LEARNING NETWORKED EXPONENTIAL FAMILIES WITH NETWORK LASSO

Alexander Jung[1]

[1]Department of Computer Science, Aalto University, Espoo, Finland; firstname.lastname(at)aalto.fi

ABSTRACT

We propose networked exponential families to jointly leverage the information in the topology as well as the attributes (features) of networked data points. Networked exponential families are a flexible probabilistic model for heterogeneous datasets with intrinsic network structure. These models can be learnt efficiently using network Lasso which implicitly pools or clusters the data points according to the intrinsic network structure and the local likelihood. The resulting method can be formulated as a non-smooth convex optimization problem which we solve using a primal-dual splitting method. This primal-dual method is appealing for big data applications as it can be implemented as a highly scalable message passing algorithm.

1. INTRODUCTION

The data generated in many important application domains have an intrinsic network structure. Such networked data arises in the study of social networks, text document collections and personalized medicine [3], [7], [40]. Network science provides powerful tools for the analysis of such data based on its intrinsic network structure [11], [31]. The network structure of datasets is complemented by the information contained in attributes (such as features or labels) of individual data points [7].

Consolidating prior work on networked (generalized) linear models [23], [28], we propose networked exponential families as a flexible probabilistic model for heterogenous and noisy data with an intrinsic network structure. By coupling the (node-wise) local parameters of an exponential family [39], we jointly capitalize on network structure and the information conveyed by the features and labels of data points.

Networked exponential families are powerful statistical models for many important application domains such as personalized (high-precision) health-care [26], or natural language processing [4], [7]. In contrast to [7], which uses a probabilistic model for the network structure of text corpora, this paper assumes the network structure as fixed and known.

To learn networked exponential families, this paper implements the network Lasso in order to simultaneously cluster and optimize a probabilistic model [16]. The implementation of nLasso is based on a primal-dual method which results in scalable message passing over the underlying data network. In contrast, to state-of-the art graph clustering methods which only use network structure, nLasso in networked exponential families jointly capitalizes on network structure and the information provided by observed node attributes. Joint clustering and optimization has been considered in [41] for probabilistic models of the network structure. In contrast, this paper considers the network structure fixed and given and use a probabilistic model for the node attributes (features and labels).

The idea of borrowing inferential power across networked data has also been used for bandit models in sequential decision making problems [13], [27]. In particular, the clustering bandit model coupled individual linear bandit models for nodes (representing users) using a domain-specific notion of similarity such as "friendship" relations in a social network.

The closest to this work is [28] 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 paper allows for arbitrarily varying weight vectors (see end of Sec. 2).

Another main difference between [28] and our approach is the choice of regularizer for the networked model. While [28], similar to most existing work on semi-supervised learning [8], 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 [14].

Minimizing the Laplacian quadratic form is a smooth convex problem resulting in a linear system. In contrast, TV minimization is a non-smooth convex optimization problem which requires more advanced techniques such as proximal methods [6], [33] (see Sec. 6). The higher computational cost of TV minimization affords improved accuracy when learning from a small number of labeled data points [30].

In order to learn networked exponential families, this paper applies the network Lasso (nLasso). The nLasso has been proposed recently as a natural extension of the Lasso to networked data [16], [17]. We show how the nLasso can be implemented efficiently using a primal dual splitting method for convex optimization. The resulting scalable learning method amounts to a message passing protocol over the data network structure.

Contribution. The main contributions of this paper are:

- The introduction of networked exponential families as a probabilistic model for networked data.
- Extending prior [23], [24], a bound on the nLasso error for general networked exponential families is presented.
- A scalable nLasso implementation using a primal-dual method for convex optimization. The proposed formu-

lation generalizes the method in [1] (for logistic regression) to arbitrary exponential families.
- Verification of computational and statistical properties of the proposed method using numerical experiments.

Outline. We introduce networked exponential families in Sec. 2. Sec. 3 details how some recently proposed models for networked data are obtained as special cases of networked exponential families. In Sec. 4, 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 Sec. 5. Sec. 6 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 Sec. 7.

Notation. We denote the ℓ_2-norm of a vector as $\|\mathbf{x}\| := \sqrt{\mathbf{x}^T\mathbf{x}}$. 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 of size $d \times d$.

2. NETWORKED EXPONENTIAL FAMILIES

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

In what follows, we assume the empirical graph $\mathcal{G}$ fixed and known. The network structure might be induced by physical proximity (in time or space), physical connection (communication networks) or statistical dependency (probabilistic graphical models) [25]. The learning of network structure in a data-driven fashion [12], [20] is beyond the scope of this paper.

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 [39]

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

The distribution (2) is parametrized by the (unknown) weight vectors $\overline{\mathbf{w}}^{(i)}$. These weight vectors as fixed (deterministic) but unknown and the main focus of this paper is the accurate estimation of these weight vectors.

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 space of all such vector-valued graph signals is

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

Similarly, we define the space of all vector-valued signals defined on the edges $\mathcal{E}$ of the empirical graph 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 properties of the exponential family (2) can be read off the log-partition or cumulant function [39]

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

The Fisher information matrix (FIM) $\mathbf{F}^{(i)}$ for (2) is the Hessian

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

The conditioning of $\mathbf{F}^{(i)}$ crucially influences the statistical and computational properties of the model (2) (see Sec. 5 and 6).

Within a networked exponential family, the node-wise models (2) 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)

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

Requiring the weight vectors $\mathbf{w}^{(i)}$, for $i \in \mathcal{V}$, to have small TV forces weight vectors 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:

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

Let us finally compare networked exponential families, obtained as the combination of (2) with a constraint on the TV (8) of weights $\overline{\mathbf{w}}$ in (8), and the RNC model put forward in [28]. First, RNC considers the special case of distributions (2) with $\mathbf{t}^{(i)} = \big(\big(\mathbf{x}^{(i)}\big)^T, 1\big)^T$ and a partitioned weight vector $\overline{\mathbf{w}} = \big(\boldsymbol{\beta}^T, \alpha^{(i)}\big)^T$ with a shared weight vector $\boldsymbol{\beta}$ which is the same for all nodes $i \in \mathcal{V}$. The intercept $\alpha^{(i)}$ is allowed to vary over nodes. 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 measure how well the weight vectors conform with the network structure of the data.

