

# ON THE DUALITY BETWEEN NETWORK FLOWS AND NETWORK LASSO

Alexander Jung

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

## ABSTRACT

Many applications generate data with an intrinsic network structure such as time series data, image data or social network data. The network Lasso (nLasso) has been proposed recently as a method for joint clustering and optimization of machine learning models for networked data. The nLasso extends the Lasso from sparse linear models to clustered graph signals. This paper explores the duality of nLasso and network flow optimization. We show that, in a very precise sense, nLasso is equivalent to a minimum-cost flow problem on the data network structure. Our main technical result is a concise characterization of nLasso solutions via existence of certain network flows. The main conceptual result is a useful link between nLasso methods and basic graph algorithms such as clustering or maximum flow.

## 1. INTRODUCTION

The network Lasso (nLasso) has been proposed recently to fit localized models to networked data [11]. Localized models allow to use different model parameters for different data nodes. However, the node-wise parameters are coupled by require them to have a small total variation (TV).

Efficient methods to process networked data are offered by graph algorithms such as clustering or network flow optimization [5]. While these graph algorithm only use the network structure, the nLasso also takes additional information into account [11]. We represent this additional information in the form of a graph signal which maps individual data points to a signal value (“label”).

We explore the duality between nLasso and a minimum-cost flow problem. This is a special case of the duality between structured norm minimization and network flow problems studied in [15]. In contrast to [15], we do not use this duality to apply network flow methods to solve nLasso but rather use the existence of certain network flows to characterize nLasso solutions.

Our analysis relates the performance of nLasso methods for joint optimization and clustering to existence of network flows which serve as a proxy measure for the connectivity of clusters. This is somewhat similar to the concept of conductivities used for the design and analysis of clustering methods in [26]. In contrast to [26], we study networked data point providing additional information in the form of a graph signal.

The main contributions of this paper are as follows.

- We show that the convex dual of nLasso is equivalent to a particular minimum-cost flow problem.
- We interpret a primal-dual method for nLasso as a distributed network flow optimization method.
- We characterize the solutions of nLasso via the existence of sufficient large network flows between cluster boundaries and sampled (labeled) nodes.
- We provide a novel interpretation of the nLasso parameter as a scaling of edge capacities in a flow network.

**Notation.** The sub-differential of a function  $g(\mathbf{x})$  at  $\mathbf{x}_0 \in \mathbb{R}^n$  is the set

$$\partial g(\mathbf{x}_0) := \{\mathbf{y} \in \mathbb{R}^n : g(\mathbf{x}) \geq g(\mathbf{x}_0) + \mathbf{y}^T(\mathbf{x} - \mathbf{x}_0) \text{ for any } \mathbf{x}\}.$$

The convex conjugate function of  $g(\mathbf{x})$  is [6]

$$g^*(\hat{\mathbf{y}}) := \sup_{\mathbf{y} \in \mathbb{R}^n} \mathbf{y}^T \hat{\mathbf{y}} - g(\mathbf{y}). \quad (1)$$

## 2. RECOVERING CLUSTERED GRAPH SIGNALS

We consider networked data whose network structure is encoded in an undirected empirical graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ . The nodes  $i \in \mathcal{V} = \{1, \dots, n\}$  of the empirical graph represented individual data points. Similar data points are connected by an edge  $\{i, j\} \in \mathcal{E}$  with some weight  $W_{i,j} > 0$  that quantifies the amount of similarity between  $i, j \in \mathcal{V}$ . We depict an example of an empirical graph  $\mathcal{G}$  in Fig. 1.

The neighbourhood  $\mathcal{N}(i)$  and degree  $d_i$  of a node  $i \in \mathcal{V}$  are defined, respectively, as

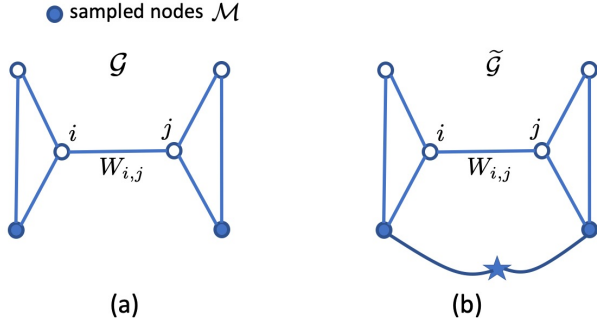
$$\mathcal{N}(i) := \{j \in \mathcal{V} : \{i, j\} \in \mathcal{E}\}, d_i := |\mathcal{N}(i)|. \quad (2)$$

For a given undirected empirical graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ , we orient the undirected edge  $\{i, j\}$  by defining the head as  $e^+ = \min\{i, j\}$  and the tail as  $e^- = \max\{i, j\}$ . Each undirected edge  $e = \{i, j\}$  is associated with the directed edge  $(e^+, e^-)$ .

