend on personal factors such as current health status. These personal factors can be characterized via a plethora of attributes, ranging from healthcare records up to the recent travel history [35]. In natural language processing, text corpora are represented as networks of documents that are connected via co-authorship [9].

To jointly capitalize on network structure and the information conveyed by high-dimensional data points, we introduce networked exponential families. Networked exponential families couple network structure with (node-wise) local parameters of an exponential family [50]. Conceptually, they unify and considerably extend non-parametric models for learning clustered or smooth graph signals and, more generally, networked (generalized) linear models [11], [28], [32]. Another special case of networked exponential families are networked time series models. Networked time series models are useful for networks of weather observation stations (see Section VII-B).

The associate editor coordinating the review of this manuscript and approving it for publication was Zhaojie Ju.

Networked exponential families can also be applied to non-parametric density estimation [36]. Indeed, we might represent partitions of the feature space as a planar graph with nodes representing individual regions. The distribution of the data points within a particular region is estimated using an exponential family whose parameters are optimized for a specific region. Another important application of networked exponential families is to stratified models [40]. Indeed, stratified models are networked exponential families with each node in the network representing one stratum [48].

Networked exponential families are powerful statistical models for many important application domains such as personalized (high-precision) health-care [31], or natural language processing [4], [9]. In contrast to [9], which uses a probabilistic model for the network structure of text corpora, this article assumes the network structure as fixed and known.

While traditional clustering methods only use network structure, methods based on networked exponential families also use the information provided by node attributes. Joint clustering and optimization has been considered in [52] for probabilistic models of the network structure. We consider the network structure fixed and given and use a probabilistic model for the node attributes (features and labels).

A. EXISTING WORK

The closest to this work is [32] which considers regression with network cohesion (RNC). The RNC model is a special case of networked exponential families. While RNC uses a shared weight vector and a local (varying) intercept term, this article allows for arbitrarily varying weight vectors (see end of Sec. II).

Another main difference between [32] and our approach is the choice of regularizer for the networked model. While [32], similar to most existing work on semi-supervised learning [10], uses the graph Laplacian quadratic form as a smoothness measure, our approach controls the non-smooth total variation (TV) of the model parameters. TV-based regularization produces predictors which are piece-wise constant over well-connected subset of nodes. This behaviour is useful in image processing of natural images which are composed or homogenous segments whose boundaries result in sharp edges [17].

This article substantially extends our prior work on networked linear models for regression and classification [1], [28], [29], [47]. We have recently derived conditions on the data network structure such that nLasso accurately learns a clustered graph signal [29]. The clustered graph signal model is a special case of a networked linear regression model (see Section III-A)

Minimizing the Laplacian quadratic form amounts to solving a linear system. In contrast, TV minimization is intrinsically non-linear which requires more advanced techniques such as proximal methods [8], [41] (see Section VI). The higher computational cost of TV minimization affords improved accuracy when learning from a small number of observed data points (see [37] and Section VII-A).

We learn the parameters of networked exponential families with the network Lasso (nLasso). The network Lasso is a recently proposed extension of the Lasso to networked data [20], [22]. It is an instance of regularized empirical risk minimization, using total variation for regularization [18], [21]. We show how the nLasso can be implemented as highly scalable message passing protocol over the data network structure. This method is privacy-preserving in the sense of not sharing any raw data but only (estimates) of parameter vectors for the node-wise exponential families.

B. CONTRIBUTION

We now summarize our main contributions.

- We introduce networked exponential families as a novel modelling paradigm for high-dimensional data points having an intrinsic network structure (''big data over networks'').
- We present sufficient conditions on the network structure and observed data points that allow to accurately learn the parameters of an underlying networked exponential family with high probability.
- We solve the nLasso for learning the parameters of a networked exponential family using a highly scalable message passing algorithm. This algorithm is suitable for federated learning and edge computing environments, where computation is carried by a collection of low-complexity units (e.g., IoT devices) [30], [33].
- We verify our theoretical findings using illustrative numerical experiments. The source files for these experiments are made available to ensure reproducible research.

C. OUTLINE

We introduce networked exponential families in Section II. Section III details how some recently proposed models for networked data are obtained as special cases of networked exponential families. In Section IV, we show how to learn a networked exponential family using an instance of the nLasso optimization problem. We present an analysis of the nLasso estimation error in Section V. Section VI presents the implementation of nLasso using a primal-dual method for convex optimization. The computational and statistical properties of nLasso in networked exponential families are illustrated in numerical experiments within Section VII.

D. NOTATION

The Euclidean norm of a vector $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T \in \mathbb{R}^d$ is $\|\mathbf{x}\|_2 := \sqrt{\sum_{r=1}^{d} x_r^2}$. The spectral norm of a matrix is $\|\mathbf{M}\| := \sup_{\|\mathbf{x}\| \leq 1} \|\mathbf{M}\mathbf{x}\|$. The convex conjugate of a function f is $f^*(\mathbf{y}) := \sup_{\mathbf{x}} (\mathbf{y}^T \mathbf{x} - f(\mathbf{x}))$. The vector $\mathbf{e}^{(j)} \in \mathbb{R}^d$ denotes the jth column of the identity matrix $\mathbf{I}_d$ of size $d \times d$. Given a subset $\mathcal{A} \subseteq \mathcal{B}$ we denote the complement as $\overline{\mathcal{A}} := \mathcal{B} \setminus \mathcal{A}$.

II. NETWORKED EXPONENTIAL FAMILIES

We consider networked data represented by an undirected weighted graph (the ''empirical graph'') $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ (see Figure 1). The nodes $i \in \mathcal{V} = \{1, \ldots, N\}$ represent individual

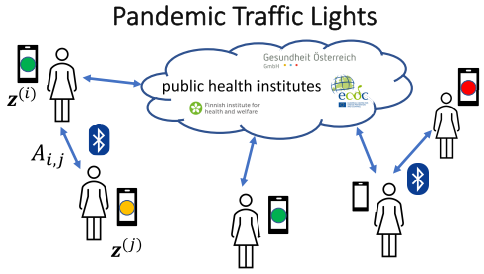

FIGURE 1. A networked exponential family is a probabilistic model for (high-dimensional) data points $\mathbf{z}^{(i)}$ related by some intrinsic network structure. The data points $\mathbf{z}^{(i)}$ might represent individuals during a pandemic which are related by contact-networks as well as bio-medical network structures.

data point (such as social network users). Data points $i, j \in \mathcal{V}$ are connected by an undirected edge $e = \{i, j\} \in \mathcal{E}$ with weight

$$A_e = A_{ij} > 0 \tag{1}$$

if they are considered similar (e.g., befriended users). We denote the edge set $\mathcal{E}$ by $\{1, \ldots, E := |\mathcal{E}|\}$. The neighbourhood of a node $i \in \mathcal{V}$ is $\mathcal{N}(i) := \{j : \{i, j\} \in \mathcal{E}\}$.

We assume the empirical graph $\mathcal{G}$ is fixed and known. The empirical graph might be obtained via physical proximity (in time or space), physical connection (communication networks) or statistical dependency (probabilistic graphical models) [15], [25], [50]. Section VIII briefly speculates on how our analysis could allow to use network design methods for data-driven learning of the empirical graph.

Beside network structure, datasets convey additional information via attributes $\mathbf{z}^{(i)} \in \mathbb{R}^d$ of data points $i \in \mathcal{V}$. We model the attributes $\mathbf{z}^{(i)}$ of data points $i \in \mathcal{V}$ as independent random variables distributed according to (a member of) some exponential family [50]

$$p(\mathbf{z}; \overline{\mathbf{w}}^{(i)}) := b^{(i)}(\mathbf{z}) \exp\left((\overline{\mathbf{w}}^{(i)})^T \mathbf{t}^{(i)}(\mathbf{z}) - \Phi^{(i)}(\overline{\mathbf{w}}^{(i)})\right). \tag{2}$$

The distribution (2) is parametrized by the weight vectors $\overline{\mathbf{w}}^{(i)} \in \mathcal{W}^{(i)}$, for $i \in \mathcal{V}$. The weight vectors are fixed but unknown and the main focus of this article is the accurate estimation (learning) of these weight vectors. In order to ensure (2) defines a valid probability measure (non-negative and total measure equal to one), we must restrict the weight vectors to some subset $\mathcal{W}^{(i)} \subseteq \mathbb{R}^d$ (see [50, Sec. 3.2]).

It is convenient to collect weight vectors $\mathbf{w}^{(i)}$ assigned to each n ode i into a vector-valued graph signal $\mathbf{w} : \mathcal{V} \to \mathbb{R}^d$ which maps a node i to the function value $\mathbf{w}^{(i)}$. The (hypothesis) space of all such vector-valued graph signals is

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

We also define the related space of vector-valued signals defined on the edges $\mathcal{E}$ of the empirical graph $\mathcal{G}$ as

$$\mathcal{D} := \{\mathbf{u} : \mathcal{E} \to \mathbb{R}^d : e \mapsto \mathbf{u}^{(e)}\}. \tag{4}$$

Strictly speaking, (2) represents a probability density function relative to some underlying base measure ν defined on the value range of the sufficient statistic $\mathbf{t}^{(i)}(\mathbf{z}^{(i)})$. Important examples of such a base measure are the counting measure for discrete-valued $\mathbf{t}^{(i)}$ or the Lesbegue measure for continuous-valued $\mathbf{t}^{(i)}$. The distribution defined by (2) depends on $\mathbf{z}^{(i)}$ only via the sufficient statistic $\mathbf{t}^{(i)}(\mathbf{z}^{(i)})$. In what follows, we suppress the argument and write $\mathbf{t}^{(i)}$ with the implicit understanding that it is a function of the random vector $\mathbf{z}^{(i)}$.

Several important properties of the model (2) can be read off the cumulant function $\Phi^{(i)}(\cdot) : \mathcal{W}^{(i)} \to \mathbb{R}$, [50]

$$\Phi^{(i)}(\mathbf{w}^{(i)}) := log \int_{\mathbf{t} \in \mathbb{R}^d} b(\mathbf{t}) \exp(-\mathbf{t}^T \mathbf{w}^{(i)}) \nu(d\mathbf{t}). \tag{5}$$

The domain $\mathcal{W}^{(i)} \subseteq \mathbb{R}^d$ of the cumulant function is given by all weight vectors $\mathbf{w}^{(i)}$ such that the integral in (5) exists and is finite. The Fisher information matrix (FIM) $\mathbf{F}^{(i)}$ for (2) is the Hessian

$$\mathbf{F}^{(i)} = \nabla^2 \Phi^{(i)}(\mathbf{w}^{(i)}), F^{(i)}_{m,n}(\mathbf{w}^{(i)}) := \frac{\partial^2 \Phi^{(i)}(\mathbf{w}^{(i)})}{\partial w^{(i)}_m w^{(i)}_n}. \tag{6}$$

The structure of $\mathbf{F}^{(i)}$ determines the statistical and computational properties of (2) (see Section V and VI).

The node-wise models (2), for all nodes $i \in \mathcal{V}$, are coupled by requiring the weight vectors $\overline{\mathbf{w}}^{(i)}$ to be similar for well-connected data points. In particular, we require the weight vectors to have a small total variation (TV)

$$\|\mathbf{w}\|_{TV} := \sum_{\{i,j\} \in \mathcal{E}} A_{ij} \|\mathbf{w}^{(j)} - \mathbf{w}^{(i)}\|. \tag{7}$$

Requiring a small TV of the weight vectors $\mathbf{w}^{(i)}$, for $i \in \mathcal{V}$, forces them to be approximately constant over well connected subsets (clusters) of nodes. It will be convenient to define the TV for a subset $\mathcal{S}$ of edges:

$$\|\mathbf{w}\|_{\mathcal{S}} := \sum_{\{i,j\} \in \mathcal{S}} A_{ij} \|\mathbf{w}^{(j)} - \mathbf{w}^{(i)}\|. \tag{8}$$

Networked exponential families are obtained as the combination of (2) with a constraint on the TV (8) of weights $\overline{\mathbf{w}}$ in (8).

