# LEARNING NETWORKED EXPONENTIAL FAMILIES WITH NETWORK LASSO

*Alexander Jung*

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

## ABSTRACT

The data arising in many important big-data applications, ranging from social networks to network medicine, consist of high-dimensional data points related by an intrinsic (complex) network structure. In order to jointly leverage the information conveyed in the network structure as well as the statistical power contained in high-dimensional data points, we propose networked exponential families. We apply the network Lasso to learn networked exponential families as a probabilistic model for heterogeneous datasets with intrinsic network structure. In order to allow for accurate learning from high-dimensional data we borrow statistical strength, via the intrinsic network structure, across the dataset. The resulting method aims at regularized empirical risk minimization using the total variation of the model parameters as regularizer. This minimization problem is a non-smooth convex optimization problem which we solve using a primal-dual splitting method. This method is appealing for big data applications as it can be implemented as a highly a scalable message passing algorithm.

***Index Terms*—** Lasso, big data over networks, exponential families, semi-supervised machine learning, transfer learning, complex networks, convex optimization, primal-dual method

## 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 [1]–[3]. Network science provides powerful tools for the analysis of such data based on its intrinsic network structure [4], [5]. However, the network structure of datasets is complemented by the information contained in attributes (such as features or labels) of individual data points [1].

In this paper, we study a particular class of statistical models for networked data which are based on modelling the statistics of data attributes using exponential families [6]–[8]. The exponential families describing the individual data points are coupled via the network structure underlying the data. The resulting networked exponential families allows to jointly captialize on the network structure and the statistical properties of features and labels assigned to individual data points.

Our approach extends prior work on networked linear and logistic regression models [9]–[11]. Indeed, the proposed network exponential family model contains linear and logistic regression as special cases. In contrast to [1], which formulates a probabilistic model for the network structure, we consider the network structure as fixed and known.

The main contribution of this paper is the application of the network Lasso (nLasso) to learning networked exponential families (see Figure 1). The nLasso has been proposed recently as a natural extension of the *least absolute shrinkage and selection operator* (Lasso) to networked data [12], [13]. 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:** Our main contributions are: (i) We present a novel implementation of logistic network Lasso by applying a primal-dual method. This method can be implemented as highly scalable message passing on the network structure underlying the data. (ii) We prove the convergence of this primal-dual method.

**Notation:** Boldface lowercase (uppercase) letters denote vectors (matrices). We denote $\mathbf{x}^T$ the transpose of vector $\mathbf{x}$. The $\ell_2$-norm of a vector $\mathbf{x}$ is $\|\mathbf{x}\| = \sqrt{\mathbf{x}^T\mathbf{x}}$. The convex conjugate of a function $f$ is defined as $f^*(\mathbf{y}) = \sup_{\mathbf{x}}(\mathbf{y}^T\mathbf{x} - f(\mathbf{x}))$.

## 2. NETWORKED EXPONENTIAL FAMILIES

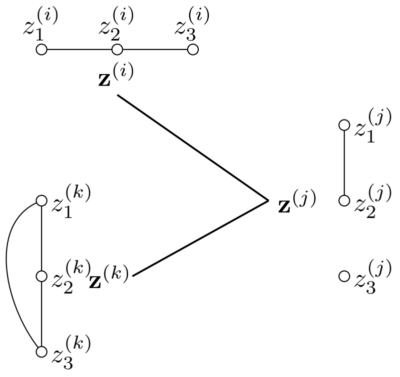

**Fig. 1**: A networked exponential family is a probabilistic model for (high-dimensional) data points $\mathbf{z}^{(i)}$ related by some intrinsic network structure. The data points $\mathbf{z}^{(i)}$ might represent scientific articles within a collection of articles that are related via joint authors. Another application for such models is learning analytics where data points $\mathbf{z}^{(i)}$ represent the digital traces left by students within learning management systems. Network medicine considers networks of proteins which are described in large databases such as https://www.uniprot.org.

We consider networked data that is represented by an undirected weighted graph (the "empirical graph") $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$. A particular node $i \in \mathcal{V} = \{1, \ldots, N\}$ of the graph represents an individual data point (such as a document, or a social network user profile).[1] Two different data points $i, j \in \mathcal{V}$ are connected by an undirected edge $\{i, j\} \in \mathcal{E}$ if they are considered similar (such as documents authored by the same person or social network profiles of befriended users). For ease of notation, we denote the edge set $\mathcal{E}$ by $\{1, \ldots, E := |\mathcal{E}|\}$.