We need the directed neighbourhoods  $\mathcal{N}^+(i) = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$ ,  $\mathcal{N}^-(i) = \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$  and the incidence matrix  $\mathbf{B} \in \mathbb{R}^{\mathcal{E} \times n}$ ,

$$B_{e,i} = 1 \text{ for } i = e^+, B_{e,i} = -1 \text{ for } i = e^-, B_{e,i} = 0 \text{ else.} \quad (3)$$

Beside the network structure, datasets carry additional information which we represent by a graph signal  $\mathbf{x} =$



**Fig. 1.** (a) Empirical graph of networked data including a sampling set of nodes whose signal values are observed. (b) Extended empirical graph obtained by adding the node  $\star$  and edges between sampled nodes  $i \in \mathcal{M}$  and  $\star$ .

$(x_1, \dots, x_n)^T \in \mathbb{R}^n$ . The graph signal values  $x_i$  might represent instantaneous amplitudes of an audio signal, the greyscale values of image pixels or the probabilities of social network members taking a particular action. We assume that signal values  $x_i$  are known only at few nodes  $i \in \mathcal{V}$  of a (small) sampling set  $\mathcal{M} \subseteq \mathcal{V}$ . Our goal is to recover the unknown signal values  $x_i$  for  $i \in \mathcal{V} \setminus \mathcal{M}$ .

To recover the signal values  $x_i$ , for  $i \in \mathcal{V}$  based on knowing them only on a (small) training set, we exploit the tendency of natural graph signals to be clustered.

**Assumption 1** (informal). *Nodes  $i, j \in \mathcal{V}$  within a well-connected subset (cluster) have similar signal values  $x_i \approx x_j$ .*

This clustering assumption is used in image processing to model images that are composed of few components within which the pixel colours are approximately constant [22]. The clustering assumption is also used in social sciences where signal values represent certain features of individuals that are similar within well-connected groups (clusters) [19]. More broadly, (variants) of the clustering assumption motivate semi-supervising learning methods [9].

To make the informal Asspt. 1 precise we need a measure for how well a graph signal conforms to the cluster structure of the empirical graph  $\mathcal{G}$ . We measure this “clusteredness” of a graph signal  $\mathbf{x}$  using the weighted TV [22, 25]

$$\|\mathbf{x}\|_{\text{TV}} := \sum_{\{i,j\} \in \mathcal{E}} W_{i,j} |x_j - x_i|. \quad (4)$$

Signal recovery methods based on TV minimization (4) turn out to be attractive statistically and computationally. These methods allow to recover clustered graph signals from very few signal samples [1, 2, 18]. This property is appealing for applications where the acquisition of signal values (labels) is costly. Computationally, TV minimization can be implemented as highly scalable message passing protocols (see Section 5).

As shown in Sec. 3, TV minimization is, in a very precise sense, equivalent to optimizing network flows. The concept of network flows is somewhat dual to the concept of graph signals. While the domain of graph signals is the node set  $\mathcal{V}$  of a graph, network flows are defined on the edges  $\mathcal{E}$  of a graph.

**Definition 1.** A network flow  $\tilde{y} : \mathcal{E} \rightarrow \mathbb{R}$  with supplies  $v_i$ , assigns each directed edge  $e = (i, j) \in \mathcal{E}$  the value  $\tilde{y}_e$  with

- the capacity constraints:

$$|\tilde{y}_e| \leq \lambda W_e \text{ for each edge } e \in \mathcal{E}, \quad (5)$$

- and the conservation law:

$$\sum_{j \in \mathcal{N}^+(i)} \tilde{y}_{(i,j)} - \sum_{j \in \mathcal{N}^-(i)} \tilde{y}_{(j,i)} = v_i \text{ for each } i \in \mathcal{V}. \quad (6)$$