Networked exponential families are somewhat similar to the RNC model put forward in [32]. Let us detail some key differences between those two modelling frameworks. First, RNC considers the special case of distributions (2) with the sufficient statistic $\mathbf{t}^{(i)} = \left((\mathbf{x}^{(i)})^T, 1\right)^T$ and a partitioned weight vector $\overline{\mathbf{w}} = \left(\boldsymbol{\beta}^T, \alpha^{(i)}\right)^T$. The component $\boldsymbol{\beta}$ is the same for all nodes $i \in \mathcal{V}$, while the intercept $\alpha^{(i)}$ is allowed to vary over nodes $\mathcal{V}$. In contrast, we allow the entire weight vector $\overline{\mathbf{w}}$ to vary between different nodes. Moreover, while the RNC model uses the smooth Laplacian quadratic form of the intercepts $\alpha^{(i)}$, we use the non-smooth TV (7) to ensure that the weight vectors conform with the network structure of the data.

III. SOME EXAMPLES

We now discuss important special cases of generic exponential families (2). These special cases are obtained for specific choices for the sufficient statistic $\mathbf{t}^{(i)}(\mathbf{z})$ and cumulant function $\Phi^{(i)}(\cdot) : \mathcal{W}^{(i)} \to \mathbb{R}$ (5).

A. NETWORKED LINEAR REGRESSION

Consider networked data points $i \in \mathcal{V}$ with features $\mathbf{x}^{(i)} \in \mathbb{R}^d$ and label $y^{(i)} \in \mathbb{R}$. Maybe the most basic (yet quite useful) model for the relation between features and labels of a data point is the linear model

$$y^{(i)} = (\mathbf{x}^{(i)})^T \mathbf{w}^{(i)} + \varepsilon^{(i)}, \tag{9}$$

with Gaussian noise $\varepsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$ of known variance σ_i^2 which can vary for different nodes $i \in \mathcal{V}$. The linear model (9) is parametrized by the weight vectors $\mathbf{w}^{(i)}$ for each $i \in \mathcal{V}$. The weight vectors are coupled by requiring a small TV (7) [28].

The model (9) is obtained from (2) using the choices $z^{(i)} := y^{(i)}$ with $\mathbf{t}^{(i)}(z) = (z/\sigma_i^2)\mathbf{x}^{(i)}$ and $\Phi^{(i)}(\mathbf{w}) = (\mathbf{w}^T\mathbf{x}^{(i)})^2/(2\sigma_i^2)$.

In some applications we might have only a crude estimate for the label of some data points. We can cope with varying levels of accuracy in observed labels by using a varying noise variance σ_i^2 in (9). For nodes $i \in \mathcal{V}$ for which we only have a rough label estimate, we use a larger noise variance σ_i^2 in (9).

B. NETWORKED LOGISTIC REGRESSION

Consider networked data points $i \in \mathcal{V}$ with features $\mathbf{x}^{(i)} \in \mathbb{R}^d$ and labels $y^{(i)} \in \{-1, 1\}$. Within logistic regression, we interpret the label y as the realization of a random variable with probability distribution

$$p(y^{(i)} = 1; \mathbf{w}^{(i)}) := 1/(1 + \exp(-(\mathbf{w}^{(i)})^T\mathbf{x}^{(i)})). \tag{10}$$

The distribution (10) is parametrized by the weight vector $\mathbf{w}^{(i)}$ for each node $i \in \mathcal{V}$. Note that (10) is the posterior distribution of label $y^{(i)}$ given the features $\mathbf{x}^{(i)}$ if the feature vector $\mathbf{x}^{(i)}$ is a Gaussian random vector conditioned on $y^{(i)}$. Networked logistic regression requires the weight vectors $\mathbf{w}^{(i)}$ in (10) to have a small TV (7) [1], [47].

The logistic regression model (10) is the special case of (2) for the choices $z^{(i)} := y^{(i)}$, $\mathbf{t}^{(i)}(z) := \mathbf{x}^{(i)}z/2$ and

$$\Phi^{(i)}(\mathbf{w}^{(i)}) = \big(\mathbf{w}^{(i)}\big)^T\mathbf{x}^{(i)}/2 + \log\big(1 + \exp\big(-\big(\mathbf{w}^{(i)}\big)^T\mathbf{x}^{(i)}\big)\big).$$

C. NETWORKED LDA

Consider a networked dataset representing a collection of text documents (such as scientific articles). The LDA is a probabilistic model for the relative frequencies of words in a document [4], [50]. Within LDA, each document is considered a blend of different topics. Each topic has a characteristic distribution of the words in the vocabulary.

A simplified form of LDA represents each document $i \in \mathcal{V}$ containing N "words" by two sequences of multinomial random variables $z_{w,1}^{(i)}, \ldots, z_{w,N}^{(i)} \in \{1, \ldots, W\}$ and $z_{t,1}^{(i)}, \ldots, z_{t,N}^{(i)} \in \{1, \ldots, T\}$ with V being the size of the vocabulary defining elementary words and T is the number of different topics. It can be shown that LDA is a special case of the exponential family (2) with particular choices for $\mathbf{t}(\cdot)$ and $\Phi^{(i)}(\cdot)$ (see [4], [50]).

IV. NETWORK LASSO

This article focuses on accurate learning of the true underlying weights $\overline{\mathbf{w}}^{(i)}$ (see (2)) based on the nodes attributes $\mathbf{z}^{(i)}$ for a small "training set" $\mathcal{M} = \{i_1, \ldots, i_M\} \subseteq \mathcal{V}$. A reasonable estimate for the weight vectors is obtained from maximizing the likelihood of observing the attributes $\mathbf{z}^{(i)}$,

$$\begin{aligned} p\big(\{\mathbf{z}^{(i)}\}_{i\in\mathcal{M}}\big) &= \prod_{i\in\mathcal{M}} p(\mathbf{z}^{(i)}; \mathbf{w}^{(i)}) \\ &\overset{(2)}{=} \prod_{i\in\mathcal{M}} b^{(i)}(\mathbf{z}^{(i)}) \exp\big(\big(\mathbf{t}^{(i)}\big)^T\mathbf{w}^{(i)} - \Phi^{(i)}(\mathbf{w}^{(i)})\big). \end{aligned} \tag{11}$$

Maximizing (11) is equivalent to minimizing

$$\widehat{E}(\mathbf{w}) := (1/M) \sum_{i\in\mathcal{M}} -\big(\mathbf{t}^{(i)}\big)^T\mathbf{w}^{(i)} + \Phi^{(i)}(\mathbf{w}^{(i)}). \tag{12}$$

The criterion (12) is not enough to learn the weights $\mathbf{w}^{(i)}$ for all $i \in \mathcal{V}$. Indeed, (12) ignores the weights of unobserved nodes $i \in \mathcal{V} \setminus \mathcal{M}$. We need to impose additional structure on the estimate for the weight vectors. In what follows, we require any reasonable estimate $\widehat{\mathbf{w}}^{(i)}$ to conform, in a specific sense, with the *cluster structure* of the empirical graph $\mathcal{G}$ [38].

Networked data is often organized as clusters (or communities) which are well-connected subset of nodes. Many supervised learning methods use a clustering assumption that nodes belonging to the same cluster represent similar data points. We implement this clustering assumption by requiring the parameter vectors $\mathbf{w}^{(i)}$ in (2) to have a small TV (7).

We are led to learning the weights $\widehat{\mathbf{w}}$ for (2) via the *regularized empirical risk minimization* (ERM)

$$\widehat{\mathbf{w}} \in \arg\min_{\mathbf{w}\in\mathcal{H}} \widehat{E}(\mathbf{w}) + \lambda\|\mathbf{w}\|_{TV}. \tag{13}$$

The learning problem (13) is an instance of the generic nLasso problem [20]. The parameter λ in (13) allows to trade-off small TV $\|\widehat{\mathbf{w}}\|_{TV}$ against small error $\widehat{E}(\widehat{\mathbf{w}})$ (cf. (12)). Choosing λ can be based on validation [22] or the error analysis in Section V.

It will be convenient to reformulate (13) using the block-incidence matrix $\mathbf{D} \in \mathbb{R}^{(dE)\times(dN)}$ as

$$\mathbf{D}_{e,i} = \begin{cases} A_{ij}\mathbf{I}_d & e = \{i,j\}, i < j \\ -A_{ij}\mathbf{I}_d & e = \{i,j\}, i > j \\ \mathbf{0} & \textit{otherwise}. \end{cases} \tag{14}$$

The e-th block of $\mathbf{Dw}$ is $A_{ij}(\mathbf{w}^{(i)} - \mathbf{w}^{(j)})$ in (7) and, in turn,

$$\|\mathbf{w}\|_{TV} = \|\mathbf{Dw}\|_{2,1} \tag{15}$$

with the norm $\|\mathbf{u}\|_{2,1} := \sum_{e\in\mathcal{E}} \|\mathbf{u}^{(e)}\|_2$ defined on $\mathcal{D}$ (see (4)). We can then reformulate the nLasso (13) as

$$\widehat{\mathbf{w}} \in \arg\min_{\mathbf{w}\in\mathcal{H}} h(\mathbf{w}) + g(\mathbf{Dw}), \tag{16}$$

with $h(\mathbf{w}) = \widehat{E}(\mathbf{w})$ and $g(\mathbf{u}) := \lambda\|\mathbf{u}\|_{2,1}$.

Related to the incidence matrix (14), is the graph Laplacian

$$\mathbf{L} = \Lambda \otimes \mathbf{I}_d - \big(\mathbf{A} \circ \mathbf{A}\big) \otimes \mathbf{I}_d, \tag{17}$$

with the (element-wise) squared weight matrix $\big(\mathbf{A} \circ \mathbf{A}\big)_{i,j} = A_{i,j}^2$ (see (1)) and the "degree matrix"

$$\Lambda = \mathrm{diag}\{\beta_1, \ldots, \beta_N\} \in \mathbb{R}^{N \times N}, \quad \text{with } \beta_i := \sum_{\{j,i\} \in \mathcal{E}} A_{i,j}^2.$$

The (sorted) eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots$ of $\mathbf{L}$ reflect the connectivity of the graph $\mathcal{G}$. A graph $\mathcal{G}$ is connected if and only if $\lambda_2 > 0$. Moreover, the spectral gap $\rho(\mathcal{G}) := \lambda_2$ provides a measure of the connectivity of the graph $\mathcal{G}$.

The Laplacian matrix $\mathbf{L}$ is closely related to the incidence matrix $\mathbf{D}$ (see (14)). Both matrices have the same nullspace. Moreover, the spectrum of $\mathbf{D}\mathbf{D}^T$ coincides with the spectrum of $\mathbf{L}$. The column blocks $\mathbf{S}^{(j)} \in \mathbb{R}^{(Nd) \times d}$ of the pseudo-inverse $\mathbf{D}^\dagger = \big(\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(|\mathcal{E}|)}\big) \in \mathbb{R}^{(Nd) \times (|\mathcal{E}|d)}$ of $\mathbf{D}$ satisfy

$$\|\mathbf{S}^{(j)}\|_{2,\infty} \leq \sqrt{2d \max_{i,j} A_{i,j}}/\rho(\mathcal{G}). \tag{18}$$

This bound can be verified using the identity $\mathbf{D}^\dagger = (\mathbf{D}\mathbf{D}^T)^\dagger \mathbf{D}^T$ and well-known vector norm inequalities (see, e.g., [23]).