3. SOME EXAMPLES

We now discuss important special cases of the model (2).

3.1. Networked Linear Regression

Consider a networked dataset whose data points $i \in \mathcal{V}$ are characterized by features $\mathbf{x}^{(i)} \in \mathbb{R}^d$ and numeric labels $y^{(i)} \in \mathbb{R}$. Maybe the most basic (yet quite useful) model for the relation between features and labels 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) [23].

The model (9) is obtained as the special case of the exponential family (2) for the scalar attributes $z^{(i)} := y^{(i)}$ 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 it is difficult to obtain accurate label information, i.e., $y^{(i)}$ is not known for some data point $i \in \mathcal{V}$. One approach to handle such partially labeled data is to use some crude estimates $\hat{y}^{(i)}$ of the labels for unlabelled nodes. We can account for varying label accuracy using heterogeneous noise variables $\varepsilon^{(i)}$. In particular, we use a larger noise variance σ_i^2 for a node $i \in \mathcal{V}$ for which we only have an estimate $\hat{y}^{(i)}$.

3.2. Networked Logistic Regression

Consider networked data points $i \in \mathcal{V}$ each characterized by features $\mathbf{x}^{(i)} \in \mathbb{R}^d$ and binary labels $y^{(i)} \in \{-1, 1\}$. Logistic regression models the relation between features and labels via

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

The distribution (10) is parametrized by the weight vector $\mathbf{w}^{(i)}$ for each node $i \in \mathcal{V}$. It can be shown that (10) is the posterior distribution of label $y^{(i)}$ given the features $\mathbf{x}^{(i)}$ if the features $\mathbf{x}^{(i)}$ is a Gaussian random vector conditioned on $y^{(i)}$.

Networked logistic regression requires the weight vectors in the node-wise models (10) to have a small TV (7) [1].

We obtain the logistic regression model (10) as the special case of the exponential family (2) for the scalar node attributes $z^{(i)} := y^{(i)}$ with $\mathbf{t}^{(i)}(z) = \mathbf{x}^{(i)} z/2$ and $\Phi^{(i)}(\mathbf{w}) = \log\big(\exp\big(\mathbf{w}^T\mathbf{x}^{(i)}/2\big) + \exp\big(-\mathbf{w}^T\mathbf{x}^{(i)}/2\big)\big)$.

3.3. 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], [39]. 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], [39]).

4. NETWORK LASSO

The goal of this paper is to develop a method for learning an accurate estimate $\widehat{\mathbf{w}}^{(i)}$ for the true weights $\overline{\mathbf{w}}^{(i)}$ (see (2)). The learning of the weight vectors $\mathbf{w}^{(i)}$ is based on the availability of 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 can be 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 (12) is equivalent to minimizing

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

Criterion (12) is not enough to learn the weights $\mathbf{w}^{(i)}$ for all $i \in \mathcal{V}$. Indeed, (12) ignores weights $\widehat{\mathbf{w}}^{(i)}$ at unobserved nodes $i \in \mathcal{V} \setminus \mathcal{M}$. Therefore, we impose additional structure on the weight vectors. Any reasonable estimate $\widehat{\mathbf{w}}^{(i)}$ should conform with the *cluster structure* of the empirical graph $\mathcal{G}$ [31].

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)

$$\widehat{\mathbf{w}} \in \operatorname*{arg\,min}_{\mathbf{w}\in\mathcal{W}} \widehat{E}(\mathbf{w}) + \lambda \|\mathbf{w}\|_{\rm TV}. \tag{13}$$

The learning problem (13) is an instance of the generic nLasso problem [16]. The parameter λ in (13) allows to trade-off small TV $\|\widehat{\mathbf{w}}\|_{\rm TV}$ against small error $\widehat{E}(\widehat{\mathbf{w}})$ (cf. (12)). Chosing λ can be based on validation [17] or the error analysis in Sec. 5.

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

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

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

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

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

$$\widehat{\mathbf{w}} \in \operatorname*{arg\,min}_{\mathbf{w}\in\mathcal{W}} h(\mathbf{w}) + g(\mathbf{D}\mathbf{w}), \tag{16}$$

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

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

with the weight matrix $\mathbf{A}$ (see (1)) and the "degree matrix"

$$\mathbf{\Lambda} = \mathrm{diag}\{d_1, \ldots, d_N\} \in \mathbb{R}^{N \times N}, \text{ with } d_i := \sum_{\{j,i\} \in \mathcal{E}} A_{i,j}.$$

The eigenvalues of $\mathbf{L}$ reflect the connectivity of the graph $\mathcal{G}$. A graph $\mathcal{G}$ is connected if and only if $\lambda_2 > 0$, with λ_2 being the smallest non-zero eigenvalue. 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., [18]).

5. ANALYSIS OF NLASSO ESTIMATION ERROR

We now characteize the 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 $\mathcal{C}_l$. The model (19) is a special case of piece-wise polynomial signal model which allows the weight vectors to vary within each cluster [9].

The model (19), which is used in [13] for networked bandit models, is meant to provide predictors that approximate the observed data well. The analysis below indicates that nLasso methods are robust to model mismatch, i.e., the true underlying weight vectors in (2) can be approximated well 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*

$$\rho_{\mathcal{P}} := \min_{\mathcal{C}_l \in \mathcal{P}} \rho(\mathcal{C}_l). \tag{20}$$

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 Sec. 6), we do not need any information about the partition $\mathcal{P}$.