### 3. NETWORK LASSO AND ITS DUAL

The cluster assumption suggests to learn graph signals by balancing the empirical error with the TV  $\|\tilde{\mathbf{x}}\|_{\text{TV}}$ ,

$$\hat{\mathbf{x}} \in \arg \min_{\tilde{\mathbf{x}} \in \mathbb{R}^{|\mathcal{V}|}} (1/2) \sum_{i \in \mathcal{M}} (x_i - \tilde{x}_i)^2 + \lambda \|\tilde{\mathbf{x}}\|_{\text{TV}}. \quad (7)$$

The optimization problem (7) is a special case of the original (generic) nLasso formulation [11]. Since the objective function and the constraints in (7) are convex, the optimization problem (7) is a convex optimization problem [6].

The nLasso (7) implements regularized risk minimization using TV (4) as regularization term [23]. The solutions  $\hat{\mathbf{x}}$  of (7) make an optimal compromise between consistency with observed signal samples  $x_i$ , for  $i \in \mathcal{M}$ , and small TV  $\|\tilde{\mathbf{x}}\|_{\text{TV}}$ . The tuning parameter  $\lambda > 0$  in (7) allows to trade a small mean squared error (MSE)  $(1/2) \sum_{i \in \mathcal{M}} (\hat{x}_i - x_i)^2$  against a small TV  $\|\hat{\mathbf{x}}\|_{\text{TV}}$  of the recovered graph signal  $\hat{\mathbf{x}}$ . A large  $\lambda$  enforces small TV, while a small  $\lambda$  favours low MSE.

The non-smooth objective function in (7) rules out gradient (descent) methods. However, the objective function is the sum of two functions that can be efficiently minimized individually. This compositional structure of (7) can be exploited by defining a dual problem.

It turns out that this dual problem has an interpretation as network (flow) optimization [5]. Moreover, by jointly considering (7) and its dual, we obtain an efficient method for simultaneously solving both problems (see Section 5).

To define the dual problem we first rewrite nLasso (7) as

$$\hat{\mathbf{x}} \in \arg \min_{\tilde{\mathbf{x}} \in \mathbb{R}^n} \mathcal{L}(\tilde{\mathbf{x}}) := g(\mathbf{B}\tilde{\mathbf{x}}) + h(\tilde{\mathbf{x}}), \quad (8)$$

with the incidence matrix  $\mathbf{B}$  (see (3)) and

$$g(\tilde{\mathbf{y}}) := \sum_{e \in \mathcal{E}} \lambda W_e |\tilde{y}_e|, \text{ and } h(\tilde{\mathbf{x}}) := (1/2) \sum_{i \in \mathcal{M}} (\tilde{x}_i - x_i)^2. \quad (9)$$

We refer to (8) as the primal problem (or formulation) of nLasso (7). The dual problem is

$$\hat{\mathbf{y}} \in \operatorname{argmax}_{\mathbf{y} \in \mathbb{R}^{|\mathcal{E}|}} \mathcal{D}(\mathbf{y}) := -h^*(-\mathbf{B}^T \mathbf{y}) - g^*(\mathbf{y}). \quad (10)$$

The objective function  $\mathcal{D}(\mathbf{y})$  of the dual problem (10) is composed of the convex conjugates (see (1)) of the components  $h(\mathbf{x})$  and  $g(\mathbf{y})$  of the primal problem (8). These convex conjugates are given explicitly by

$$h^*(\tilde{\mathbf{x}}) = \sup_{\mathbf{z} \in \mathbb{R}^n} \mathbf{z}^T \tilde{\mathbf{x}} - h(\mathbf{z}) \quad (11)$$

$$\stackrel{(9)}{=} \begin{cases} \infty & \text{if } \tilde{x}_i \neq 0 \text{ for some } i \in \mathcal{V} \setminus \mathcal{M} \\ (1/2) \sum_{i \in \mathcal{M}} \tilde{x}_i x_i + \tilde{x}_i^2/2 & \text{otherwise,} \end{cases}$$

and

$$g^*(\mathbf{y}) = \sup_{\mathbf{z} \in \mathbb{R}^{\mathcal{E}}} \mathbf{z}^T \mathbf{y} - g(\mathbf{z}) \stackrel{(9)}{=} \sup_{\mathbf{z} \in \mathbb{R}^{\mathcal{E}}} \mathbf{z}^T \mathbf{y} - \lambda \sum_{e \in \mathcal{E}} W_e |z_e|$$

$$= \begin{cases} \infty & \text{if } |y_e| > \lambda W_e \text{ for some } e \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