V. STATISTICAL ASPECTS

We now turn to the characterization of statistical properties of nLasso by analysing the prediction error $\widetilde{\mathbf{w}} = \widehat{\mathbf{w}} - \overline{\mathbf{w}}$ incurred by a solution $\widehat{\mathbf{w}}$ of the nLasso problem (13). In order to analyze the error incurred by the nLasso (13), we assume that the true weight vectors are clustered

$$\overline{\mathbf{w}}^{(i)} = \sum_{\mathcal{C} \in \mathcal{P}} \mathbf{v}^{(\mathcal{C})} \mathcal{I}_{\mathcal{C}}[i]. \tag{19}$$

Here, $\mathbf{v}^{(\mathcal{C})} \in \mathbb{R}^d$ is the value of the true weigh vector for all nodes in the cluster $\mathcal{C}$. We also used the indicator map $\mathcal{I}_{\mathcal{C}}[i] = 1$ for $i \in \mathcal{C}$ and $\mathcal{I}_{\mathcal{C}}[i] = 0$ otherwise.

The model (19) involves a partitioning $\mathcal{P} = \{\mathcal{C}_1, \ldots, \mathcal{C}_{|\mathcal{P}|}\}$ of the nodes $\mathcal{V}$ into disjoint subsets ("cluster") $\mathcal{C}_l$. The model (19) is a special case of piece-wise polynomial signal model which allows the weight vectors to vary within each cluster [12].

In principle, our analysis applies to an arbitrary choice for the partition $\mathcal{P}$. However, the analysis is most useful if the partition is such that the boundary

$$\partial\mathcal{P} := \{\{i,j\} \in \mathcal{E} : i \in \mathcal{C}_l, j \in \mathcal{C}_{l'}, l \neq l'\} \tag{20}$$

is small in some sense. In particular, we focus on partitions such that $\sum_{e \in \partial\mathcal{P}} A_e$ is small.

We will use the model (19) is a ("zero-order") approximation for the true underlying weight vectors in (2). The analysis below indicates that nLasso methods are robust to model mismatch, i.e., the true underlying weight vectors in (2) can be well approximated by (19).

Assumption 1: Node attributes $\mathbf{z}^{(i)}$ are distributed according to (2) *with weight vectors $\overline{\mathbf{w}}^{(i)}$ that are piece-wise constant over some partition $\mathcal{P} = \{\mathcal{C}_1, \ldots, \mathcal{C}_{|\mathcal{P}|}\}$ (see* (19)*). We measure the clusteredness of the partition $\mathcal{P}$ using the spectral gap*

$$\rho_{\mathcal{P}} := \min_{\mathcal{C}_l \in \mathcal{P}} \rho(\mathcal{C}_l). \tag{21}$$

We emphasize that the partition underlying the model (19) is only required for the analysis of the nLasso error. For the implementation of nLasso (see Section VI), we do not need any information about the partition $\mathcal{P}$.

The next assumption requires the node-wise exponential families to be well conditioned. In particular, we require the eigenvalues of the FIMs $\mathbf{F}^{(i)}$ (see (6) to be upper and lower bounded with known constants.

Assumption 2: The FIM $\mathbf{F}^{(i)}$ (see (6)*) satisfies*

$$U\mathbf{I} \succeq \mathbf{F}^{(i)}(\mathbf{w}^{(i)}) \succeq L\mathbf{I} \quad \text{for any } \mathbf{w}^{(i)} \in \mathcal{W}^{(i)}, \tag{22}$$

with some constants $U \geq L > 1$.

Our third and final assumption involves the training set $\mathcal{M}$ and requires the cluster boundaries to be well connected to nodes in the training set. Conceptually, this assumption is similar to inconvertibility conditions such as the restricted isometry property or the compatibility condition in compressed sensing [7], [16]. This assumption ensures that no clustered weight vectors (see (19)) can be small on the entire training set $\mathcal{M}$. We measure the size of the weight vectors on the training set via the norm

$$\|\mathbf{w}\|_{\mathcal{M}} := \sqrt{(1/M) \sum_{i \in \mathcal{M}} \big\|\mathbf{w}^{(i)}\big\|^2}.$$

Assumption 3: There is $K > 0$ and $L > 1$ such that for any piece-wise constant $\mathbf{z} \in \mathcal{H}$ (see (3) *and* (7)*),*

$$L\|\mathbf{z}\|_{\partial\mathcal{P}} \leq K\|\mathbf{z}\|_{\mathcal{M}} + \|\mathbf{z}\|_{\overline{\partial\mathcal{P}}}. \tag{23}$$

Note that both, Assumption 2 and 3, use the same constant L in the lower bounds (22) and (23), respectively. Our main analytic result is an upper bound on the probability that the nLasso error exceeds a given threshold η.

Theorem 1: Consider networked data $\mathcal{G}$ and training set $\mathcal{M}$ such that Assumption 1, 2 and 3 are satisfied with condition number (see 23) $\kappa := \frac{K+3}{L-3} < 1$. We estimate the weight vectors $\overline{\mathbf{w}}$ using a solution $\widehat{\mathbf{w}}$ of nLasso (13) *with $\lambda := \eta/(5\kappa^2)$ using some pre-specified error level $\eta > 0$. Then,*

$$\begin{aligned} P\{\|\widehat{\mathbf{w}} - \overline{\mathbf{w}}\|_{TV} \geq \eta\} \\ \leq 2|\mathcal{P}| \max_{l=1,\ldots,|\mathcal{P}|} \exp\left(-\frac{|\mathcal{C}_l|\eta^2}{8 \cdot 25dU\kappa^2}\right) \\ + 2|\mathcal{E}| \exp\left(-\frac{M\rho_{\mathcal{P}}^2\eta^2}{64 \cdot 25Ud\|\mathbf{A}\|_\infty^2\kappa^4}\right). \end{aligned} \tag{24}$$

The bound (24) becomes useful, tending towards zero, for sufficiently large clusters $\mathcal{C}_l$ and sufficiently large training set $\mathcal{M}$. In particular, the bound is useful for massive datasets represented by some large empirical graph which is composed of a modest number of clusters or segments. One application

involving large empirical graphs and a rather small number of clusters is image segmentation (see [46] and Section VII-C).

The bound (24) indicates that, for a prescribed accuracy level η, the training set size M has to scale according to $\kappa^4/\rho_{\mathcal{P}}^2$. Thus, the sample size required by Algorithm 1 scales with the fourth power of the condition number $\kappa = \frac{K+3}{L-3}$ (see Assumption 3) and inversely with the spectral gap $\rho_{\mathcal{P}}$ of the partitioning $\mathcal{P}$.

Thus, nLasso methods (13) (such as Algorithm 1) require less training data if the condition number κ is small and the spectral gap $\rho_{\mathcal{P}}$ is large. This is reasonable, since having a small condition number $\kappa = \frac{K+3}{L-3}$ (see Assumption 3) typically requires the edges within clusters to have larger weights on average than the weights of the boundary edges.

It also makes sense that nLasso is more accurate for a larger spectral gap $\rho_{\mathcal{P}}$. Indeed, a large spectral gap $\rho_{\mathcal{P}}$ indicates that the nodes within each cluster $\mathcal{C}_l$ are well connected. A graph $\mathcal{G}$ consisting of well-connected clusters $\mathcal{C}_l$ favours clustered graph signals (see (19)) as solutions of nLasso (13).

VI. COMPUTATIONAL ASPECTS

The objective function in (16) is highly structured as a sum of a smooth convex function $h(\mathbf{w})$ and a non-smooth convex function $g(\mathbf{D}\mathbf{w})$. Both of these two components can be optimized efficiently when considered separately. This suggests to use proximal methods to solve (16) [41].

A recently popularized instance of proximal methods is the alternating direction method of multipliers (ADMM) [5], [20]. However, we will choose another type of proximal method which is based on a dual problem to (16) [8], [42]. This primal-dual method is appealing since its analysis provides natural choices for the algorithm parameters. In contrast, tuning the ADMM parameter is non-trivial [39].

A. PRIMAL-DUAL METHOD

To develop an efficient method for solving (16), we start with reformulating the problem (16) as a saddle-point problem

$$\min_{\mathbf{w}\in\mathbb{R}^{dN}} \max_{\mathbf{u}\in\mathcal{D}} \mathbf{u}^T\mathbf{D}\mathbf{w} + h(\mathbf{w}) - g^*(\mathbf{u}), \tag{25}$$

with the convex conjugate g^* of g [8].

Any solution $(\widehat{\mathbf{w}}, \widehat{\mathbf{u}})$ of (25) is characterized by [43]

$$-\mathbf{D}^T\widehat{\mathbf{u}} \in \partial h(\widehat{\mathbf{w}}), \quad \text{and } \mathbf{D}\widehat{\mathbf{w}} \in \partial g^*(\widehat{\mathbf{u}}). \tag{26}$$

This condition is, in turn, equivalent to

$$\begin{aligned} \widehat{\mathbf{w}} - \mathbf{T}\mathbf{D}^T\widehat{\mathbf{u}} &\in (\mathbf{I}_{dN} + \mathbf{T}\partial h)(\widehat{\mathbf{w}}), \quad \text{and} \\ \widehat{\mathbf{u}} + \mathbf{\Sigma}\mathbf{D}\widehat{\mathbf{w}} &\in (\mathbf{I}_{dE} + \mathbf{\Sigma}\partial g^*)(\widehat{\mathbf{u}}), \end{aligned} \tag{27}$$

with positive definite matrices $\mathbf{\Sigma} \in \mathbb{R}^{dE\times dE}$, $\mathbf{T} \in \mathbb{R}^{dN\times dN}$. The matrices $\mathbf{\Sigma}, \mathbf{T}$ are design parameters whose choice will be detailed below. The condition (27) lends naturally to the following coupled fixed point iterations [42]

$$\mathbf{w}_{k+1} = (\mathbf{I} + \mathbf{T}\partial h)^{-1}(\mathbf{w}_k - \mathbf{T}\mathbf{D}^T\mathbf{u}_k) \tag{28}$$

$$\mathbf{u}_{k+1} = (\mathbf{I} + \mathbf{\Sigma}\partial g^*)^{-1}(\mathbf{u}_k + \mathbf{\Sigma}\mathbf{D}(2\mathbf{w}_{k+1} - \mathbf{w}_k)). \tag{29}$$

If the matrices $\mathbf{\Sigma}$ and $\mathbf{T}$ in (28), (29) satisfy

$$\|\mathbf{\Sigma}^{1/2}\mathbf{D}\mathbf{T}^{1/2}\|^2 < 1, \tag{30}$$

the sequence $\mathbf{w}_{k+1}$ (see (28), (29)) converges to a solution of (13) [42, Thm. 1]. The condition (30) is satisfied for

$$\mathbf{\Sigma} := diag\{(1/(2A_e))\mathbf{I}\}_{e\in\mathcal{E}}, \quad \mathbf{T} := diag\{(\tau/d^{(i)})\mathbf{I}\}_{i\in\mathcal{V}}, \tag{31}$$

with $d^{(i)} = \sum_{j\neq i} A_{ij}$ and some $\tau < 1$ [42, Lemma 2].