Each edge $e = \{i, j\} \in \mathcal{E}$ is endowed with a positive weight $A_e = A_{ij} > 0$ which quantifies the amount of similarity between data points $i, j \in \mathcal{V}$. The neighborhood of a node $i \in \mathcal{V}$ is $\mathcal{N}(i) := \{j : \{i, j\} \in \mathcal{E}\}$.

Beside the network structure, datasets convey additional information in the form of attributes $\mathbf{z}^{(i)} \in \mathbb{R}^d$ associated with each data point $i \in \mathcal{V}$. For a document collection with data points representing text documents, the attributes could be frequencies of particular words [1].

We model the attributes $\mathbf{z}^{(i)}$ of data points $i \in \mathcal{V}$ as independent random variables distributed according to an exponential family [7]

$$p(\mathbf{z}; \mathbf{w}^{(i)}) := b^{(i)}(\mathbf{z}) \exp\big((\mathbf{w}^{(i)})^T \mathbf{t}^{(i)}(\mathbf{z}) - \Phi(\mathbf{w}^{(i)})\big). \quad (1)$$

Note that the distribution (1) is parametrized by the (unknown) weight vectors $\mathbf{w}^{(i)}$. The networked exponential family combines the node-wise models (1) by requiring the weight vectors to be similar for well-connected data points. In particular, we require the weight vectors to have small total variation (TV)

$$\|\mathbf{w}\|_{\mathrm{TV}} := \sum_{\{i,j\} \in \mathcal{E}} A_{ij} \|\mathbf{w}^{(j)} - \mathbf{w}^{(i)}\|. \quad (2)$$

Requiring small TV of weight vectors $\mathbf{w}^{(i)}$, for $i \in \mathcal{V}$, typically implies that weight vectors are approximately constant over well connected subsets (clusters) of nodes.

## 3. SOME EXAMPLES

Before we turn to the problem of learning networked exponential families (1) in Section 4, we now discuss some important special cases of the model (1).

### 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

$$y^{(i)} = (\mathbf{x}^{(i)})^T \mathbf{w}^{(i)} + \varepsilon^{(i)}, \quad (3)$$

with Gaussian noise $\varepsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$ of known variance $\sigma^2$. The linear model (3), for each node $i \in \mathcal{V}$, is parametrized by the weight vector $\mathbf{w}^{(i)}$. The networked linear regression model requires the weight vectors in the individual linear models (3) to have a small TV (2) [9], [10].

The model (3) is obtained as the special case of the exponential family (1) for the scalar attributes $z^{(i)} := y^{(i)}$ with $\mathbf{t}^{(i)}(z) = (z/\sigma^2)\mathbf{x}^{(i)}$ and $\Phi(\mathbf{w}) = (1/(2\sigma^2))\|\mathbf{w}\|^2$.

### 3.2. Networked Logistic Regression

Consider a networked dataset whose data points $i \in \mathcal{V}$ are characterized by features $\mathbf{x}^{(i)} \in \mathbb{R}^d$ and binary labels $y^{(i)} \in \{-1, 1\}$. Maybe the most basic (yet quite useful) model for the relation between features and labels is the linear model

$$p(y^{(i)} = 1) := 1/(1 + \exp(-(\mathbf{x}^{(i)})^T \mathbf{w}^{(i)})). \quad (4)$$

The logistic regression model (4) is parametrized by the weight vector $\mathbf{w}^{(i)}$ for each node $i \in \mathcal{V}$. The networked logistic regression model requires the weight vectors in the node-wise logistic regression models (4) to have a small TV (2) [14].

We obtain the logistic regression model (4) as the special case of the exponential family (1) for the scalar attributes $z^{(i)} := y^{(i)}$ with $\mathbf{t}^{(i)}(z) = \mathbf{x}^{(i)} z$ and $\Phi(\mathbf{w}) = 1/\big(\exp\big((\mathbf{x}^{(i)})^T \mathbf{w}^{(i)}\big) + \exp\big(-(\mathbf{x}^{(i)})^T \mathbf{w}^{(i)}\big)\big)$.