The relation between the primal problem (8) and the dual problem (10) is made precise by [21, Thm. 31.3]. In particular, the optimal values of (8) and (10) coincide:

$$\min_{\tilde{\mathbf{x}} \in \mathbb{R}^n} \underbrace{g(\mathbf{B}\tilde{\mathbf{x}}) + h(\tilde{\mathbf{x}})}_{\mathcal{L}(\tilde{\mathbf{x}})} = \max_{\mathbf{y} \in \mathbb{R}^{\mathcal{E}}} \underbrace{-h^*(-\mathbf{B}^T \mathbf{y}) - g^*(\mathbf{y})}_{\mathcal{D}(\mathbf{y})}. \quad (13)$$

According to [21, Thm. 31.3], a pair of vectors  $\hat{\mathbf{x}} \in \mathbb{R}^{|\mathcal{V}|}$ ,  $\hat{\mathbf{y}} \in \mathbb{R}^{\mathcal{E}}$  are solutions to the primal (8) and dual problem (10), respectively, if and only if

$$-(\mathbf{B}^T \hat{\mathbf{y}}) \in \partial h(\hat{\mathbf{x}}), \mathbf{B}\hat{\mathbf{x}} \in \partial g^*(\hat{\mathbf{y}}). \quad (14)$$

Given any dual solution  $\hat{\mathbf{y}} \in \mathbb{R}^{\mathcal{E}}$  to (10), every nLasso solution  $\hat{\mathbf{x}}$  must satisfy (14). The condition (14) also motivates a primal-dual method to solve (7) (see Section 5).

Our main result is the equivalence of the nLasso dual (10) to a minimum-cost flow problem for an extended empirical graph  $\tilde{\mathcal{G}}$  (see Fig. 1-(b)). The graph  $\tilde{\mathcal{G}}$  is obtained from the empirical graph  $\mathcal{G}$  by adding the node  $\star$  and the edges  $(i, \star)$  for each sampled node  $i \in \mathcal{M}$ .

**Proposition 2.** *The dual problem (10) of nLasso (7) is equivalent to the minimum-cost flow problem*

$$\min_{\tilde{\mathbf{y}} \in \mathbb{R}^{\mathcal{E}}} \sum_{i \in \mathcal{M}} \tilde{y}_{(i, \star)} ((1/2)\tilde{y}_{(i, \star)} - x_i), \quad (15)$$

$$\text{s.t. } \sum_{j \in \mathcal{N}^+(i)} \tilde{y}_{(i, j)} - \sum_{j \in \mathcal{N}^-(i)} \tilde{y}_{(j, i)} = 0 \text{ for all } i \in \mathcal{V} \cup \{\star\} \quad (16)$$

$$|\tilde{y}_e| \leq \lambda W_e \text{ for all } e \in \mathcal{E}.$$

The capacity constraints in (16) do not include the augmented edges  $(i, \star)$  for  $i \in \mathcal{M}$ . The role of the nLasso parameter  $\lambda$  in (16) is a scaling of the edge capacities  $W_{i, j}$ .

The problem (15) is an instance of a minimum-cost flow problem with convex separable cost functions (see [5, Ch. 8]). Efficient methods for such flow problems are presented in [5]. The special case of a minimum-cost flow problem with convex quadratic functions, such as in (15), is studied in [16, 24]. Instead of applying network flow methods, we will directly solve nLasso using a primal-dual method (see Section 5).

#### 4. STATISTICAL ASPECTS

To study nLasso solutions (7), we will use a simple but useful model which implements the cluster assumption Asspt. 1,

$$x_i = c_k \text{ for all nodes } i \in \mathcal{C}_k \text{ with coefficients } c_k \in \mathbb{R}. \quad (17)$$

The signal model (17) involves an arbitrary but fixed partition  $\mathcal{F} = \{\mathcal{C}_1, \dots, \mathcal{C}_F\}$  of the nodes  $\mathcal{V}$  into disjoint clusters. Piecewise constant signals are a special case of the large class of piecewise polynomial graph signals [10].