The update (29) involves the resolvent operator

$$(\mathbf{I} + \mathbf{\Sigma}\partial g^*)^{-1}(\mathbf{v}) = \arg\min_{\mathbf{v}'\in\mathcal{D}} g^*(\mathbf{v}') + (1/2)\|\mathbf{v}' - \mathbf{v}\|^2_{\mathbf{\Sigma}^{-1}}, \tag{32}$$

where $\|\mathbf{v}\|_{\mathbf{\Sigma}} := \sqrt{\mathbf{v}^T\mathbf{\Sigma}\mathbf{v}}$. The convex conjugate g^* of g (see (16)) can be decomposed as $g^*(\mathbf{v}) = \sum_{e=1}^{E} g_2^*(\mathbf{v}^{(e)})$ with the convex conjugate g_2^* of the scaled ℓ_2-norm $\lambda\|.\|$. Moreover, since $\mathbf{\Sigma}$ is a block diagonal matrix, the e-th block of the resolvent operator $(\mathbf{I}_{dE} + \mathbf{\Sigma}\partial g^*)^{-1}(\mathbf{v})$ can be obtained by the Moreau decomposition as [41, Sec. 6.5]

$$\begin{aligned} &((\mathbf{I}_{dE} + \mathbf{\Sigma}\partial g^*)^{-1}(\mathbf{v}))^{(e)} \\ &\overset{(32)}{=} \arg\min_{\mathbf{v}'\in\mathbb{R}^d} g_2^*(\mathbf{v}') + (1/(2\sigma^{(e)}))\|\mathbf{v}' - \mathbf{v}^{(e)}\|^2 \\ &= \mathbf{v}^{(e)} - \sigma^{(e)}(\mathbf{I}_d + (\lambda/\sigma^{(e)})\partial\|.\|)^{-1}(\mathbf{v}^{(e)}/\sigma^{(e)}) \\ &= \begin{cases} \lambda\mathbf{v}^{(e)}/\|\mathbf{v}^{(e)}\| & \text{if } \|\mathbf{v}^{(e)}\| > \lambda \\ \mathbf{v}^{(e)} & otherwise, \end{cases} \end{aligned}$$

where $(a)_+ = \max\{a, 0\}$ for $a \in \mathbb{R}$.

The update (28) involves the resolvent operator $(\mathbf{I}+\mathbf{T}\partial h)^{-1}$ of h (see (12) and (16)), which does not admit a simple closed-form solution in general. Using (31), the update (28) decomposes into independent node-wise updates

$$\mathbf{w}_{k+1}^{(i)} := \begin{cases} \arg\min_{\mathbf{w}\in\mathbb{R}^d} g^{(i)}(\mathbf{w}) & \text{for } i \in \mathcal{M} \\ \overline{\mathbf{w}}^{(i)} & \text{for } i \in \mathcal{V}\setminus\mathcal{M} \end{cases} \tag{33}$$

with $g^{(i)}(\mathbf{w}) := -\mathbf{w}^T\mathbf{t}^{(i)} + \Phi^{(i)}(\mathbf{w}) + \tilde{\tau}^{(i)}\|\mathbf{w} - \overline{\mathbf{w}}^{(i)}\|^2$, $\tilde{\tau}^{(i)} := M/(2\tau^{(i)})$ and

$$\overline{\mathbf{w}} := \mathbf{w}_k - \mathbf{T}\mathbf{D}^T\mathbf{u}_k. \tag{34}$$

The update (33) is a regularized maximum likelihood estimator for exponential families [50, Eq. 3.38]. The varying regularization term $\tilde{\tau}^{(i)}\|\mathbf{w} - \overline{\mathbf{w}}^{(i)}\|^2$ enforces $\mathbf{w}_{k+1}^{(i)}$ to be close to $\overline{\mathbf{w}}^{(i)}$. The vector $\overline{\mathbf{w}}^{(i)}$ is a corrected version of the previous iterate $\mathbf{w}_k^{(i)}$ (see (34)).

In general, there is no closed-form solution for the update (33). However, the update (33) is a smooth convex optimization problem that can be solved efficiently using iterative methods such as L-BGFS [34]. We detail a computationally cheap iterative method for approximately solving (33) in Sec. VI-C.

Let us denote the approximate solution to (33) by $\widehat{\mathbf{w}}_{k+1}^{(i)}$ and assume that it is sufficiently accurate such that

$$e_k = \|\widehat{\mathbf{w}}_{k+1}^{(i)} - \mathbf{w}_{k+1}^{(i)}\| \leq 1/k^2. \tag{35}$$

We require the approximation quality (for approximating the update (33)) to increase with the iteration number k. According to [13, Thm. 3.2], the error bound (35) ensures the sequences obtained by (28) and (29) when replacing the exact update (33) with the approximation $\widehat{\mathbf{w}}_{k+1}$ still converge to a saddle-point of (25) and, in turn, a solution of the nLasso problem (16).

Algorithm 1 Primal-Dual nLasso

Input: $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathbf{A})$, $\{\mathbf{z}^{(i)}\}_{i\in\mathcal{M}}$, $\mathcal{M}$, λ, $\mathbf{D}$
Init: set $\mathbf{\Sigma}$, $\mathbf{T}$ via (31), $k:=0$, $\widehat{\mathbf{w}}_0:=0$, $\widehat{\mathbf{u}}_0:=0$
1: **repeat**
2: $\quad\widehat{\mathbf{w}}_{k+1}:=\widehat{\mathbf{w}}_k-\mathbf{T}\mathbf{D}^T\widehat{\mathbf{u}}_k$
3: $\quad$**for** each observed node $i\in\mathcal{M}$ **do**
4: $\quad\quad$compute $\widehat{\mathbf{w}}_{k+1}^{(i)}$ by (approximately) solving (33)
5: $\quad$**end for**
6: $\quad\overline{\mathbf{u}}:=\mathbf{u}_k+\mathbf{\Sigma}\mathbf{D}(2\widehat{\mathbf{w}}_{k+1}-\widehat{\mathbf{w}}_k)$
7: $\quad\widehat{\mathbf{u}}_{k+1}^{(e)}=\overline{\mathbf{u}}^{(e)}-\left(1-\frac{\lambda}{\|\overline{\mathbf{u}}^{(e)}\|}\right)_+\overline{\mathbf{u}}^{(e)}$ for $e\in\mathcal{E}$
8: $\quad k:=k+1$
9: **until** stopping criterion is satisfied
Output: $(\widehat{\mathbf{w}}_k,\widehat{\mathbf{u}}_k)$.

Note that Algorithm 1 requires as input only the empirical graph along with the observed node attributes $\mathbf{z}^{(i)}$, for $i\in\mathcal{M}$. Algorithm 1 does not require any specification of a partition of the empirical graph. Moreover, in contrast to the ADMM implementation of nLasso (see [20, Alg. 1]), the proposed Algorithm 1 does not involve any additional tuning parameter for solving (16).

B. COMPUTATIONAL COMPLEXITY

Algorithm 1 can be implemented as message passing over the empirical graph $\mathcal{G}$ (see [1]). During each iteration, messages are passed over each edge $\{i,j\}\in\mathcal{E}$ in the empirical graph. The computation of a single message requires a constant amount of computation. The precise amount of computation, measured by the number of additions and multiplications, required for a single message depends on the particular instance of the update (33).

For a fixed number of iterations used for Algorithm 1, its computational complexity scales linearly with the number of edges $\mathcal{E}$. For bounded degree graphs, such as grid or chain graphs, this implies a linear scaling of complexity with number of data points.

However, the overall complexity for Algorithm 1 depends crucially on the number of iterations required to achieve accurate learning. A worst-case analysis shows that, even when computing the exact updates (33), the number of iterations scales inversely with the required estimation accuracy [8]. This convergence speed is optimal for chain graphs [26].

C. APPROXIMATE PRIMAL UPDATE

We discuss a simple iterative method for approximately solving the primal update (33). A solution $\mathbf{w}_{k+1}^{(i)}$ of (33) is characterized by the zero gradient condition [6]

$$\nabla f\big(\mathbf{w}_{k+1}^{(i)}\big)=\mathbf{0} \tag{36}$$

with $f(\mathbf{w}):=-\mathbf{w}^T\mathbf{z}^{(i)}+\Phi^{(i)}(\mathbf{w})+\tilde{\tau}^{(i)}\|\mathbf{w}-\overline{\mathbf{w}}^{(i)}\|^2$. Applying basic calculus to (36),

$$\mathbf{w}^{(i)}=\overline{\mathbf{w}}^{(i)}+(\tau^{(i)}/M)\big(\mathbf{z}^{(i)}-\nabla\Phi^{(i)}(\mathbf{w}^{(i)})\big). \tag{37}$$

The necessary and sufficient condition (37) (for $\mathbf{w}^{(i)}$ to solve (33)) is a fixed point equation $\mathbf{w}^{(i)}=\mathcal{T}(\mathbf{w}^{(i)})$ with

$$\mathcal{T}:\mathbb{R}^d\to\mathbb{R}^d:\mathbf{w}\mapsto\overline{\mathbf{w}}^{(i)}+(\tau^{(i)}/M)\big(\mathbf{z}^{(i)}-\nabla\Phi^{(i)}(\mathbf{w})\big). \tag{38}$$

By the mean-value theorem [45, Thm. 9.19.], the map $\mathcal{T}$ is Lipschitz with constant $(\tau^{(i)}/M)\|\mathbf{F}(\mathbf{w})\|$ where $\mathbf{F}^{(i)}$ is the FIM (6). Thus, if we choose $\tau^{(i)}$ such that

$$R:=(\tau^{(i)}/M)\|\mathbf{F}(\mathbf{w})\|<1, \tag{39}$$

the map $\mathcal{T}$ in (38) is a contraction and the fixed-point iteration

$$\widetilde{\mathbf{w}}^{(r+1)}=\mathcal{T}\widetilde{\mathbf{w}}^{(r)}\overset{(38)}{=}\overline{\mathbf{w}}^{(i)}+(\tau^{(i)}/M)\big(\mathbf{z}^{(i)}-\nabla\Phi^{(i)}(\widetilde{\mathbf{w}}^{(r)})\big) \tag{40}$$

will converge to a solution of (33).

Moreover, if (39) is satisfied, we can bound the deviation between the iterate $\mathbf{w}^{(r)}$ and the (unique) solution $\mathbf{w}_{k+1}^{(i)}$ of (33) as (see [45, Proof of Thm. 9.23])

$$\|\widetilde{\mathbf{w}}^{(r)}-\mathbf{w}_{k+1}^{(i)}\|\leq(R^r/(1-R))\|\widetilde{\mathbf{w}}^{(1)}-\widetilde{\mathbf{w}}^{(0)}\|. \tag{41}$$

Thus, if we use the approximation $\widehat{\mathbf{w}}_{k+1}^{(i)}:=\widetilde{\mathbf{w}}^{(r)}$ for the update (33), we can ensure (35) by iterating (40) for at least

$$r\geq\log\big[(1-R)\|\widetilde{\mathbf{w}}^{(1)}-\widetilde{\mathbf{w}}^{(0)}\|/k^2\big]/\log R. \tag{42}$$

Computing the update (40) requires the evaluation of the gradient $\nabla\Phi^{(i)}(\widetilde{\mathbf{w}}^{(r)})$ of the cumulant function $\Phi^{(i)}(\mathbf{w})$. According to [50, Prop. 3.1.],

$$\nabla\Phi^{(i)}(\mathbf{w})=E\{\mathbf{t}(\mathbf{z}^{(i)})\}\quad\text{with }\mathbf{z}^{(i)}\sim p(\mathbf{z};\mathbf{w}). \tag{43}$$

In general, the expectations (43) cannot be computed exactly in closed-form. A notable exception are exponential families $p(\mathbf{z};\mathbf{w})$ obtained from a probabilistic graphical model defined on a triangulated graph such as a tree. In this case it is possible to compute (43) in closed-form (see [50, Sec. 2.5.2]). Another special case of (2) for which (43) can be evaluated in closed-form is linear and logistic regression (see Sec. III).