Assumption 2. *The FIM $\mathbf{F}^{(i)}$ (see* (6)*) is bounded as $U\mathbf{I} \succeq \mathbf{F}^{(i)} \succeq L\mathbf{I}$ for any weights $\overline{\mathbf{w}}$ with some constant $L > 1$.*

Assumption 3. *There are constants $K, L > 1$ such that for any $\mathbf{z} \in \mathcal{W}$ (see* (3)*) which is piece-wise constant on partition $\mathcal{P}$,*

$$L\|\mathbf{z}\|_{\partial\mathcal{P}} \leq K\|\mathbf{z}\|_{\mathcal{M}} + \|\mathbf{z}\|_{\overline{\partial\mathcal{P}}}. \tag{21}$$

The main analytic result of this paper 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 Asspt.* 1*,* 2 *and* 3 *are satisfied with (see* 21*)*

$$L > 3, \text{ and } K \in (1, L-2), \tag{22}$$

and corresponding condition number $\kappa := \frac{K+3}{L-3} > 1$. Based on the observed noisy labels y_i, we estimate the underlying weight vectors $\overline{\mathbf{w}}$ using a solution $\widehat{\mathbf{w}}$ to the nLasso problem (13) *with $\lambda := \eta/(5\kappa^2)$ using some pre-specified error level $\eta > 0$. Then,*

$$\mathrm{P}\{\|\widehat{\mathbf{w}} - \overline{\mathbf{w}}\|_{\mathrm{TV}} \geq \eta\} \leq 2|\mathcal{P}| \max_{l=1,\ldots,|\mathcal{P}|} \exp\left(-\frac{|\mathcal{C}_l|\eta^2}{8 \cdot 25dU\kappa^2}\right) + 2|\mathcal{E}| \exp\left(-\frac{M\rho_{\mathcal{P}}^2\eta^2}{64 \cdot 25Ud\|\mathbf{A}\|_\infty^2\kappa^4}\right). \tag{23}$$

The bound (23) 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 Alg. 1 scales with the fourth power of the condition number $\kappa = \frac{K+3}{L-3}$ (see Asspt. 3) and inversely with the spectral gap $\rho_{\mathcal{P}}$ of the partitioning $\mathcal{P}$. Thus, nLasso methods (13) (such as Alg. 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 Asspt. 3) typically requires the edges within clusters to have larger weights on average than the weights of the boundary edges. Moreover, it is reasonable that nLasso tends to be more accurate for a larger spectral gap $\rho_{\mathcal{P}}$, which requires the nodes within each cluster $\mathcal{C}_l$ to be well connected. Indeed, an graph $\mathcal{G}$ consisting of well-connected clusters $\mathcal{C}_l$ favours clustered graph signals (see (19)) as solutions of nLasso (13).

6. A PRIMAL-DUAL METHOD

The nLasso (16) is a convex optimization problem with a non-smooth objective function which rules out the use of gradient descent methods. However, the objective function is highly structured since it is the sum of a smooth convex function $h(\mathbf{w})$ and a non-smooth convex function $g(\mathbf{D}\mathbf{w})$, which can be optimized efficiently when considered separately. This suggests to use some proximal method [33] for solving (16).

One particular example of a proximal method is the alternating direction method of multipliers (ADMM) which has been considered in [16]. However, we will choose another type of proximal method which is based on a dual problem to (16) [6], [34]. These primal-dual methods are attractive since their analysis provides natural choices for the algorithm parameters. In contrast, tuning the ADMM parameter is non-trivial [32].

6.1. Primal-Dual Method

The preconditioned primal-dual method [34] launches from reformulating the problem (16) as a saddle-point problem

$$\min_{\mathbf{w}\in\mathbb{R}^{dN}} \max_{\mathbf{u}\in\mathcal{D}} \mathbf{u}^T\mathbf{D}\mathbf{w} + h(\mathbf{w}) - g^*(\mathbf{u}), \tag{24}$$

with the convex conjugate g^* of g [6].

Any solution $(\widehat{\mathbf{w}}, \widehat{\mathbf{u}})$ of (24) is characterized by [35]

$$-\mathbf{D}^T\widehat{\mathbf{u}} \in \partial h(\widehat{\mathbf{w}}), \text{ and } \mathbf{D}\widehat{\mathbf{w}} \in \partial g^*(\widehat{\mathbf{u}}). \tag{25}$$

This condition is, in turn, equivalent to

$$\begin{aligned} \widehat{\mathbf{w}} - \mathbf{T}\mathbf{D}^T\widehat{\mathbf{u}} &\in (\mathbf{I}_{dN} + \mathbf{T}\partial h)(\widehat{\mathbf{w}}), \text{ and} \\ \widehat{\mathbf{u}} + \mathbf{\Sigma}\mathbf{D}\widehat{\mathbf{w}} &\in (\mathbf{I}_{dE} + \mathbf{\Sigma}\partial g^*)(\widehat{\mathbf{u}}), \end{aligned} \tag{26}$$

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 (26) lends naturally to the following coupled fixed point iterations [34]

$$\mathbf{w}_{k+1} = (\mathbf{I} + \mathbf{T}\partial h)^{-1}(\mathbf{w}_k - \mathbf{T}\mathbf{D}^T\mathbf{u}_k) \tag{27}$$

$$\mathbf{u}_{k+1} = (\mathbf{I} + \mathbf{\Sigma}\partial g^*)^{-1}(\mathbf{u}_k + \mathbf{\Sigma}\mathbf{D}(2\mathbf{w}_{k+1} - \mathbf{w}_k)). \tag{28}$$

If the matrices $\mathbf{\Sigma}$ and $\mathbf{T}$ in (27), (28) satisfy

$$\|\mathbf{\Sigma}^{1/2}\mathbf{D}\mathbf{T}^{1/2}\|^2 < 1, \tag{29}$$

the sequence $\mathbf{w}_{k+1}$ (see (27), (28)) converges to a solution of (13) [34, Thm. 1]. The condition (29) is satisfied for

$$\mathbf{\Sigma} := \mathrm{diag}\{(1/(2A_e))\mathbf{I}\}_{e\in\mathcal{E}}, \mathbf{T} := \mathrm{diag}\{(\tau/d^{(i)})\mathbf{I}\}_{i\in\mathcal{V}}, \tag{30}$$

with $d^{(i)} = \sum_{j\neq i} A_{ij}$ and some $\tau < 1$ [34, Lem. 2].