Combining Proposition 2 with the optimality condition (see [5, Prop. 8.2.]) offers a concise characterization of nLasso (7) solutions via existence of certain network flows.

**Corollary 3.** *Consider a graph signal (17) and a flow  $\tilde{y}_e$  on  $\tilde{\mathcal{G}}$  satisfying (16) and*

$$|\tilde{y}_e| \begin{cases} = \lambda W_e \text{ for } e \in \partial \mathcal{F}, \text{ and} \\ < \lambda W_e \text{ otherwise.} \end{cases} \quad (18)$$

*The flow  $\tilde{y}_e$  solves (15) if and only if, for each cluster  $\mathcal{C}_k$ ,*

$$x_i - \tilde{y}_{(i, \star)} = x_j - \tilde{y}_{(j, \star)} \text{ for any } i, j \in \mathcal{C}_k \cap \mathcal{M}. \quad (19)$$

*Given a flow  $\tilde{y}_e$  satisfying (18), (19), any nLasso (7) solution  $\hat{\mathbf{x}}$  is constant over non-saturated edges,*

$$\hat{x}_i = \hat{x}_j \text{ for } (i, j) \in \mathcal{E} \text{ with } |\tilde{y}_{(i, j)}| < \lambda W_{i, j}. \quad (20)$$

*Moreover, for each  $j \in \mathcal{C}_k$ , the nLasso value is*

$$\hat{x}_j = x_i - \left[ \sum_{i' \in \mathcal{N}^+(i)} \tilde{y}_{(i, i')} - \sum_{i' \in \mathcal{N}^-(i)} \tilde{y}_{(i', i)} \right] \text{ for some } i \in \mathcal{M} \cap \mathcal{C}_k. \quad (21)$$

*Proof.* The optimality condition [5, Prop. 8.2.] reveals that conditions (18) and (19) are necessary and sufficient for the flow  $\tilde{y}$  to be a solution to the minimum-cost flow problem (15).

Consider a flow  $\tilde{y}$  that satisfies (18) and (19) and therefore solves (15). By Theorem 2, we can obtain a solution  $\hat{\mathbf{y}}$  to the nLasso dual problem (10) by  $\hat{y}_e := \tilde{y}_e$  for each edge  $e \in \mathcal{E}$ .

Given this particular (optimal) dual solution  $\hat{\mathbf{y}}$ , any solution  $\hat{\mathbf{x}}$  to TV minimization has to satisfy (14). Combining (14) with properties of the sub-differential  $\partial g^*(\mathbf{y})$  (see (12) and [21, Sec. 32]) yields (20).  $\square$

The usefulness of Prop. 3 depends on the ability to construct flows on  $\tilde{\mathcal{G}}$  satisfying (18) and (19). This might be easy

for simple graph structures such as chains (see Sec. 6). Another option is to use tractable probabilistic models, such as stochastic block models, for the empirical graph [17]. A large deviation analysis allows then to obtain characterization of network flows that hold with high probability [14].

The condition (18) can be used to guide the choice of the nLasso parameter  $\lambda$  (see (7)). Using a larger value  $\lambda$  will typically make condition (3) more likely to be satisfied, if the clusters  $C_k$  are sufficiently well connected. However, larger values of  $\lambda$  will result in a bias of the nLasso estimate  $\hat{x}_i$  due to (21). Thus, condition (3) and (21) can help to avoid choosing  $\lambda$  neither too small (which would make (3) unlikely to hold) nor too large (which would imply a large bias via (21)).

## 5. COMPUTATIONAL ASPECTS

The characterization (14) of solutions to the nLasso (7) and its dual suggests to apply a convex primal-dual method [8]. The implementation of this method follows similar lines as in [13] and results in the iterations

$$\tilde{x}_i := 2\hat{x}_i^{(k)} - \hat{x}_i^{(k-1)} \text{ for } i \in \mathcal{V} \quad (22)$$

$$\hat{y}_e^{(k+1)} := \hat{y}_e^{(k)} + (1/2)(\tilde{x}_i - \tilde{x}_j) \text{ for } e = (i, j) \in \mathcal{E} \quad (23)$$

$$\hat{y}_e^{(k+1)} := \hat{y}_e^{(k+1)} \max\{1, |\hat{y}_e^{(k+1)}|/(\lambda W_e)\} \text{ for } (i, j) \in \mathcal{E} \quad (24)$$