D. PARTIALLY OBSERVED MODELS

The learning Algorithm 1 can be adapted easily to cope with partially observed exponential families [50]. In particular, for the networked LDA described in Sec. III, we typically have access only to the word variables $z_{w,1}^{(i)},\ldots,z_{w,N}^{(i)}$ of some documents $i\in\mathcal{M}\subseteq\mathcal{V}$. However, for (approximately) computing the update step (33) we would also need the values

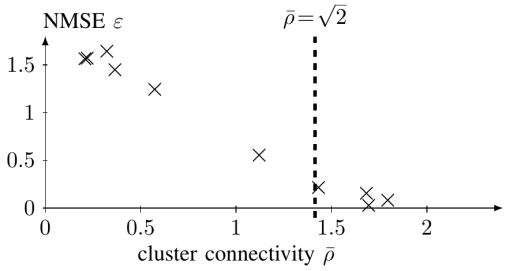

FIGURE 2. nLasso error for networked linear regression.

of the topic variables $z_{t,1}^{(i)},\ldots,z_{t,N}^{(i)}$ but those are not observed since they are latent (hidden) variables. In this case we can approximate (33) by some "Expectation-Maximization" (EM) principle (see [50, Sec. 6.2]). An alternative to EM methods, based on the method of moments, for learning (latent variable) topic models has been studied in a recent line of work [2].

VII. NUMERICAL EXPERIMENTS

We report on the numerical results obtained by applying particular instances of Algorithm 1 to different datasets. The source code to reproduce these experiments can be found at https://github.com/alexjungaalto/nLassoExpFamPDSimulations.

A. TWO-CLUSTER DATASET

This experiment constructs an empirical graph $\mathcal{G}$ by sparsely connecting two random graphs $\mathcal{C}_1$ and $\mathcal{C}_2$, each of size $N/2=40$ and with average degree 10. The nodes of $\mathcal{G}$ 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 (9) with zero noise $\varepsilon^{(i)}=0$ and piecewise constant weight vectors $\mathbf{w}^{(i)}=\mathbf{a}$ for $i\in\mathcal{C}_1$ and $\mathbf{w}^{(i)}=\mathbf{b}$ for $i\in\mathcal{C}_2$ with some two (different) fixed vectors $\mathbf{a},\mathbf{b}\in\mathbb{R}^2$. We assume that the labels $y^{(i)}$ are known for the nodes in a small training set $\mathcal{M}$ which includes three data points from each cluster, $|\mathcal{M}\cap\mathcal{C}_1|=|\mathcal{M}\cap\mathcal{C}_2|=3$.

As shown in [27], the validity of (23) in Assumption 3, depends on the connectivity of the cluster nodes with the boundary edges $\partial:=\{\{i,j\}\in\mathcal{E}:i\in\mathcal{C}_1,j\in\mathcal{C}_2\}$ which connect nodes in different clusters. To quantify the connectivity of the observed nodes $\mathcal{M}$ with the cluster boundary, we compute, for each cluster $\mathcal{C}_l$, the normalized flow value $\rho^{(l)}$ from one particular in each cluster $\mathcal{C}_l$ and the cluster boundary ∂. We normalize this flow by the boundary size $|\partial|$.

Fig. 2 depicts 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 $\rho^{(2)}$ (having same distribution). According to Fig. 2 there are two regimes of levels of connectivity. For connectivity $\bar{\rho}>\sqrt{2}$, Algorithm 1 is able to learn piece-wise constant weights $\mathbf{w}^{(i)}$.

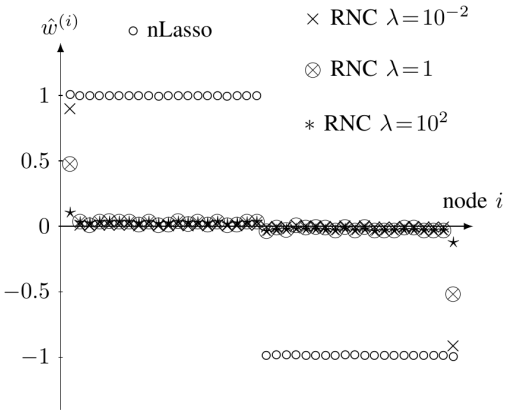

FIGURE 3. Weights learnt by Algorithm 1 and RNC [32].

To compare the effect of using TV (7) in Algorithm 1 instead of the graph Laplacian quadratic form (see [32]) as network regularizer, a networked signal in noise model $y^{(i)}=w^{(i)}+\varepsilon^{(i)}$ is considered. The noise $\varepsilon^{(i)}$ is i.i.d. with zero mean and known variance σ^2. The signal weights are piece-wise constant with $\bar{w}^{(i)}=1$ for $i\in\mathcal{C}_1$ and $\bar{w}^{(i)}=-1$ for $i\in\mathcal{C}_2$. The labels $y^{(i)}$ are observed for the nodes $\mathcal{M}=\{1,2,3,N-2,N-1,N\}$. Algorithm 1 is then used to learn weights $\hat{w}^{(i)}$ using a fixed number of 1000 iterations and $\lambda=10$. The RNC estimator reduces to to one matrix inversion (see [32, Eq. 2.4]) and is computed for the choices $\lambda\in\{1/100,1,100\}$ of the RNC regularization parameter. The resulting estimates $\hat{w}^{(i)}$ are shown in Fig. 3.

According to Figure 3, Algorithm 1 accurately learns the piece-wise constant weights $\bar{w}^{(i)}$ from only two labels $y^{(i)}$, for $i\in\{1,N\}$. In contrast, RNC fails to leverage the network structure in order to learn the weights from a small number of labels.

B. WEATHER DATA

In this experiment, we consider networked data obtained from the Finnish meteorological institute. The empirical graph $\mathcal{G}$ of this data represents Finnish weather stations. which are initially connected by an edge to their $K=3$ nearest neighbors. The feature vector $\mathbf{x}^{(i)}\in\mathbb{R}^3$ of node $i\in\mathcal{V}$ contains the local (daily mean) temperature for the preceding three days. The label $y^{(i)}\in\mathbb{R}$ is the current day-average temperature.

We apply Algorithm 1 to learn the weight vectors $\mathbf{w}^{(i)}$ for a localized linear model (9). For the sake of illustration we focus on the weather stations in the capital region around Helsinki. These stations are represented by nodes $\mathcal{C}=\{23,18,22,15,12,13,9,7,5\}$ and we assume that labels $y^{(i)}$ are available for all nodes outside $\mathcal{C}$ and for the nodes $i\in\{12,13,15\}\subseteq\mathcal{C}$. Thus, for more than half of the nodes in $\mathcal{C}$ we do not know the labels $y^{(i)}$ but predict them via $\hat{y}=\big(\widehat{\mathbf{w}}^{(i)}\big)^T\mathbf{x}^{(i)}$ with the weight vectors $\widehat{\mathbf{w}}^{(i)}$ obtained from Algorithm 1 (using $\lambda=1/7$ and a fixed number of 10^4 iterations). The normalized average squared prediction error

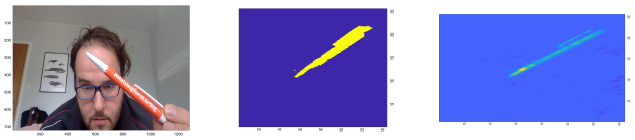

FIGURE 4. Left: Original image. Middle: Grabcut. Right: Algorithm 1.

is $\approx 10^{-1}$ and only slightly larger than the prediction error incurred by fitting a single linear model to the cluster $\mathcal{C}$.

C. IMAGE SEGMENTATION

We now discuss an experiment which show-cases Algorithm 1 for image segmentation [19], [44]. An image can be represented by an empirical graph whose nodes $i \in \mathcal{V}$ are image pixels at coordinates $\big(p^{(i)}, q^{(i)}\big) \in \{1,\ldots,P\} \times \{1,\ldots,Q\}$ (see Figure 4). Two nodes $i,j \in \mathcal{V}$ are connected by an edge $\{i,j\} \in \mathcal{E}$ if $p^{(i)} - p^{(j)} = 1$ or $q^{(i)} - q^{(j)} = 1$.

We assign all edges $\{i,j\} \in \mathcal{E}$ the same weight $W_{i,j} = 1$. Pixels $i \in \mathcal{V}$ are characterized by feature vectors $\mathbf{x}^{(i)}$ obtained by normalizing (zero mean and unit variance) the red, green and blue components of each pixel.

We then constructed a training set $\mathcal{M}$ of labeled data points by combining a background set $\mathcal{B} \subseteq \mathcal{V}$ ($y^{(i)} = 0$) and a foreground set $\mathcal{F} \subseteq \mathcal{V}$ ($y^{(i)} = 1$). These sets are determined based on the normalized redness $r^{(i)} := x_1^{(i)} / \max_{j \in \mathcal{V}} x_1^{(j)}$,

$$\mathcal{B} := \{i \in \mathcal{V} : r^{(i)} < 1/2\}, \quad \text{and } \mathcal{F} := \{i \in \mathcal{V} : r^{(i)} > 9/10\}. \tag{44}$$

We apply Algorithm 1, with $\lambda = 100$ and fixed number of 10 iterations, to learn the weights $\mathbf{w}^{(i)}$ for a networked logistic regression model (see Section III-B). For the update (33) in Algorithm 1 we used a single Newton step. The resulting predictions $\big(\widehat{\mathbf{w}}^{(i)}\big)^T \mathbf{x}^{(i)}$ are shown on the right of Figure 4. The middle of Figure 4 depicts the hard segmentation obtained by the "GrabCut" method [44]. Using MATLAB version 19 on a standard laptop, Algorithm 1 is almost ten times faster than GrabCut.

VIII. CONCLUSION

We have introduced networked exponential families as a flexible statistical modeling paradigm for networked data. The error of nLasso applied to learning networked exponential families has been analyzed. An efficient implementation of nLasso has been proposed using a primal-dual method for convex optimization. Directions for future research include a more detailed analysis of the convergence of nLasso for typical network structures as well as data-driven learning of the network structure (graphical model selection). The analysis of nLasso presented in Section V might guide the design of network structure by relating Assumption 3 to network flow problems (see [29]).

IX. PROOFS

We first collect some helper results in Sec. IX-A that will be used in Section IX-B to obtain a detailed derivation of Theorem 1.

A. HELPER RESULTS

Lemma 2: For any two vector signals $\mathbf{u}, \mathbf{v} \in \mathcal{H}$ (see (3)) defined on an empirical graph $\mathcal{G}$,

$$\sum_{i \in \mathcal{V}} \big(\mathbf{u}^{(i)}\big)^T \mathbf{v}^{(i)} \leq (1/|\mathcal{V}|)\Big(\sum_{i \in \mathcal{V}} \mathbf{v}^{(i)}\Big)^T \sum_{j \in \mathcal{V}} \mathbf{u}^{(j)} + \big\|\big(\mathbf{D}^\dagger\big)^T \mathbf{v}\big\|_{2,\infty} \|\mathbf{u}\|_{TV}. \tag{45}$$

Here, $\mathbf{D} \in \mathbb{R}^{(d|\mathcal{E}|) \times (d|\mathcal{V}|)}$ denotes the block-wise incidence matrix (14) of the empirical graph $\mathcal{G}$.

Proof: Any graph signal $\mathbf{u}$ can be decomposed as

$$\mathbf{u} = \mathbf{P}\mathbf{u} + (\mathbf{I} - \mathbf{P})\mathbf{u}, \tag{46}$$

with $\mathbf{P}$ denoting the orthogonal projection matrix on the nullspace of the block-wise graph Laplacian matrix $\mathbf{L}$ (17).