The update (28) involves the resolvent operator

$$(\mathbf{I} + \mathbf{\Sigma}\partial g^*)^{-1}(\mathbf{v}) = \operatorname*{arg\,min}_{\mathbf{v}'\in\mathcal{D}} g^*(\mathbf{v}') + (1/2)\|\mathbf{v}' - \mathbf{v}\|^2_{\mathbf{\Sigma}^{-1}}, \tag{31}$$

where $\|\mathbf{v}\|_{\mathbf{\Sigma}} := \sqrt{\mathbf{v}^T\Sigma\mathbf{v}}$. 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 [33, Sec. 6.5]

$$\begin{aligned} &((\mathbf{I}_{dE} + \mathbf{\Sigma}\partial g^*)^{-1}(\mathbf{v}))^{(e)} \\ &\overset{(31)}{=} \operatorname*{arg\,min}_{\mathbf{v}'\in\mathbb{R}^d} g_2^*(\mathbf{v}') + (1/(2\sigma^{(e)}))\|\mathbf{v}' - \mathbf{v}^{(e)}\|^2 \\ &= \mathbf{v}^{(e)} - \sigma^{(e)}(\mathbf{I}_d + (\lambda/\sigma^{(e)})\partial\|.\|)^{-1}(\mathbf{v}^{(e)}/\sigma^{(e)}) \\ &= \begin{cases} \lambda\mathbf{v}^{(e)}/\|\mathbf{v}^{(e)}\| & \text{if } \|\mathbf{v}^{(e)}\| > \lambda \\ \mathbf{v}^{(e)} & \text{otherwise,} \end{cases} \end{aligned}$$

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

The update (27) 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 (30), the update (27) decomposes into independent node-wise updates

$$\mathbf{w}_{k+1}^{(i)} := \begin{cases} \operatorname*{arg\,min}_{\mathbf{w}\in\mathbb{R}^d} g^{(i)}(\mathbf{w}) \text{ for } i \in \mathcal{M} \\ \overline{\mathbf{w}}^{(i)} \text{ for } i \in \mathcal{V}\setminus\mathcal{M} \end{cases} \tag{32}$$

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

$$\overline{\mathbf{w}} := \mathbf{w}_k - \mathbf{T}\mathbf{D}^T\mathbf{u}_k. \tag{33}$$

It is important to note that the update (32), for $i \in \mathcal{M}$, amounts to a regularized maximum likelihood estimator for exponential families [39, Eq. 3.38]. The regularization term $\tilde{\tau}^{(i)}\|\mathbf{w} - \overline{\mathbf{w}}^{(i)}\|^2$, which varies as iterations proceed, 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 (33)).

In general, there is no closed-form solution for the update (32). However, the update (32) is a smooth convex optimization problem that can be solved efficiently using iterative methods such as L-BGFS [29]. We detail a computationally cheap iterative method for approximately solving (32) in Sec. 6.3.

Let us denote the approximate solution to (32) by $\widehat{\mathbf{w}}_{k+1}^{(i)}$ and assume that it is sufficiently accurate such that

$$e_k = \|\widehat{\mathbf{w}}_{k+1}^{(i)} - \mathbf{w}_{k+1}^{(i)}\| \leq 1/k^2. \tag{34}$$

Thus, we require the approximation quality (for approximating the update (32)) to increase with the iteration number k. According to [10, Thm. 3.2], the error bound (34) ensures the sequences obtained by (27) and (28) when replacing the exact update (32) with the approximation $\widehat{\mathbf{w}}_{k+1}$ still converge to a saddle-point of (24) 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 (30), $k := 0$, $\widehat{\mathbf{w}}_0 := 0$, $\widehat{\mathbf{u}}_0 := 0$

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 (32)
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: $(\widehat{\mathbf{w}}_k, \widehat{\mathbf{u}}_k)$.

The primal-dual implementation of nLasso in Alg. 1 requires only the empirical graph along with the observed node attributes $\mathbf{z}^{(i)}$, for $i \in \mathcal{M}$, as input. As already mentioned above, Alg. 1 does not require any specification of a partition of the empirical graph. Moreover, in contrast to the ADMM implementation of nLasso (see [16, Alg. 1]), the proposed Alg. 1 does not involve unspecified tuning parameters.

6.2. Computational Complexity

It can be shown that Alg. 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 required for a single message depends on the particular instance of the update (32).

For a fixed number of iterations used for Alg. 1, its 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 Alg. 1 depends crucially on the number of iterations required to achieve accurate learning. A worst-case analysis shows that, for exact updates in (32), the number of iterations scales inversely with the required estimation accuracy [6]. Moreover, this convergence speed cannot be improved for chain graphs [21].

6.3. Approximate Primal Update

We now detail a simple iterative method for computing an approximate solution $\widehat{\mathbf{w}}_{k+1}^{(i)}$ to the primal update (32). A solution $\widehat{\mathbf{w}}$ of (32) is characterized by the zero gradient condition [5]

$$\nabla f(\widehat{\mathbf{w}}) = \mathbf{0} \tag{35}$$

with

$$f(\mathbf{w}) := -\mathbf{w}^T\mathbf{z}^{(i)} + \Phi^{(i)}(\mathbf{w}) + \tilde{\tau}^{(i)}\|\mathbf{w}-\overline{\mathbf{w}}^{(i)}\|^2. \tag{36}$$

Inserting (36) into (35), and using some basic calculus,

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

The necessary and sufficient condition (37) (for $\mathbf{w}^{(i)}$ to solve (32)) 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 [37, 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 (32).

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 (39) as (see [37, Proof of Thm. 9.23])

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

Thus, if we use the approximation $\widehat{\mathbf{w}}_{k+1}^{(i)} := \widetilde{\mathbf{w}}^{(r)}$ for the update (32), we can ensure (34) by iterating (40) for at least

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

Note that computing the iterates (40) requires the evaluation of the gradient $\nabla\Phi^{(i)}(\widetilde{\mathbf{w}}^{(r)})$ of the log partition function $\Phi^{(i)}(\mathbf{w})$. According to [39, Prop. 3.1.],

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

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 [39, 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. 3).

6.4. Partially Observed Models

The learning Algorithm 1 can be adapted easily to cope with partially observed exponential families [39]. In particular, for the networked LDA described in Sec. 3, 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 (32) we would also need the values 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 (32) by some "Expectation-Maximization" (EM) principle (see [39, 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].