$$\hat{x}_i^{(k+1)} := \hat{x}_i^{(k)} - \gamma_i \left[ \sum_{j \in \mathcal{N}^+(i)} \hat{y}_{(i,j)}^{(k+1)} - \sum_{j \in \mathcal{N}^-(i)} \hat{y}_{(j,i)}^{(k+1)} \right] \text{ for } i \in \mathcal{V} \quad (25)$$

$$\hat{x}_i^{(k+1)} := (\gamma_i x_i + \hat{x}_i^{(k+1)})/(\gamma_i + 1) \text{ for every } i \in \mathcal{M} \quad (26)$$

$$\bar{x}_i^{(k)} := (1 - 1/k)\bar{x}_i^{(k-1)} + (1/k)\hat{x}_i^{(k)} \text{ for } i \in \mathcal{V}. \quad (27)$$

Here,  $k = 0, 1, \dots$  denotes the iteration counter and  $\gamma_i := 1/d_i$  is the inverse node degree (2).

Standard results on convergence of primal-dual methods ensure that, irrespective of the initializations  $\hat{x}^{(0)}$  and  $\hat{y}^{(0)}$ , the iterates  $\bar{x}^{(k)}$  converge to a solution of nLasso (7) [8]. Moreover, the rate at which the sub-optimality, in terms of objective value, decreases is  $1/k$ . This rate is essentially optimal, in a worst-case sense, for message passing methods [12].

We can interpret the updates (22)-(27) as a message passing rules for network-flow optimization []. In particular, the iterate  $\hat{y}^{(k)}$  is a flow which tends to a solution  $\hat{y}$  of the nLasso dual problem (10).

The update (24) aims at enforcing the capacity constraints (5) for the flow iterate  $\hat{y}^{(k)}$ . The update (25) amounts to adjusting the current nLasso estimate  $\hat{x}_i^{(k)}$ , for each node  $i \in \mathcal{V}$  by the demand induced by the current flow  $\hat{y}^{(k)}$ . Thus, (25) enforces the conservation law (6) with demands  $v_i = \hat{x}_i^{(k)}$ .

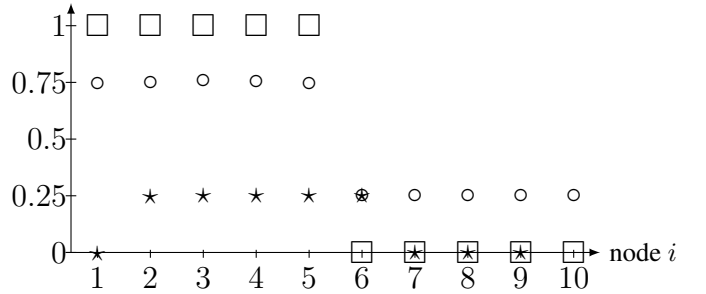
For each unobserved node  $i \in \mathcal{V} \setminus \mathcal{M}$ , we can interpret the iterate  $\hat{x}_i^{(k)}$  as the (scaled) cumulative demand induced by the flows  $\hat{y}^{(k')}$  for  $k' = 1, \dots, k$ . The labeled nodes  $i \in \mathcal{M}$  have a constant supply  $\hat{x}_i^{(k)} = x_i$  whose amount is the label  $x_i$ . The update (23) balances the discrepancies between the

cumulated demands  $\hat{x}_i^{(k)}$  by adapting the flow  $\hat{y}_{(i,j)}^{(k)}$  through an edge  $e = (i, j) \in \mathcal{E}$  according to the difference  $(\tilde{x}_i - \tilde{x}_j)$ .

## 6. NUMERICAL EXPERIMENTS

We verify Prop. 3 numerically using a chain-structured empirical graph  $\mathcal{G}$  which might represent time series data [7]. The chain structured empirical graph  $\mathcal{G}$  contains  $n = 10$  nodes which are partitioned into two clusters  $C_1 = \{1, \dots, 5\}$  and  $C_2 = \{6, \dots, 10\}$ . Intra-cluster edges  $e$  connecting nodes within the same cluster have unit weight  $W_e = 1$ , while the boundary edge  $e = \{5, 6\}$  has weight  $W_e = 1/4$ .