For a connected graph, the nullspace $\mathcal{K}(\mathbf{L})$ is spanned by d graph signals (see [49])

$$\mathbf{v}^{(j)} = \mathbf{1} \otimes \mathbf{e}^{(j)} \in \mathcal{H}, \quad \text{for } j \in \{1,\ldots,d\}. \tag{47}$$

Here, we used the constant graph signal $\mathbf{1} \in \mathbb{R}^{\mathcal{V}}$ assigning all nodes the same signal value 1. The projection matrix associated with the nullspace $\mathcal{K}(\mathbf{L})$ is

$$\mathbf{P} = \underbrace{(1/(\mathbf{1}^T\mathbf{1}))}_{=1/|\mathcal{V}|} \sum_{j=1}^{d} \mathbf{1}\big(\mathbf{1}\big)^T \otimes \mathbf{M}^{(j)}. \tag{48}$$

Here, $\mathbf{M}^{(j)} := \mathbf{e}^{(j)}\big(\mathbf{e}^{(j)}\big)^T$. Therefore,

$$\mathbf{P}\mathbf{u} \overset{(48)}{=} (1/|\mathcal{V}|) \sum_{j=1}^{d} \sum_{i \in \mathcal{V}} u_j^{(i)} \mathbf{1} \otimes \mathbf{e}^{(j)}. \tag{49}$$

The projection matrix on the orthogonal complement of $\mathcal{K}(\mathbf{L}) \subseteq \mathcal{H}$ is $\mathbf{I} - \mathbf{P}$. Then (see [24]),

$$\mathbf{I} - \mathbf{P} = \mathbf{D}^\dagger \mathbf{D}. \tag{50}$$

with the block-wise incidence matrix $\mathbf{D}$ (14). Combining (49) and (50) with (46),

$$\sum_{i \in \mathcal{V}} \big(\mathbf{u}^{(i)}\big)^T \mathbf{v}^{(i)} = (1/|\mathcal{V}|) \sum_{i,i' \in \mathcal{V}} \big(\mathbf{u}^{(i)}\big)^T \mathbf{v}^{(i')} + \mathbf{v}^T \mathbf{D}^\dagger \mathbf{D}\mathbf{u}. \tag{51}$$

To further develop (51), we define the norms

$$\|\mathbf{u}\|_{2,\infty} := \max_{e \in \mathcal{E}} \big\|\mathbf{u}^{(e)}\big\|_2, \text{ and } \|\mathbf{u}\|_{2,1} := \sum_{e \in \mathcal{E}} \|\mathbf{u}^{(e)}\|_2 \tag{52}$$

on the space $\mathcal{D}$ of vector-valued edge signals (see (4)). By the Cauchy-Schwarz inequality $\mathbf{a}^T\mathbf{b} \leq \|\mathbf{a}\|_2 \|\mathbf{b}\|_2$,

$$\mathbf{u}^T \mathbf{v} \leq \|\mathbf{u}\|_{2,\infty} \|\mathbf{v}\|_{2,1} \text{ for any } \mathbf{u}, \mathbf{v} \in \mathcal{D}. \tag{53}$$

Combining (51) with the inequality $\mathbf{a}^T\mathbf{b} \leq \|\mathbf{a}\|_2 \|\mathbf{b}\|_2$,

$$\sum_{i \in \mathcal{V}} \big(\mathbf{u}^{(i)}\big)^T \mathbf{v}^{(i)}$$

$$\leq (1/|\mathcal{V}|)\sum_{i,j\in\mathcal{V}} \big(\mathbf{u}^{(i)}\big)^{T}\mathbf{v}^{(j)} + \big\|\big(\mathbf{D}^{\dagger}\big)^{T}\mathbf{v}\big\|_{2,\infty}\|\mathbf{D}\mathbf{u}\|_{2,1}. \quad (54)$$

The result (45) follows from (54) by using (15). □

Applying Lemma 2 to the subgraphs induced by a partition $\mathcal{P}=\{\mathcal{C}_1,\ldots,\mathcal{C}_{|\mathcal{P}|}\}$, yields the following result.

Corollary 3: Consider an empirical graph $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathbf{A})$ and partition $\mathcal{P}=\{\mathcal{C}_1,\ldots,\mathcal{C}_{|\mathcal{P}|}\}$. Let $\mathcal{C}_l$ also denote the induced subgraph of a cluster and assume they are connected. For any two graph signals $\mathbf{u},\mathbf{v}\in\mathcal{H}$,

$$\sum_{i\in\mathcal{M}} \big(\mathbf{v}^{(i)}\big)^{T}\mathbf{u}^{(i)} \leq \max_{l=1,\ldots,|\mathcal{P}|}(1/|\mathcal{C}_l|)\big\|\sum_{i\in\mathcal{C}_l}\mathbf{v}^{(i)}\big\|_2 \sum_{j\in\mathcal{M}}\|\mathbf{u}^{(j)}\|_2 + \max_{l=1,\ldots,|\mathcal{P}|}\big\|\big(\mathbf{D}^{\dagger}_{\mathcal{C}_l}\big)^{T}\mathbf{v}_{\mathcal{C}_j}\big\|_{2,\infty}\|\mathbf{u}\|_{TV}. \quad (55)$$

Here, $\mathbf{D}_{\mathcal{C}_l}$ denotes the block-wise incidence matrix of the induced subgraph $\mathcal{C}_l$ (see (14)).

The proof of Theorem 1 (see Section IX-B) will require a large deviation bound for weighted sums of independent random vectors $\mathbf{z}^{(i)}$ distributed according to (2).

Lemma 4: Consider M independent random vectors $\mathbf{z}^{(i)}$, for $i\in\mathcal{M}$, distributed according to (2). For fixed unit-norm vectors $\|\mathbf{m}^{(i)}\|=1$, denote $y^{(i)}:=\big(\mathbf{m}^{(i)}\big)^{T}\mathbf{t}^{(i)}(\mathbf{z}^{(i)})$ and $\mu^{(i)}:=E\big\{y^{(i)}\big\}$. If $\nabla^2\Phi^{(i)}\preceq U\mathbf{I}$ for all $i\in\mathcal{M}$, then

$$P\Big\{\big|(1/M)\sum_{i\in\mathcal{M}}\big(y^{(i)}-\mu^{(i)}\big)\big|\geq\eta\Big\}\leq 2\exp\big(-M\eta^2/(2U)\big). \quad (56)$$

Proof: Set

$$y:=\sum_{i\in\mathcal{M}}y^{(i)}, \quad \text{and } \mu:=\sum_{i\in\mathcal{M}}\mu^{(i)}. \quad (57)$$

By Markov's inequality, for any $\theta>0$,

$$\begin{aligned} &P\Big\{(1/M)\sum_{i\in\mathcal{M}}\big(y^{(i)}-\mu^{(i)}\big)\geq\eta\Big\}\\ &=P\{y-\mu\geq M\eta\}\\ &=P\{\exp(\theta y)\geq\exp(\theta(M\eta+\mu))\}\\ &\leq\exp(-\theta(M\eta+\mu))E\{\exp(\theta y)\}\\ &=\exp(-\theta(M\eta+\mu))\prod_{i\in\mathcal{M}}E\{\exp(\theta y^{(i)})\}. \end{aligned} \quad (58)$$

The last equality in (58) is due to the independence of the random variables $y^{(i)}$.

Combining (58) with

$$E\{\exp(\theta y^{(i)})\}\overset{(5)}{=}\exp(\Phi^{(i)}(\overline{\mathbf{w}}^{(i)}+\theta\mathbf{m}^{(i)})-\Phi^{(i)}(\overline{\mathbf{w}}^{(i)})) \quad (59)$$

yields

$$P\{y-\mu\geq\eta\}\leq\exp(-\theta(M\eta+\mu)+\sum_{i\in\mathcal{M}}\Phi^{(i)}(\overline{\mathbf{w}}^{(i)}+\theta\mathbf{m}^{(i)})-\Phi^{(i)}(\overline{\mathbf{w}}^{(i)})). \quad (60)$$

Similarly,

$$P\{y-\mu\leq-\eta\}\leq\exp(-\theta(M\eta+\mu)+\sum_{i\in\mathcal{M}}\Phi^{(i)}(\overline{\mathbf{w}}^{(i)}+\theta\mathbf{m}^{(i)})-\Phi^{(i)}(\overline{\mathbf{w}}^{(i)})). \quad (61)$$

A union bound allows to sum up (60) and (62) to obtain

$$P\{|y-\mu|\geq\eta\}\leq 2\exp(-\theta(M\eta+\mu)+\sum_{i\in\mathcal{M}}\Phi^{(i)}(\overline{\mathbf{w}}^{(i)}+\theta\mathbf{m}^{(i)})-\Phi^{(i)}(\overline{\mathbf{w}}^{(i)})). \quad (62)$$

Using Taylor's theorem and $\nabla\Phi^{(i)}(\overline{\mathbf{w}}^{(i)})=E\{\mathbf{t}^{(i)}\}$ [50],

$$\begin{aligned}&\Phi^{(i)}(\overline{\mathbf{w}}^{(i)}+\theta\mathbf{m}^{(i)})-\Phi^{(i)}(\overline{\mathbf{w}}^{(i)})\\ &=\theta\mu^{(i)}+(\theta^2/2)\big(\mathbf{m}^{(i)}\big)^{T}\nabla^2\Phi^{(i)}(\overline{\mathbf{w}}^{(i)}\\ &\quad+\theta^{(i)}\mathbf{m}^{(i)})\mathbf{m}^{(i)}\end{aligned} \quad (63)$$

with some $\theta^{(i)}\in[0,\theta]$. Inserting $\nabla^2\Phi^{(i)}\preceq U\mathbf{I}$ into (63),

$$\Phi^{(i)}(\overline{\mathbf{w}}^{(i)}+\theta\mathbf{m}^{(i)})-\Phi^{(i)}(\overline{\mathbf{w}}^{(i)})\geq\theta\mu^{(i)}+\theta^2U/2,$$

and, in turn via (62),

$$P\{|y-\mu|\geq\eta\}\leq\exp(-\theta M\eta+M\theta^2U/2). \quad (64)$$

Optimizing (64) by choosing θ suitably yields (56). ■

Applying Lemma 4 using $\mathbf{m}=\mathbf{e}^{(l)}$ and using $\|\mathbf{x}\|_2\leq\sqrt{d}\|\mathbf{x}\|_\infty$, for any $\mathbf{x}\in\mathbb{R}^d$ yields the following result.

Corollary 5: Consider independent $\mathbf{z}^{(i)}$, for $i\in\mathcal{M}$, distributed according to (2). If $\nabla^2\Phi^{(i)}\preceq U\mathbf{I}$ for all $i\in\mathcal{M}$, then

$$P\Big\{\big\|(1/M)\sum_{i\in\mathcal{M}}\big(\mathbf{z}^{(i)}-E\{\mathbf{z}^{(i)}\}\big)\big\|\geq\eta\Big\}\leq 2\exp\big(-M\eta^2/(2dU)\big). \quad (65)$$

B. PROOF OF THEOREM 1

The high-level outline of the proof is as follows: First, we verify that the nLasso error is approximately clustered in the sense that the TV of the nLasso error over the inter-cluster edges can be bounded via the TV of the nLasso error over the intra-cluster edges. Using this fact, we can invoke Assumption 3 to upper bound the size of the nLasso error.