7. NUMERICAL EXPERIMENTS

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

7.1. Two-Cluster Dataset

We generate the 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, i.e., $|\mathcal{M} \cap \mathcal{C}^{(1)}| = |\mathcal{M} \cap \mathcal{C}^{(2)}| = 3$.

As shown in [22] the performance of nLasso type methods (for learning problems similar to but different from (2)) 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. In order to quantify the connectivity of the labeled 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 $|\partial|$.

In Fig. 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 Alg. 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). Accorcind to Fig. 1 there are two regimes of levels of connectivity. For sufficiently large connectivity $\bar{\rho} > \sqrt{2}$, Alg. 1 is able to capitalize on the network structure in order to learn the piece-wise constant weight vectors $\mathbf{w}^{(i)}$.

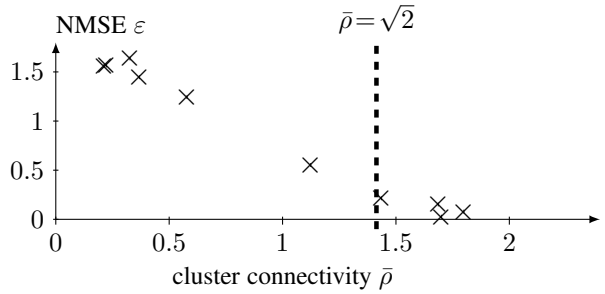

Fig. 1. nLasso error for networked linear regression.

7.2. 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 use Alg. 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 Alg. 1 (using $\lambda = 1/7$ and a fixed number of 10^4 iterations). The normalized average squared prediction error is $\approx 10^{-1}$ and only slightly larger than the prediction error incurred by fitting a single linear model to the cluster $\mathcal{C}$.

7.3. Image Segmentation

This experiment revolves around using Alg. 1 for image segmentation [15], [36]. 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 Fig. 2).

Different nodes i, j 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 $r^{(i)} := x_1^{(i)} / \max_{j \in \mathcal{V}} x_1^{(j)}$,

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

We apply Alg. 1, with $\lambda = 100$ and fixed number of 10 iterations, to learn the weights $\mathbf{w}^{(i)}$ for a networked logistic regression model (see Sec. 3.2). For the update (32) in Alg. 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 Fig. 2. The middle of Fig. 2 depicts the hard segmentation obtained by the "GrabCut" method [36]. Using MATLAB version 19 on a standard laptop, Alg. 1 is almost ten times faster than GrabCut.

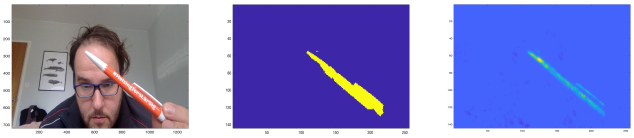

Fig. 2. Left: Original image. Middle: Grabcut. Right: Alg. 1

8. 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). In particular, the analysis underlying Sec. 5 might guide the design of network structure by relating Asspt. 3 to network flow problems (see [24]).

ACKNOWLEDGMENTS

We thank Roope Tervo from the Finnish Meteorological Institute for providing a Python script that allows to download weather data from the FMI.

9. REFERENCES

[1] H. Ambos, N. Tran, and A. Jung. Classifying big data over networks via the logistic network lasso. In *Proc. Asilomar Conf. on Sig., Sys., and Comp.*, 2018.

[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.

[3] A. Barabási, N. Gulbahce, and J. Loscalzo. Network medicine: a network-based approach to human disease. *Nature Reviews Genetics*, 12(56), 2011.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Jan. 2003.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge Univ. Press, Cambridge, UK, 2004.

[6] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imag. Vis.*, 40(1), 2011.

[7] J. Chang and D. M. Blei. Relational topic models for document networks. In *Proc. of the 12th Int. Conf. on Art. Int. Stat. (AISTATS)*, Florida, USA, 2009.

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

[9] 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.

[10] L. Condat. A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *Journal of Opt. Th. and App.*, 158(2):460–479, Aug. 2013.

[11] S. Cui, A. Hero, Z.-Q. Luo, and J.M.F. Moura, editors. *Big Data over Networks*. Cambridge Univ. Press, 2016.

[12] X. Dong, D. Thanou, M. Rabbat, and P. Frossard. Learning graphs from data: A signal representation perspective. *IEEE Signal Processing Magazine*, 2019(3), May 2019.

[13] C. Gentile, S. Li, and G. Zapella. Online clustering of bandits. In *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, volume 32, Beijing, China, 2014.

[14] D. Goldfarb and W. Yin. Parametric maximum flow algorithms for fast total variation minimization. *SIAM J. Sc. Comp.*, 2009.

[15] L. Grady. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11), 2006.

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

[17] T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity. The Lasso and its Generalizations*. CRC Press, 2015.

[18] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge Univ. Press, Cambridge, UK, 1985.

[19] J.-C. Hütter and P. Rigollet. Optimal rates for total variation denoising. In *Annual Conference on Learning Theory*, volume 49, Jun. 2016.

[20] A. Jung. Learning the conditional independence structure of stationary time series: A multitask learning approach. *IEEE Trans. Signal Processing*, 63(21), Nov. 2015.

[21] A. Jung. On the complexity of sparse label propagation. *Front. Appl. Math. Stat.*, 4:22, July 2018.

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

[23] A. Jung and N. Tran. Localized linear regression in networked data. *IEEE Sig. Proc. Letters*, 26(7), July 2019.

[24] A. Jung and N. Vesselinova. Analysis of network lasso for semi-supervised regression. In *The 22nd Int. Conf. Art. Int. Stat. (AISTATS)*, Okinawa, Japan, April 2019.

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

[26] B. J. Lengerich, B. Aragam, and E. P. Xing. Personalized regression enables samples-specific pan-cancer analysis. *Bioinformatics*, 34, 2018.

[27] S. Li, A. Karatzoglou, and C. Gentile. Collaborative filtering bandits. In *Proc. of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, Pisa, Italy, Jul. 2016.

[28] T. Li, E. Levina, and J. Zhu. Prediction models for network-linked data. *Ann. App. Stat.*, 2019.