### 3.3. Networked Latent Dirichlet Allocation

Consider a networked dataset representing a collection of text documents (such as scientific articles). The latent Dirichlet allocation model (LDA) is a probabilistic model for the relative frequencies of words in a document [7], [15]. 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 $z^{(i)}_{w,1}, \ldots, z^{(i)}_{w,N} \in \{1, \ldots, W\}$ and $z^{(i)}_{t,1}, \ldots, z^{(i)}_{t,N} \in \{1, \ldots, T\}$ 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 (1) with particular choices for $\mathbf{t}(\cdot)$ and $\Phi(\cdot)$ (see [7], [15]).

## 4. NETWORK LASSO

Our goal is to develop a method for learning an accurate estimate $\widehat{\mathbf{w}}^{(i)}$ for the weight vectors $\mathbf{w}^{(i)}$ at $i = 1, \ldots, N$. The learning of the weight vectors $\mathbf{w}^{(i)}$ is based on the availability of the nodes attributes $\mathbf{z} = \mathbf{z}^{(i)}$ for a small sampling 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 $\mathbf{z}^{(i)}$:

$$\begin{aligned} p\big(\{\mathbf{z}^{(i)}\}_{i \in \mathcal{M}}\big) &= \prod_{i=1}^{M} p(\mathbf{z}^{(i)}; \mathbf{w}^{(i)}) \\ &\overset{(1)}{=} \prod_{i=1}^{M} b^{(i)}(\mathbf{z}^{(i)}) \exp\big((\mathbf{z}^{(i)})^T \mathbf{w}^{(i)} - \Phi(\mathbf{w}^{(i)})\big). \quad (5) \end{aligned}$$

It is easy to show that maximizing (6) is equivalent to minimizing the empirical risk

$$\widehat{E}(\mathbf{w}) := (1/M) \sum_{i \in \mathcal{M}} -\big(\mathbf{z}^{(i)}\big)^T \mathbf{w}^{(i)} + \Phi(\mathbf{w}^{(i)}). \quad (6)$$

[1] With a slight abuse of notation, we refer by $i \in \mathcal{V}$ to a node of the empirical graph as well as the data point which is represented by that node.

The criterion (6) by itself is not sufficient for learning of the weights $\mathbf{w}$ since (6) since it completely ignores the weights $\widehat{\mathbf{w}}^{(i)}$ at unobserfved nodes $i \in \mathcal{V} \setminus \mathcal{M}$. Therefore, we need to impose some additional structure on the weight vectors. In particular, any reasonable estimate $\widehat{\mathbf{w}}$ should conform with the *cluster structure* of the empirical graph $\mathcal{G}$ [4].

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

$$\widehat{\mathbf{w}} \in \operatorname*{argmin}_{\mathbf{w} \in \mathcal{C}} \widehat{E}(\mathbf{w}) + \lambda \|\mathbf{w}\|_{\mathrm{TV}}. \tag{7}$$

The learning problem (7) is an instance of the generic nLasso problem [12]. The parameter $\lambda$ in (7) allows to trade-off small TV $\|\widehat{\mathbf{w}}\|_{\mathrm{TV}}$ against small error $\widehat{E}(\widehat{\mathbf{w}})$ (cf. (6)). The choice of $\lambda$ can be guided by cross validation [16].

It will be convenient to reformulate (7) using vector notation. We represent a graph signal $\mathbf{w} \in \mathcal{C}$ as the vector

$$\mathbf{w} = ((\mathbf{w}^{(1)})^T, \ldots, (\mathbf{w}^{(N)})^T)^T \in \mathbb{R}^{dN}. \tag{8}$$

Define a partitioned matrix $\mathbf{D} \in \mathbb{R}^{(dE) \times (dN)}$ block-wise as

$$\mathbf{D}_{e,i} = \begin{cases} A_{ij}\mathbf{I}_d & e = \{i,j\}, i < j \\ -A_{ij}\mathbf{I}_d & e = \{i,j\}, i > j \\ \mathbf{0} & \text{otherwise}, \end{cases} \tag{9}$$

where $\mathbf{I}_d \in \mathbb{R}^{d \times d}$ is the identity matrix. The term $A_{ij}(\mathbf{w}^{(i)} - \mathbf{w}^{(j)})$ in (2) is the $e$-th block of $\mathbf{D}\mathbf{w}$. Using (8) and (9), we can reformulate the lnLasso (7) as