Any solution $\widehat{\mathbf{w}}$ of the nLasso problem (13) satisfies

$$\sum_{i\in\mathcal{M}}\big[\Phi^{(i)}(\widehat{\mathbf{w}}^{(i)})-(\widehat{\mathbf{w}}^{(i)})^{T}\mathbf{t}^{(i)}\big]+M\lambda\|\widehat{\mathbf{w}}\|_{TV} \leq\sum_{i\in\mathcal{M}}\big[\Phi^{(i)}(\overline{\mathbf{w}}^{(i)})-(\overline{\mathbf{w}}^{(i)})^{T}\mathbf{t}^{(i)}\big]+M\lambda\|\overline{\mathbf{w}}\|_{TV}. \quad (66)$$

We can rewrite (66) as

$$\sum_{i\in\mathcal{M}}\big(\boldsymbol{\varepsilon}^{(i)}\big)^{T}\widehat{\mathbf{w}}^{(i)}-\big(\overline{\mathbf{t}}^{(i)}\big)^{T}\widehat{\mathbf{w}}^{(i)}+\Phi^{(i)}(\widehat{\mathbf{w}}^{(i)})+\lambda\|\widehat{\mathbf{w}}\|_{TV} \leq\sum_{i\in\mathcal{M}}\big(\boldsymbol{\varepsilon}^{(i)}\big)^{T}\overline{\mathbf{w}}^{(i)}-\big(\overline{\mathbf{t}}^{(i)}\big)^{T}\overline{\mathbf{w}}^{(i)}+\Phi^{(i)}(\overline{\mathbf{w}}^{(i)})+\lambda\|\overline{\mathbf{w}}\|_{TV} \quad (67)$$

with $\bar{\mathbf{t}}^{(i)} := E\{\mathbf{t}^{(i)}\}$ and "observation noise" $\boldsymbol{\varepsilon}^{(i)} := \bar{\mathbf{t}}^{(i)} - \mathbf{t}^{(i)}$. To further develop (67), we make use of

$$\arg\min_{\mathbf{w}\in\mathbb{R}^d} -\mathbf{w}^T\bar{\mathbf{t}}^{(i)} + \Phi^{(i)}(\mathbf{w}) = \overline{\mathbf{w}}^{(i)}, \tag{68}$$

with the true weight vector $\overline{\mathbf{w}}^{(i)}$ underlying (2). The identity (68) can be verified by the zero-gradient condition and evaluating the gradient of $\Phi^{(i)}(\mathbf{w})$ (see [50, Proposition 3.1.]). Combining (67) with (68),

$$\begin{aligned}
&\sum_{i\in\mathcal{M}} \big(\boldsymbol{\varepsilon}^{(i)}\big)^T\widehat{\mathbf{w}}^{(i)} - \big(\bar{\mathbf{t}}^{(i)}\big)^T\widehat{\mathbf{w}}^{(i)} + \Phi^{(i)}(\widehat{\mathbf{w}}^{(i)}) + \lambda\|\widehat{\mathbf{w}}\|_{TV} \\
&\overset{(67)}{\leq} \sum_{i\in\mathcal{M}} \big(\boldsymbol{\varepsilon}^{(i)}\big)^T\overline{\mathbf{w}}^{(i)} - \big(\bar{\mathbf{t}}^{(i)}\big)^T\overline{\mathbf{w}}^{(i)} + \Phi^{(i)}(\overline{\mathbf{w}}^{(i)}) + \lambda\|\overline{\mathbf{w}}\|_{TV} \\
&\overset{(68)}{\leq} \sum_{i\in\mathcal{M}} \big(\boldsymbol{\varepsilon}^{(i)}\big)^T\overline{\mathbf{w}}^{(i)} - \big(\bar{\mathbf{t}}^{(i)}\big)^T\widehat{\mathbf{w}}^{(i)} + \Phi^{(i)}(\widehat{\mathbf{w}}^{(i)}) + \lambda\|\overline{\mathbf{w}}\|_{TV},
\end{aligned}$$

and, in turn,

$$\sum_{i\in\mathcal{M}} \big(\boldsymbol{\varepsilon}^{(i)}\big)^T\widetilde{\mathbf{w}}^{(i)} + \lambda\|\widehat{\mathbf{w}}\|_{TV} \leq \lambda\|\overline{\mathbf{w}}\|_{TV} \tag{69}$$

with the nLasso (estimation) error $\widetilde{\mathbf{w}} := \widehat{\mathbf{w}} - \overline{\mathbf{w}}$.

Let us assume for the moment that the observation noise $\boldsymbol{\varepsilon}^{(i)}$ is sufficiently small such that

$$\big|(1/M)\sum_{i\in\mathcal{M}} \big(\boldsymbol{\varepsilon}^{(i)}\big)^T\widetilde{\mathbf{w}}^{(i)}\big| \leq \lambda\kappa\|\widetilde{\mathbf{w}}\|_{\mathcal{M}} + (\lambda/2)\|\widetilde{\mathbf{w}}\|_{TV} \tag{70}$$

for every $\widetilde{\mathbf{w}} \in \mathcal{H}$. Here, we used the condition number $\kappa = \frac{K+3}{L-3}$ as defined in Theorem 1.

Inserting (70) into (69),

$$\|\widehat{\mathbf{w}}\|_{TV} \leq (1/2)\|\widetilde{\mathbf{w}}\|_{TV} + \|\overline{\mathbf{w}}\|_{TV} + \kappa\|\widetilde{\mathbf{w}}\|_{\mathcal{M}}, \tag{71}$$

and, in turn, via the decomposition property $\|\mathbf{w}\|_{TV} = \|\mathbf{w}\|_{\partial\mathcal{P}} + \|\mathbf{w}\|_{\mathcal{E}\setminus\partial\mathcal{P}}$ (see (8)),

$$\begin{aligned}
&\|\widehat{\mathbf{w}}\|_{\mathcal{E}\setminus\partial\mathcal{P}} \\
&\leq (1/2)\|\widetilde{\mathbf{w}}\|_{TV} + \|\overline{\mathbf{w}}\|_{TV} - \|\widehat{\mathbf{w}}\|_{\partial\mathcal{P}} + \kappa\|\widetilde{\mathbf{w}}\|_{\mathcal{M}} \\
&\overset{(a)}{\leq} (1/2)\|\widetilde{\mathbf{w}}\|_{TV} + \|\overline{\mathbf{w}}\|_{\partial\mathcal{P}} - \|\widehat{\mathbf{w}}\|_{\partial\mathcal{P}} + \kappa\|\widetilde{\mathbf{w}}\|_{\mathcal{M}} \\
&\overset{(b)}{\leq} (1/2)\|\widetilde{\mathbf{w}}\|_{TV} + \|\overline{\mathbf{w}} - \widehat{\mathbf{w}}\|_{\partial\mathcal{P}} + \kappa\|\widetilde{\mathbf{w}}\|_{\mathcal{M}}.
\end{aligned} \tag{72}$$

Here, step (a) is valid since we assume the true underlying weight vectors $\overline{\mathbf{w}}^{(i)}$ to be clustered according to (19). Step (b) uses the triangle inequality for the semi-norm $\|\cdot\|_{\partial\mathcal{P}}$ (see (8)).

Since $\|\widehat{\mathbf{w}}\|_{\mathcal{E}\setminus\partial\mathcal{P}} = \|\widetilde{\mathbf{w}}\|_{\mathcal{E}\setminus\partial\mathcal{P}}$, we can rewrite (72) as

$$\begin{aligned}
(1/2)\|\widetilde{\mathbf{w}}\|_{\mathcal{E}\setminus\partial\mathcal{P}} &\leq (3/2)\|\widetilde{\mathbf{w}}\|_{\partial\mathcal{P}} + \kappa\|\widetilde{\mathbf{w}}\|_{\mathcal{M}} \\
&\overset{\kappa\leq 1}{<} (3/2)\|\widetilde{\mathbf{w}}\|_{\partial\mathcal{P}} + \|\widetilde{\mathbf{w}}\|_{\mathcal{M}}.
\end{aligned} \tag{73}$$

Thus, for sufficiently small observation noise (such that (70) is valid), the nLasso error $\widetilde{\mathbf{w}} = \widehat{\mathbf{w}} - \overline{\mathbf{w}}$ is approximately clustered according to (19).

So far, we verified the nLasso error $\widetilde{\mathbf{w}}$ to be clustered. For some edge $\{i,j\} \in \mathcal{E}$, the error difference $\widetilde{\mathbf{w}}^{(i)} - \widetilde{\mathbf{w}}^{(j)}$, with $i,j \in \mathcal{C}_l$ belonging to the same cluster within the partition $\mathcal{P}$ underlying (19), tends to be small.

The next step is to verify that the nLasso error $\widetilde{\mathbf{w}} = \widehat{\mathbf{w}} - \overline{\mathbf{w}}$ (see (13)) cannot be too large. Applying the triangle inequality for the TV semi-norm to (67),

$$\begin{aligned}
&\sum_{i\in\mathcal{M}} \big(\boldsymbol{\varepsilon}^{(i)}\big)^T\widetilde{\mathbf{w}}^{(i)} - \big(\bar{\mathbf{x}}^{(i)}\big)^T\widehat{\mathbf{w}}^{(i)} + \Phi^{(i)}(\widehat{\mathbf{w}}^{(i)}) \\
&\qquad\leq \sum_{i\in\mathcal{M}} -\big(\bar{\mathbf{x}}^{(i)}\big)^T\overline{\mathbf{w}}^{(i)} + \Phi^{(i)}(\overline{\mathbf{w}}^{(i)}) + M\lambda\|\widetilde{\mathbf{w}}\|_{TV}.
\end{aligned} \tag{74}$$

Using Taylor's theorem and Assumption 2,

$$\Phi^{(i)}(\widehat{\mathbf{w}}^{(i)}) - \Phi^{(i)}(\overline{\mathbf{w}}^{(i)}) - \big(\bar{\mathbf{x}}^{(i)}\big)^T\big(\widehat{\mathbf{w}}^{(i)} - \overline{\mathbf{w}}^{(i)}\big) \geq L\|\widetilde{\mathbf{w}}^{(i)}\|_2^2. \tag{75}$$

Inserting (75) into (74),

$$(1/M)\sum_{i\in\mathcal{M}} \big[-\big(\boldsymbol{\varepsilon}^{(i)}\big)^T\widetilde{\mathbf{w}}^{(i)} + L\|\widetilde{\mathbf{w}}^{(i)}\|_2^2\big] \leq \lambda\|\widetilde{\mathbf{w}}\|_{\partial\mathcal{P}}. \tag{76}$$

Combining (70) with (76),

$$L\|\widetilde{\mathbf{w}}\|_{\mathcal{M}}^2 \leq \lambda\|\widetilde{\mathbf{w}}\|_{\partial\mathcal{P}} + \kappa\lambda\|\widetilde{\mathbf{w}}\|_{\mathcal{M}}. \tag{77}$$

Combining (73) with (23) yields

$$\|\widetilde{\mathbf{w}}\|_{\partial\mathcal{P}} \leq \kappa\|\widetilde{\mathbf{w}}\|_{\mathcal{M}} \tag{78}$$

and, in turn via (77),

$$\|\widetilde{\mathbf{w}}\|_{\mathcal{M}} \leq 2\lambda\kappa/L. \tag{79}$$

Inserting (79) into (78) and (73),

$$\begin{aligned}
\|\widetilde{\mathbf{w}}\|_{TV} &= \|\widetilde{\mathbf{w}}\|_{\partial\mathcal{P}} + \|\widetilde{\mathbf{w}}\|_{\mathcal{E}\setminus\partial\mathcal{P}} \\
&\overset{(73)}{\leq} \|\widetilde{\mathbf{w}}\|_{\partial\mathcal{P}} + 3\|\widetilde{\mathbf{w}}\|_{\partial\mathcal{P}} + \kappa\|\widetilde{\mathbf{w}}\|_{\mathcal{M}} \\
&\overset{(78)}{\leq} 5\kappa\|\widetilde{\mathbf{w}}\|_{\mathcal{M}} \\
&\overset{(79)}{\leq} 10\lambda\kappa^2/L.
\end{aligned} \tag{80}$$