[29] A. Mokhtari and A. Ribeiro. Global convergence of online limited memory bfgs. *Jour. Mach. Learning Res.*, 16:3151 – 3181, Dec. 2015.

[30] B. Nadler, N. Srebro, and X. Zhou. Statistical analysis of semi-supervised learning: The limit of infinite unlabelled data. In *Advances in Neural Information Processing Systems 22*, pages 1330–1338. 2009.

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

[32] R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. I. Jordan. A general analysis of the convergence of admm. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, Lille, France, 2015.

[33] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013.

[34] T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *IEEE ICCV*, Barcelona, Spain, Nov. 2011.

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

[36] C. Rother, V. Kolmogorov, and A. Blake. "Grabcut" - interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (SIGGRAPH)*, 2004.

[37] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, New York, 3 edition, 1976.

[38] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, Dec. 2007.

[39] M. J. Wainwright and M. I. Jordan. *Graphical Models, Exponential Families, and Variational Inference*, volume 1 of *Foundations and Trends in Machine Learning*. Now Publishers, Hanover, MA, 2008.

[40] W. W. Zachary. An information flow model for conflict and fission in small groups. *J. Anthro. Res.*, 33(4), 1977.

[41] P. Zhang, C. Moore, and L. Zdeborová. Phase transitions in semisupervised clustering of sparse networks. *Phys. Rev. E*, 90:052802, Nov 2014.

10. PROOFS

We first collect some helper results in Sec. 10.1 that will be used in Sec. 10.2 to obtain a detailed derivation of Theorem 1.

10.1. Helper Results

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

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

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 [38])

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

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 $\mathcal{K}(\mathbf{L}) \subseteq \mathcal{W}$ is $\mathbf{I}-\mathbf{P}$. Then (see [19]),

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

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

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

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

$$\sum_{i \in \mathcal{V}} \big(\mathbf{u}^{(i)}\big)^T \mathbf{v}^{(i)} \leq (1/|\mathcal{V}|) \sum_{i,j \in \mathcal{V}} \big(\mathbf{u}^{(i)}\big)^T \mathbf{v}^{(j)} + \big\|\big(\mathbf{D}^{\dagger}\big)^T \mathbf{v}\big\|_{2,\infty} \|\mathbf{D}\mathbf{u}\|_{2,1}. \tag{52}$$

The result (45) follows from (52) by using (15). □

Applying Lem. 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 $\mathbf{u},\mathbf{v} \in \mathcal{W}$,*

$$\sum_{i \in \mathcal{M}} \big(\mathbf{v}^{(i)}\big)^T \mathbf{u}^{(i)} \leq \max_{l=1,\ldots,|\mathcal{P}|} (1/|\mathcal{C}_l|) \big\| \sum_{i \in \mathcal{C}_l} \mathbf{v}^{(i)} \big\|_2 \sum_{j \in \mathcal{M}} \|\mathbf{u}^{(j)}\|_2 + \max_{l=1,\ldots,|\mathcal{P}|} \big\|\big(\mathbf{D}_{\mathcal{C}_l}^{\dagger}\big)^T \mathbf{v}_{\mathcal{C}_j}\big\|_{2,\infty} \|\mathbf{u}\|_{\mathrm{TV}}. \tag{53}$$

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 10.2) 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 $\mu^{(i)} := \mathrm{E}\{y^{(i)}\}$. If $\nabla^2\Phi^{(i)} \preceq U\mathbf{I}$ for all $i \in \mathcal{M}$, then*

$$\mathrm{P}\big\{\big|(1/M)\sum_{i \in \mathcal{M}} \big(y^{(i)} - \mu^{(i)}\big)\big| \geq \eta\big\} \leq 2\exp\big(-M\eta^2/(2U)\big). \tag{54}$$

Proof. Set

$$y := \sum_{i \in \mathcal{M}} y^{(i)}, \text{ and } \mu := \sum_{i \in \mathcal{M}} \mu^{(i)}. \tag{55}$$

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

$$\begin{aligned} \mathrm{P}\big\{(1/M)\sum_{i \in \mathcal{M}} \big(y^{(i)} - \mu^{(i)}\big) \geq \eta\big\} &= \mathrm{P}\{y - \mu \geq M\eta\} \\ &= \mathrm{P}\{\exp(\theta y) \geq \exp(\theta(M\eta + \mu))\} \\ &\leq \exp(-\theta(M\eta+\mu))\mathrm{E}\{\exp(\theta y)\} \\ &= \exp(-\theta(M\eta+\mu)) \prod_{i \in \mathcal{M}} \mathrm{E}\{\exp(\theta y^{(i)})\}. \end{aligned} \tag{56}$$

The last equality in (56) is due to the independence of the random variables $y^{(i)}$.

Combining (56) with

$$\mathrm{E}\{\exp(\theta y^{(i)})\} \overset{(5)}{=} \exp(\Phi^{(i)}(\overline{\mathbf{w}}^{(i)} + \theta\mathbf{m}^{(i)}) - \Phi^{(i)}(\overline{\mathbf{w}}^{(i)})) \tag{57}$$

yields

$$\mathrm{P}\{y - \mu \geq \eta\} \leq \exp(-\theta(M\eta+\mu) + \sum_{i \in \mathcal{M}} \Phi^{(i)}(\overline{\mathbf{w}}^{(i)} + \theta\mathbf{m}^{(i)}) - \Phi^{(i)}(\overline{\mathbf{w}}^{(i)})). \tag{58}$$

Similarly,

$$\mathrm{P}\{y - \mu \leq -\eta\} \leq \exp(-\theta(M\eta+\mu) + \sum_{i \in \mathcal{M}} \Phi^{(i)}(\overline{\mathbf{w}}^{(i)} + \theta\mathbf{m}^{(i)}) - \Phi^{(i)}(\overline{\mathbf{w}}^{(i)})). \tag{59}$$

A union bound allows to sum up (58) and (60) to obtain

$$\mathrm{P}\{|y - \mu| \geq \eta\} \leq 2\exp(-\theta(M\eta+\mu) + \sum_{i \in \mathcal{M}} \Phi^{(i)}(\overline{\mathbf{w}}^{(i)} + \theta\mathbf{m}^{(i)}) - \Phi^{(i)}(\overline{\mathbf{w}}^{(i)})). \tag{60}$$