We iterate the updates (22)-(27) for a fixed number of  $K = 1000$  iterations to recover a piece-wise constant graph signal (17), with  $c_1 = 1, c_2 = 0$ , from its values on the sampling set  $\mathcal{M} = \{2, 7\}$ . The nLasso parameter was set to  $\lambda = 1$  (see (7)). To construct a flow  $\hat{y}_e$ , for edges  $e = (i, i+1)$ , that solves



**Fig. 2.** Piece-wise constant graph signal  $x_i$  ("□"), nLasso iterate  $\hat{x}_i^{(K)}$  ("○") and dual iterate  $\hat{y}_{(i,i+1)}^{(K)}$  ("★").

the nLasso dual (15), we have to ensure (18) and (19). The condition (18) is ensured by choosing  $\hat{y}_e = \lambda W_e = 1/4$  for the boundary edge  $e = \{5, 6\}$ . The condition (19) is trivially satisfied since each cluster contains only one single sampled node. The flow values  $\hat{y}_e$  of intra-cluster edges  $(i, i+1)$ , with  $i \neq 5$ , can be determined using the conservation law (16) for each node  $i \notin \mathcal{M}$  outside the sampling set. This results in

$$\hat{y}_{(i,i+1)} = \begin{cases} 1/4 & \text{for } i = 2, \dots, 6 \\ 0 & \text{otherwise.} \end{cases} \quad (28)$$

which resembles the iterate  $\hat{y}_e^{(K)}$  in Fig. 2. Given the dual solution  $\hat{y}_e$ , we obtain an nLasso solution  $\hat{x}$  via (20), (21). The simulation source code is available at <https://github.com/alexjungaalto/ResearchPublic/tree/master/FlowsNLasso>.

## 7. CONCLUSION

We have developed a theory of duality between nLasso and a minimum-cost network flow problem. This duality has been used to characterize nLasso solutions via the existence of certain network flows. Our work opens up several interesting research directions. It is interesting to study how parametric flow algorithms could be used to efficiently compute entire nLasso solution paths for varying  $\lambda$  in (7). Understanding the behavior of nLasso in terms of network flows could help to

guide model reduction techniques by sparsifying the empirical graph without scarifying nLasso accuracy [3].

## 8. REFERENCES

- [1] A. El Alaoui, X. Cheng, A. Ramdas, M. J. Wainwright, and M. I. Jordan. Asymptotic behavior of  $\ell_p$ -based Laplacian regularization in semi-supervised learning. In *Conf. on Learn. Th.*, pages 879–906, June 2016.
- [2] A.I. Aviles-Rivero, N. Papadakis, R. Li, S.M. Alsaleh, and R.T. Tan C.-B. Schoenlieb. Beyond supervised classification: Extreme minimal supervision with the graph 1-laplacian. *hal-02170176*, 2019.
- [3] J. Batson, D.A. Spielman, N. Srivastava, and S.-H. Teng. Spectral sparsification of graphs: theory and algorithms. *Communications of the ACM*, Aug. 2013.
- [4] H. Bauschke and P. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, New York, 2nd edition, 2017.
- [5] Dimitri P. Bertsekas. *Network Optimization: Continuous and Discrete Models*. Athena Scientific, 1998.
- [6] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge Univ. Press, Cambridge, UK, 2004.
- [7] P. J. Brockwell and R. A. Davis. *Time Series: Theory and Methods*. Springer New York, 1991.
- [8] 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.
- [9] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. The MIT Press, Cambridge, Massachusetts, 2006.
- [10] 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.
- [11] D. Hallac, J. Leskovec, and S. Boyd. Network lasso: Clustering and optimization in large graphs. In *Proc. SIGKDD*, pages 387–396, 2015.
- [12] A. Jung. On the complexity of sparse label propagation. *Front. Appl. Math. Stat.*, 4:22, July 2018.
- [13] A. Jung and N. Tran. Localized linear regression in networked data. *IEEE Sig. Proc. Lett.*, 26(7), Jul. 2019.
- [14] D.R. Karger. Random sampling in cut, flow, and network design problems. *Mathematics of Operations Research*, 24(2), 1999.
- [15] J. Mairal, R. Jenatton, F.R. Bach, and G.R. Obozinski. Network flow algorithms for structured sparsity. In *Advances in Neural Information Processing Systems 23*, pages 1558–1566, 2010.
- [16] M. Minoux. A polynomial algorithm for minimum quadratic cost flow problems. *European Journal of Operational Research*, 18(3):377–387, Dec. 1984.
- [17] E. Mossel, J. Neeman, and A. Sly. Stochastic block models and reconstruction. *ArXiv e-prints*, Feb. 2012.
- [18] 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.
- [19] M. E. J. Newman. *Networks: An Introduction*. Oxford Univ. Press, 2010.
- [20] T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *IEEE ICCV*, Barcelona, Spain, Nov. 2011.
- [21] R. T. Rockafellar. *Convex Analysis*. Princeton Univ. Press, Princeton, NJ, 1970.
- [22] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.
- [23] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1999.
- [24] L.A. Vegh. A strongly polynomial algorithm for a class of minimum- cost flow problems with separable convex objectives. *SIAM J. Comput.*, 45(5):1729–1761, 2016.
- [25] Y.-X. Wang, J. Sharpnack, A.J. Smola, and R.J. Tibshirani. Trend filtering on graphs. *Journal of Machine Learning Research*, 17(105):1–41, 2016.
- [26] H. Yin, A.R. Benson, J. Leskovec, and D.F. Gleich. Local higher-order graph clustering. In *Proc. KDD’17*, Halifax, NS, Canada, Aug. 2017.