$$\widehat{\mathbf{w}} \in \operatorname*{argmin}_{\mathbf{w} \in \mathbb{R}^{dN}} h(\mathbf{w}) + g(\mathbf{D}\mathbf{w}), \tag{10}$$

with

$$h(\mathbf{w}) = \widehat{E}(\mathbf{w}) \text{ and } g(\mathbf{u}) := \lambda \sum_{e=1}^{E} \|\mathbf{u}^{(e)}\| \tag{11}$$

with stacked vector $\mathbf{u} = (\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(E)})) \in \mathbb{R}^{dE}$.

## 5. PRIMAL-DUAL METHOD

The nLasso (10) is a convex optimization problem with a non-smooth objective function which rules out the use of gradient descent methods [17]. 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 [18], [19] for solving (10).

One particular example of a proximal method is the alternating direction method of multipliers (ADMM) which has been considered in [12]. However, we will choose another proximal method which is based on a dual problem to (10). Based on this dual problem, efficient primal-dual methods have been proposed recently [20], [21]. These methods are attractive since their analysis provides natural choices for the algorithm parameters. In contrast, tuning the ADMM parameters is non-trivial [22].

The preconditioned primal-dual method [20] launches from reformulating the problem (10) as a saddle-point problem

$$\min_{\mathbf{w} \in \mathbb{R}^{dN}} \max_{\mathbf{u} \in \mathbb{R}^{dE}} \mathbf{u}^T \mathbf{D}\mathbf{w} + h(\mathbf{w}) - g^*(\mathbf{u}), \tag{12}$$

with the convex conjugate $g^*$ of $g$ [21].

The solutions $(\widehat{\mathbf{w}}, \widehat{\mathbf{u}})$ of (12) are characterized by [23, Thm 31.3]

$$\begin{aligned} -\mathbf{D}^T\widehat{\mathbf{u}} &\in \partial h(\widehat{\mathbf{w}}) \\ \mathbf{D}\widehat{\mathbf{w}} &\in \partial g^*(\widehat{\mathbf{u}}). \end{aligned} \tag{13}$$

This condition is, in turn, equivalent to

$$\begin{aligned} \widehat{\mathbf{w}} - \mathbf{T}\mathbf{D}^T\widehat{\mathbf{u}} &\in (\mathbf{I}_{dN} + \mathbf{T}\partial h)(\widehat{\mathbf{w}}), \\ \widehat{\mathbf{u}} + \mathbf{\Sigma}\mathbf{D}\widehat{\mathbf{w}} &\in (\mathbf{I}_{dE} + \mathbf{\Sigma}\partial g^*)(\widehat{\mathbf{u}}), \end{aligned} \tag{14}$$

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 (14) lends naturally to the following coupled fixed point iterations [20]

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

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

If the matrices $\mathbf{\Sigma}$ and $\mathbf{T}$ in (15), (16) satisfy

$$\|\mathbf{\Sigma}^{1/2}\mathbf{D}\mathbf{T}^{1/2}\|^2 < 1, \tag{17}$$

the sequences obtained from iterating (15) and (16) converge to a saddle point of the problem (12) [20, Thm. 1]. The condition (17) is satisfied for the choice $\mathbf{\Sigma} = \{(1/(2A_e))\mathbf{I}_d\}_{e \in \mathcal{E}}$ and $\{(\tau/d^{(i)})\mathbf{I}_d\}_{i \in \mathcal{V}}$, with node degree $d^{(i)} = \sum_{j \neq i} A_{ij}$ and some $\tau < 1$ [20, Lem. 2].