Using Taylor's theorem and $\nabla\Phi^{(i)}(\overline{\mathbf{w}}^{(i)}) = \mathrm{E}\{\mathbf{t}^{(i)}\}$ [39],

$$\Phi^{(i)}(\overline{\mathbf{w}}^{(i)} + \theta\mathbf{m}^{(i)}) - \Phi^{(i)}(\overline{\mathbf{w}}^{(i)}) = \theta\mu^{(i)} + (1/2)\big(\theta^{(i)}\big)^2 \big(\mathbf{m}^{(i)}\big)^T \nabla^2\Phi^{(i)}(\overline{\mathbf{w}}^{(i)} + \theta^{(i)}\mathbf{m}^{(i)})\mathbf{m}^{(i)} \tag{61}$$

with some $\theta^{(i)} \in [0,\theta]$. Inserting $\nabla^2\Phi^{(i)} \preceq U\mathbf{I}$ into (61),

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

and, in turn via (60),

$$\mathrm{P}\{|y - \mu| \geq \eta\} \leq \exp(-\theta M\eta + M\theta^2 U/2). \tag{62}$$

Optimizing (62) by choosing θ suitably yields (54). □

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 M independent random vectors $\mathbf{z}^{(i)}$, 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*

$$\mathrm{P}\{\big\|(1/M)\sum_{i\in\mathcal{M}}\big(\mathbf{z}^{(i)} - \mathrm{E}\{\mathbf{z}^{(i)}\}\big)\big\| \geq \eta\} \leq 2\exp\big(-M\eta^2/(2dU)\big). \quad (63)$$

10.2. Proof of Theorem 1

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

$$\sum_{i\in\mathcal{M}}\big[\Phi^{(i)}(\widehat{\mathbf{w}}^{(i)}) - (\widehat{\mathbf{w}}^{(i)})^T\mathbf{t}^{(i)}\big] + M\lambda\|\widehat{\mathbf{w}}\|_{\mathrm{TV}} \leq \sum_{i\in\mathcal{M}}\big[\Phi^{(i)}(\overline{\mathbf{w}}^{(i)}) - (\overline{\mathbf{w}}^{(i)})^T\mathbf{t}^{(i)}\big] + M\lambda\|\overline{\mathbf{w}}\|_{\mathrm{TV}}. \quad (64)$$

We can rewrite (64) as

$$\sum_{i\in\mathcal{M}}\big(\boldsymbol{\varepsilon}^{(i)}\big)^T\widehat{\mathbf{w}}^{(i)} - \big(\bar{\mathbf{t}}^{(i)}\big)^T\widehat{\mathbf{w}}^{(i)} + \Phi^{(i)}(\widehat{\mathbf{w}}^{(i)}) + \lambda\|\widehat{\mathbf{w}}\|_{\mathrm{TV}} \quad (65)$$

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

with $\bar{\mathbf{t}}^{(i)} := \mathrm{E}\big\{\mathbf{t}^{(i)}\big\}$ and "observation noise" $\boldsymbol{\varepsilon}^{(i)} := \bar{\mathbf{t}}^{(i)} - \mathbf{t}^{(i)}$. To further develop (65), we make use of

$$\operatorname*{arg\,min}_{\mathbf{w}\in\mathbb{R}^d} -\mathbf{w}^T\bar{\mathbf{t}}^{(i)} + \Phi^{(i)}(\mathbf{w}) = \overline{\mathbf{w}}^{(i)}, \quad (66)$$

with the true weight vector $\overline{\mathbf{w}}^{(i)}$ underlying (2). The identity (66) can be verified by the zero-gradient condition and evaluating the gradient of $\Phi^{(i)}(\mathbf{w})$ (see [39, Proposition 3.1.]). Combining (65) with (66),

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

with 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

$$\big|(1/M)\sum_{i\in\mathcal{M}}\big(\boldsymbol{\varepsilon}^{(i)}\big)^T\widetilde{\mathbf{w}}^{(i)}\big| \leq \lambda\kappa\|\widetilde{\mathbf{w}}\|_{\mathcal{M}} + (\lambda/2)\|\widetilde{\mathbf{w}}\|_{\mathrm{TV}} \quad (68)$$

for every $\widetilde{\mathbf{w}} \in \mathcal{W}$. Here, we used $\kappa := \frac{K+1}{L-1}$ and

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

Combining (68) with (67),

$$\|\widehat{\mathbf{w}}\|_{\mathrm{TV}} \leq (1/2)\|\widetilde{\mathbf{w}}\|_{\mathrm{TV}} + \|\overline{\mathbf{w}}\|_{\mathrm{TV}} + \kappa\|\widetilde{\mathbf{w}}\|_{\mathcal{M}}, \quad (69)$$

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

$$\begin{aligned}\|\widehat{\mathbf{w}}\|_{\mathcal{E}\setminus\partial\mathcal{P}} &\leq (1/2)\|\widetilde{\mathbf{w}}\|_{\mathrm{TV}} + \|\overline{\mathbf{w}}\|_{\mathrm{TV}} - \|\widehat{\mathbf{w}}\|_{\partial\mathcal{P}} + \kappa\|\widetilde{\mathbf{w}}\|_{\mathcal{M}} \\ &\overset{(a)}{\leq} (1/2)\|\widetilde{\mathbf{w}}\|_{\mathrm{TV}} + \|\overline{\mathbf{w}}\|_{\partial\mathcal{P}} - \|\widehat{\mathbf{w}}\|_{\partial\mathcal{P}} + \kappa\|\widetilde{\mathbf{w}}\|_{\mathcal{M}} \\ &\overset{(b)}{\leq} (1/2)\|\widetilde{\mathbf{w}}\|_{\mathrm{TV}} + \|\overline{\mathbf{w}} - \widehat{\mathbf{w}}\|_{\partial\mathcal{P}} + \kappa\|\widetilde{\mathbf{w}}\|_{\mathcal{M}}, \end{aligned} \quad (70)$$

where 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 (70) as

$$(1/2)\|\widetilde{\mathbf{w}}\|_{\mathcal{E}\setminus\partial\mathcal{P}} \leq (3/2)\|\widetilde{\mathbf{w}}\|_{\partial\mathcal{P}} + \kappa\|\widetilde{\mathbf{w}}\|_{\mathcal{M}} \overset{\kappa<1}{<} (3/2)\|\widetilde{\mathbf{w}}\|_{\partial\mathcal{P}} + \|\widetilde{\mathbf{w}}\|_{\mathcal{M}}. \quad (71)$$