According to (80), we can ensure a prescribed error level $\|\widetilde{\mathbf{w}}\|_{TV} \leq \eta$ by setting ($L > 1$)

$$\lambda := \eta/(5\kappa^2). \tag{81}$$

The final step of the proof is to control the probability of (70) to hold. By Corollary 3, (70) holds if

$$\max_{\mathcal{C}_l\in\mathcal{P}} (1/|\mathcal{C}_l|)\big\|\sum_{i\in\mathcal{C}_l}\boldsymbol{\varepsilon}_i\big\|_2 \leq (\lambda/2)\kappa, \tag{82}$$

and simultaneously

$$\max_{\mathcal{C}_l\in\mathcal{P}} \big\|\big(\mathbf{D}_{\mathcal{C}_l}^{\dagger}\big)^T\boldsymbol{\varepsilon}_{\mathcal{C}_l}\big\|_{2,\infty} \leq M\lambda/4. \tag{83}$$

We first bound the probability that (82) fails to hold. For a particular cluster $\mathcal{C}_l$, (65) yields

$$P\{(1/|\mathcal{C}_l|)\| \sum_{i\in\mathcal{C}_l} \boldsymbol{\varepsilon}_i \|_2 \leq (\lambda/2)\kappa\} \leq 2\exp\left(- \frac{|\mathcal{C}_l|\lambda^2\kappa^2}{8dU}\right). \tag{84}$$

Combining this with a union bound over all $\mathcal{C}_l \in \mathcal{P}$ yields

$$P\{\text{“(82) invalid”}\} \leq 2|\mathcal{P}| \max_{l=1,\ldots,|\mathcal{P}|} \exp\left(- \frac{|\mathcal{C}_l|\lambda^2\kappa^2}{8dU}\right). \tag{85}$$

For controlling the probability of (83) failing to hold, we combine (18) with Lemma 4. This yields, using a union bound over all edges $e \in \mathcal{E}$,

$$P\{\text{“(83) invalid”}\} \leq 2|\mathcal{E}| \exp\left(- \frac{M\rho_{\mathcal{P}}^2\lambda^2}{64Ud\|\mathbf{A}\|_\infty^2}\right). \tag{86}$$

A union bound yields (24) by summing the bounds (85) and (86) for the choice (81).

ACKNOWLEDGMENT

Roope Tervo from Finnish Meteorological Institute provided a Python script to download the weather data used in Section VII-B.

REFERENCES

[1] H. Ambos, N. Tran, and A. Jung, “Classifying big data over networks via the logistic network lasso,” in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, Oct. 2018, pp. 855–858.
[2] S. Arora, R. Ge, F. Koehler, T. Ma, and A. Moitra, “Provable algorithms for inference in topic models,” in *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, New York, NY, USA, 2016, pp. 2859–2867.
[3] A.-L. Barabási, N. Gulbahce, and J. Loscalzo, “Network medicine: A network-based approach to human disease,” *Nature Rev. Genet.*, vol. 12, no. 1, pp. 56–68, Jan. 2011.
[4] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation,” *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*, vol. 3. Hanover, MA, USA: Now Publishers, 2010.
[6] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[7] P. Bühlmann and S. van de Geer, *Statistics for High-Dimensional Data*. New York, NY, USA: Springer, 2011.
[8] 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, pp. 120–145, May 2011.
[9] J. Chang and D. M. Blei, “Relational topic models for document networks,” in *Proc. 12th Int. Conf. Artif. Intell. Statist. (AISTATS)*, vol. 5. Clearwater Beach, FL, USA: JMLR, 2009.
[10] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. Cambridge, MA, USA: MIT Press, 2006.
[11] S. Chen, R. Varma, A. Sandryhaila, and J. Kovacevic, “Discrete signal processing on graphs: Sampling theory,” *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6510–6523, Dec. 2015.
[12] S. Chen, R. Varma, A. Singh, and J. Kovacevic, “Representations of piecewise smooth signals on graphs,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Shanghai, China, Mar. 2016, pp. 6370–6374.
[13] L. Condat, “A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms,” *J. Optim. Theory Appl.*, vol. 158, no. 2, pp. 460–479, Aug. 2013.
[14] S. Cui, A. Hero, Z.-Q. Luo, and J. M. F. Moura, Eds., *Big Data Over Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2016.
[15] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, “Learning graphs from data: A signal representation perspective,” *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 44–63, May 2019.
[16] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*. New York, NY, USA: Springer, 2012.
[17] D. Goldfarb and W. Yin, “Parametric maximum flow algorithms for fast total variation minimization,” *SIAM J. Sci. Comput.*, vol. 31, no. 5, pp. 3712–3743, Jan. 2009.
[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[19] L. Grady, “Random walks for image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006.
[20] D. Hallac, J. Leskovec, and S. Boyd, “Network lasso: Clustering and optimization in large graphs,” in *Proc. SIGKDD*, 2015, pp. 387–396.
[21] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning* (Springer Series in Statistics). New York, NY, USA: Springer, 2001.
[22] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning With Sparsity. The Lasso and Its Generalizations*. Boca Raton, FL, USA: CRC Press, 2015.
[23] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.
[24] J.-C. Hütter and P. Rigollet, “Optimal rates for total variation denoising,” in *Proc. Annu. Conf. Learn. Theory*, vol. 49, Jun. 2016, pp. 1115–1146.
[25] A. Jung, “Learning the conditional independence structure of stationary time series: A multitask learning approach,” *IEEE Trans. Signal Process.*, vol. 63, no. 21, pp. 5677–5690, Nov. 2015.
[26] A. Jung, “On the complexity of sparse label propagation,” *Frontiers Appl. Math. Statist.*, vol. 4, p. 22, Jul. 2018.
[27] A. Jung, N. T. Quang, and A. Mara, “When is network lasso accurate?” *Frontiers Appl. Math. Stat.*, vol. 3, p. 28, Jan. 2018.
[28] A. Jung and N. Tran, “Localized linear regression in networked data,” *IEEE Signal Process. Lett.*, vol. 26, no. 7, pp. 1090–1094, Jul. 2019.
[29] A. Jung and N. Vesselinova, “Analysis of network lasso for semi-supervised regression,” in *Proc. 22nd Int. Conf. Artif. Intell. Stat. (AISTATS)*, Okinawa, Japan, Apr. 2019, pp. 380–387.
[30] K. Yang, T. Jiang, Y. Shi, and Z. Ding, “Federated learning via over-the-air computation,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.
[31] B. J. Lengerich, B. Aragam, and E. P. Xing, “Personalized regression enables sample-specific pan-cancer analysis,” *Bioinformatics*, vol. 34, no. 13, pp. i178–i186, Jul. 2018.
[32] T. Li, E. Levina, and J. Zhu, “Prediction models for network-linked data,” *Ann. Appl. Statist.*, vol. 13, no. 1, pp. 132–164, Mar. 2019.
[33] M. De Donno, K. Tange, and N. Dragoni, “Foundations and evolution of modern computing paradigms: Cloud, IoT, edge, and fog,” *IEEE Access*, vol. 7, pp. 150936–150948, 2019.
[34] A. Mokhtari and A. Ribeiro, “Global convergence of online limited memory BFGS,” *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 3151–3181, Jan. 2015.
[35] B. C. Muzyka, “Host factors affecting disease transmission,” *Dental Clinics North Amer.*, vol. 40, no. 2, p. 263, 1996.
[36] É. A. Nadaraya, “On non-parametric estimates of density functions and regression curves,” *Theory Probab. Appl.*, vol. 10, no. 1, pp. 186–190, Jan. 1965.
[37] B. Nadler, N. Srebro, and X. Zhou, “Statistical analysis of semi-supervised learning: The limit of infinite unlabelled data,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1330–1338.
[38] M. E. J. Newman, *Networks: An Introduction*. Oxford, U.K.: Oxford Univ. Press, 2010.
[39] R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. I. Jordan, “A general analysis of the convergence of ADMM,” in *Proc. 32nd Int. Conf. Mach. Learn.*, Lille, France, vol. 37, 2015, pp. 343–352.
[40] E. Ollier and V. Viallon, “Regression modeling on stratified data with the lasso,” *Biometrika*, vol. 104, no. 1, pp. 83–96, 2017.
[41] N. Parikh and S. Boyd, “Proximal algorithms,” *Found. Trends Optim.*, vol. 1, no. 3, pp. 123–231, 2013.
[42] T. Pock and A. Chambolle, “Diagonal preconditioning for first order primal-dual algorithms in convex optimization,” in *Proc. IEEE ICCV*, Barcelona, Spain, Nov. 2011, pp. 1762–1769.
[43] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 1970.
[44] C. Rother, V. Kolmogorov, and A. Blake, “‘GrabCut’—Interactive foreground extraction using iterated graph cuts,” in *Proc. ACM Trans. Graph. (SIGGRAPH)*, 2004, pp. 1–7.

[45] W. Rudin, *Principles of Mathematical Analysis*, 3rd ed. New York, NY, USA: McGraw-Hill, 1976.
[46] M. R. Sabuncu, B. T. T. Yeo, K. Van Leemput, B. Fischl, and P. Golland, "A generative model for image segmentation based on label fusion," *IEEE Trans. Med. Imag.*, vol. 29, no. 10, pp. 1714–1729, Oct. 2010.
[47] N. Tran, H. Ambos, and A. Jung, "Classifying partially labeled networked data VIA logistic network lasso," in *Proc. ICASSP-IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020, pp. 3832–3836.
[48] J. Tuck, S. Barratt, and S. Boyd, "A distributed method for fitting Laplacian regularized stratified models," 2019, *arXiv:1904.12017*. [Online]. Available: http://arxiv.org/abs/1904.12017
[49] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, Dec. 2007.
[50] M. J. Wainwright and M. I. Jordan, *Graphical Models, Exponential Families, and Variational Inference, Volume 1 of Foundations and Trends in Machine Learning*. Hanover, MA, USA: Now Publishers, 2008.
[51] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropol. Res.*, vol. 33, no. 4, pp. 452–473, Dec. 1977.
[52] P. Zhang, C. Moore, and L. Zdeborová, "Phase transitions in semisupervised clustering of sparse networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 90, no. 5, Nov. 2014, Art. no. 052802.

ALEXANDER JUNG (Member, IEEE) received the Ph.D. degree (Hons.) from TU Vienna, in 2012 (Austria's Promotio sub auspiciis Praesidentis rei publicae). He is currently an Assistant Professor of machine learning with the Department of Computer Science, Aalto University. His research interest includes massive datasets with intrinsic network structure. Together with his collaborators, he pioneered the characterization of network structure and statistical properties of data that allow for accurate learning from big data over networks. The excellence of his research work is documented by numerous publications in top-tier journals. He is the first author of a paper that received a Best Student Paper Award at the premium signal processing conference IEEE ICASSP, in 2011. In 2018, he was awarded an Amazon Web Services (AWS) Machine Learning Award. He has been chosen as Teacher of the Year by the Department of Computer Science, in 2018. He currently serves as an Associate Editor for the IEEE SIGNAL PROCESSING LETTERS and the Chair of the Joint Chapter for Signal Processing and Circuit and Systems within the IEEE Finland Section.