The update (16) involves the resolvent operator

$$(\mathbf{I} + \mathbf{\Sigma}\partial g^*)^{-1}(\mathbf{v}) = \operatorname*{argmin}_{\mathbf{v}' \in \mathbb{R}^{dE}} g^*(\mathbf{v}') + (1/2)\|\mathbf{v}' - \mathbf{v}\|^2_{\mathbf{\Sigma}^{-1}}, \tag{18}$$

where $\|\mathbf{v}\|_{\mathbf{\Sigma}} := \sqrt{\mathbf{v}^T \Sigma \mathbf{v}}$. The convex conjugate $g^*$ of $g$ (see (11)) 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 $\ell_2$-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 [19, Sec. 6.5]

$$\begin{aligned} &((\mathbf{I}_{dE} + \mathbf{\Sigma}\partial g^*)^{-1}(\mathbf{v}))^{(e)} \\ &\overset{(18)}{=} \operatorname*{argmin}_{\mathbf{v}' \in \mathbb{R}^{d}} g_2^*(\mathbf{v}') + (1/(2\sigma^{(e)}))\|\mathbf{v}' - \mathbf{v}^{(e)}\|^2 \\ &= \mathbf{v}^{(e)} - \sigma^{(e)}(\mathbf{I}_d + (\lambda/\sigma^{(e)})\partial\|.\|)^{-1}(\mathbf{v}^{(e)}/\sigma^{(e)}) \\ &= \begin{cases} \lambda\mathbf{v}^{(e)}/\|\mathbf{v}^{(e)}\| & \text{if } \|\mathbf{v}^{(e)}\| > \lambda \\ \mathbf{v}^{(e)} & \text{otherwise}, \end{cases} \end{aligned}$$

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

The update (15) involves the resolvent operator $(\mathbf{I} + \mathbf{T}\partial h)^{-1}$ of $h$ (see (6) and (11)), which does not have a closed-form solution in general. Still, for the choice $\mathbf{T} = \operatorname{diag}\{\tau^{(i)}\mathbf{I}_d\}_{i=1}^{N}$, the update (15) decomposes into separate node-wise updates

$$\mathbf{w}^{(i)}_{k+1} := \operatorname*{argmin}_{\mathbf{w} \in \mathbb{R}^d} -\mathbf{w}^T\mathbf{z}^{(i)} + \Phi(\mathbf{w}) + \tilde{\tau}^{(i)}\|\mathbf{w} - \overline{\mathbf{w}}^{(i)}\|^2 \tag{19}$$

with $\overline{\mathbf{w}} := \mathbf{w}_k - \mathbf{T}\mathbf{D}^T\mathbf{u}_k$ and $\tilde{\tau}^{(i)} := M/(2\tau^{(i)})$.

In general, it is not possible to compute the update (19) exactly.[2] However, since (19) amounts to the unconstrained minimization of a smooth and convex objective function we can apply efficient convex optimization methods [24]. In fact, the update (19) is a regularized maximum likelihood problem for the exponential family (1). This can be solved efficiently using quasi-Newton methods such as L-BGFS [25], [26].

Let us denote the approximate solution to (19) by $\widehat{\mathbf{w}}^{(i)}_{k+1}$ and assume that it is sufficiently accurate such that

$$e_k = \|\widehat{\mathbf{w}}^{(i)}_{k+1} - \mathbf{w}^{(i)}_{k+1}\| \leq 1/k^2. \tag{20}$$

Thus, we require the approximation quality (for approximating the update (19)) to increase with the iteration number $k$. As can be verified using [27, Thm. 3.2], the error bound (20) ensures the sequences obtained by (15) and (16) when replacing the exact update (19) with the approximation $\widehat{\mathbf{w}}_{k+1}$ still converge to a saddle-point of (12) and, in turn, a solution of the nLasso problem (10).

The learning Algorithm 1 can be adapted easily to cope with partially observed exponential families [7]. In particular, for the networked LDA described in Section 3, we typically have access only to the word variables $z^{(i)}_{w,1}, \ldots, z^{(i)}_{w,N}$ of some documents $i \in \mathcal{M} \subseteq \mathcal{V}$. However, for (approximately) computing the update step (19) we would also need the values of the topic variables $z^{(i)}_{t,1}, \ldots, z^{(i)}_{t,N}$ but those are not observed (latent variables). In this case we can approximate (19) by some "Expectation-Maximization" method (see [7, Sec. 6.2])