Thus, for sufficiently small observation noise (such that (68) 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. To this end, we apply the triangle inequality for TV to (65) yielding

$$\sum_{i\in\mathcal{M}}\big(\boldsymbol{\varepsilon}^{(i)}\big)^T\widetilde{\mathbf{w}}^{(i)} - \big(\bar{\mathbf{x}}^{(i)}\big)^T\widehat{\mathbf{w}}^{(i)} + \Phi^{(i)}(\widehat{\mathbf{w}}^{(i)}) \leq \sum_{i\in\mathcal{M}} -\big(\bar{\mathbf{x}}^{(i)}\big)^T\overline{\mathbf{w}}^{(i)} + \Phi^{(i)}(\overline{\mathbf{w}}^{(i)}) + M\lambda\|\widetilde{\mathbf{w}}\|_{\mathrm{TV}}. \quad (72)$$

Using Taylor's theorem and Asspt. 2,

$$\Phi^{(i)}(\widehat{\mathbf{w}}^{(i)}) - \Phi^{(i)}(\overline{\mathbf{w}}^{(i)}) - \big(\bar{\mathbf{x}}^{(i)}\big)^T\big(\widehat{\mathbf{w}}^{(i)} - \overline{\mathbf{w}}^{(i)}\big) \geq L\|\widetilde{\mathbf{w}}^{(i)}\|_2^2. \quad (73)$$

Inserting (73) into (72),

$$(1/M)\sum_{i\in\mathcal{M}}\big[-\big(\boldsymbol{\varepsilon}^{(i)}\big)^T\widetilde{\mathbf{w}}^{(i)} + L\|\widetilde{\mathbf{w}}^{(i)}\|_2^2\big] \leq \lambda\|\widetilde{\mathbf{w}}\|_{\partial\mathcal{P}}. \quad (74)$$

Combining (68) with (74),

$$L\|\widetilde{\mathbf{w}}\|_{\mathcal{M}}^2 \leq \lambda\|\widetilde{\mathbf{w}}\|_{\partial\mathcal{P}} + \kappa\lambda\|\widetilde{\mathbf{w}}\|_{\mathcal{M}}. \quad (75)$$

Combining (71) with (21) yields

$$\|\widetilde{\mathbf{w}}\|_{\partial\mathcal{P}} \leq \kappa\|\widetilde{\mathbf{w}}\|_{\mathcal{M}} \quad (76)$$

and, in turn via (75),

$$\|\widetilde{\mathbf{w}}\|_{\mathcal{M}} \leq 2\lambda\kappa/L. \quad (77)$$

Inserting (77) into (76) and (71),

$$\begin{aligned}\|\widetilde{\mathbf{w}}\|_{\mathrm{TV}} &= \|\widetilde{\mathbf{w}}\|_{\partial\mathcal{P}} + \|\widetilde{\mathbf{w}}\|_{\mathcal{E}\setminus\partial\mathcal{P}} \\ &\overset{(71)}{\leq} \|\widetilde{\mathbf{w}}\|_{\partial\mathcal{P}} + 3\|\widetilde{\mathbf{w}}\|_{\partial\mathcal{P}} + \kappa\|\widetilde{\mathbf{w}}\|_{\mathcal{M}} \\ &\overset{(76)}{\leq} 5\kappa\|\widetilde{\mathbf{w}}\|_{\mathcal{M}} \\ &\overset{(77)}{\leq} 5\lambda\kappa^2/L. \end{aligned} \quad (78)$$

According to (78), we can ensure a prescribed error level $\|\widetilde{\mathbf{w}}\|_{\mathrm{TV}} \leq \eta$ by setting ($L > 1$)

$$\lambda := \eta/(5\kappa^2). \quad (79)$$

The final step of the proof is to control the probability of (68) to hold. By Cor. 3, (68) holds if

$$\max_{\mathcal{C}_l \in \mathcal{P}} (1/|\mathcal{C}_l|) \| \sum_{i \in \mathcal{C}_l} \boldsymbol{\varepsilon}_i \|_2 \leq (\lambda/2)\kappa, \tag{80}$$

and simultaneously

$$\max_{\mathcal{C}_l \in \mathcal{P}} \| \left(\mathbf{D}^{\dagger}_{\mathcal{C}_l}\right)^T \boldsymbol{\varepsilon}_{\mathcal{C}_l} \|_{2,\infty} \leq M\lambda/4. \tag{81}$$

We first bound the probability that (80) fails to hold. For a particular cluster $\mathcal{C}_l$, (63) yields

$$\mathrm{P}\{(1/|\mathcal{C}_l|) \| \sum_{i \in \mathcal{C}_l} \boldsymbol{\varepsilon}_i \|_2 \leq (\lambda/2)\kappa\} \leq 2 \exp\left(-\frac{|\mathcal{C}_l|\lambda^2\kappa^2}{8dU}\right). \tag{82}$$

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

$$\mathrm{P}\{\text{“(80) invalid”}\} \leq 2|\mathcal{P}| \max_{l=1,\ldots,|\mathcal{P}|} \exp\left(-\frac{|\mathcal{C}_l|\lambda^2\kappa^2}{8dU}\right). \tag{83}$$

For controlling the probability of (81) failing to hold, we combine (18) with Lem. 4. This yields, using a union bound over all edges $e \in \mathcal{E}$,

$$\mathrm{P}\{\text{“(81) invalid”}\} \leq 2|\mathcal{E}| \exp\left(-\frac{M\rho^2_{\mathcal{P}}\lambda^2}{64Ud\|\mathbf{A}\|^2_{\infty}}\right). \tag{84}$$

A union bound yields (23) by summing the bounds (83) and (84) for the choice (79).