**Algorithm 1** Learning Networked Exponential Family

```
Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, $\{\mathbf{z}^{(i)}\}_{i \in \mathcal{M}}$, $\mathcal{M}$, $\lambda$, $\mathbf{D}$
Initialize: $\mathbf{\Sigma} = \mathrm{diag}\{\sigma^{(e)} = 1/(2A_e)\mathbf{I}_d\}_{e=1}^{E}$, $\mathbf{T} = \mathrm{diag}\{\tau^{(i)} = (0.9/d^{(i)})\mathbf{I}_d\}_{i \in \mathcal{V}}$, $\beta_i = \tau^{(i)}/|\mathcal{M}|$ $k := 0$, $\widehat{\mathbf{w}}_0 := 0$, $\widehat{\mathbf{u}}_0 := 0$
repeat
    $\widehat{\mathbf{w}}_{k+1} := \widehat{\mathbf{w}}_k - \mathbf{T}\mathbf{D}^T\widehat{\mathbf{u}}_k$
    for each observed node $i \in \mathcal{M}$ do
        compute $\widehat{\mathbf{w}}^{(i)}_{k+1}$ by approximately solving (19)
    end for
    $\overline{\mathbf{u}} := \mathbf{u}_k + \mathbf{\Sigma}\mathbf{D}(2\widehat{\mathbf{w}}_{k+1} - \widehat{\mathbf{w}}_k)$
    $\widehat{\mathbf{u}}^{(e)}_{k+1} = \overline{\mathbf{u}}^{(e)} - \left(1 - \frac{\lambda}{\|\overline{\mathbf{u}}^{(e)}\|}\right)_{+} \overline{\mathbf{u}}^{(e)}$ for $e \in \mathcal{E}$
    $k := k+1$
until stopping criterion is satisfied
Output: $(\widehat{\mathbf{w}}_k, \widehat{\mathbf{u}}_k)$.
```

## 6. REFERENCES

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

[2] A. Barabási, N. Gulbahce, and J. Loscalzo, "Network medicine: a network-based approach to human disease," *Nature Reviews Genetics*, vol. 12, no. 56, 2010.

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

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

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

[6] L. D. Brown, *Fundamentals of Statistical Exponential Families*, Institute of Mathematical Statistics, Hayward, CA, 1986.

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

[8] A. Jung, S. Schmutzhard, and F. Hlawatsch, "The RKHS approach to minimum variance estimation revisited: Variance bounds, sufficient statistics, and exponential families," *IEEE Trans. Inf. Theory*, vol. 60, no. 7, pp. 4050–4065, Jul. 2014.

[9] A. Jung and N. Tran, "Localized linear regression in networked data," *submitted to a journal, arXiv preprint https://arxiv.org/abs/1903.11178*, 2019.

[10] M.Yamada, T. Koh, T. Iwata, J. Shawe-Taylor, and S. Kaski, "Localized Lasso for High-Dimensional Regression," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, USA, Apr. 2017, vol. 54, pp. 325–333, PMLR.

[11] H. Ambos, N. Tran, and A. Jung, "Classifying big data over networks via the logistic network lasso," in *Proc. 52nd Asilomar Conf. Signals, Systems, Computers*, Oct./Nov. 2018.

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

[13] Trevor Hastie, Robert Tibshirani, and Martin Wainwright, *Statistical Learning with Sparsity. The Lasso and its Generalizations*, CRC Press, 2015.

[14] H. Ambos, N. Tran, and A. Jung, "Classifying big data over networks via the logistic network lasso," in *Proc. 52nd Asilomar Conference on Signals, Systems, and Computers*. 2018, 10.1109/ACSSC.2018.8645260.

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

[16] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The Elements of Statistical Learning*, Springer Series in Statistics. Springer, New York, NY, USA, 2001.

[17] A. Jung, "A fixed-point of view on gradient methods for big data," *Frontiers in Applied Mathematics and Statistics*, vol. 3, 2017.

[18] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*,

[2] 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. For such special cases it is possible to solve (19) in closed-form (see [7, Sec. 6.1.1])

H. Bauschke, R. Burachik, P. Combettes, V. Elser, D. Luke, and H. Wolkowicz, Eds., vol. 49. Springer New York, 2011.

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

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

[21] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imag. Vis.*, vol. 40, no. 1, 2011.

[22] 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*, Lille, France, 2015, vol. 37.

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

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

[25] G. Andrew and J. Gao, "Scalable training of l1-regularized log-linear models," in *Proc. of the 24th International Conference on Machine Learning (ICML)*, Corvallis, OR, 2007.

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

[27] L. Condat, "A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms," *Journal of Optimization Theory and Applications*, vol. 158, no. 2, pp. 460–479, Aug. 2013.