## 9. SUPPLEMENTARY MATERIAL

To derive (22)-(27) as a fixed point iteration based on the optimality condition (14), we rewrite (14) as

$$\begin{aligned}\hat{\mathbf{x}} - \mathbf{\Gamma}\mathbf{B}^T\hat{\mathbf{y}} &\in \hat{\mathbf{x}} + \mathbf{\Gamma}\partial h(\hat{\mathbf{x}}) \\ 2\mathbf{\Lambda}\mathbf{B}\hat{\mathbf{x}} + \hat{\mathbf{y}} &\in \mathbf{\Lambda}\partial g^*(\hat{\mathbf{y}}) + \mathbf{\Lambda}\mathbf{B}\hat{\mathbf{x}} + \hat{\mathbf{y}},\end{aligned}\quad (29)$$

with the invertible diagonal matrices

$$\mathbf{\Lambda} := (1/2)\mathbf{I} \in \mathbb{R}^{\mathcal{E} \times \mathcal{E}}, \mathbf{\Gamma} := \text{diag}\{\gamma_i = 1/d_i\}_{i=1}^n \in \mathbb{R}^{n \times n}. \quad (30)$$

The particular choice (30) ensures that [20, Lemma 2]

$$\|\mathbf{\Gamma}^{1/2}\mathbf{B}^T\mathbf{\Lambda}^{1/2}\|_2 < 1,$$

which ensure converge of the proposed method. There are other choices than (30) that ensure convergence. Data-driven tuning of the matrices  $\mathbf{\Gamma}$ ,  $\mathbf{\Lambda}$  is beyond the scope of this paper.

We further develop the characterization (29) using the resolvent operators for the (set-valued) operators  $\mathbf{\Lambda}\partial g^*(\mathbf{y})$  and  $\mathbf{\Gamma}\partial h(\mathbf{x})$  (see (8) and [20, Sec. 1.1.]),

$$\begin{aligned}(\mathbf{I} + \mathbf{\Lambda}\partial g^*)^{-1}(\tilde{\mathbf{y}}) &:= \arg \min_{\mathbf{z} \in \mathbb{R}^{|\mathcal{E}|}} g^*(\mathbf{z}) + (1/2)\|\tilde{\mathbf{y}} - \mathbf{z}\|_{\mathbf{\Lambda}^{-1}}^2 \\ (\mathbf{I} + \mathbf{\Gamma}\partial h)^{-1}(\tilde{\mathbf{x}}) &:= \arg \min_{\mathbf{z} \in \mathbb{R}^{|\mathcal{V}|}} h(\mathbf{z}) + (1/2)\|\tilde{\mathbf{x}} - \mathbf{z}\|_{\mathbf{\Gamma}^{-1}}^2.\end{aligned}\quad (31)$$

Applying [4, Prop. 23.2] and [4, Prop. 16.44] to the optimality condition (29) yields the equivalent condition (for  $\hat{\mathbf{x}}$ ,  $\hat{\mathbf{y}}$  to be primal and dual optimal)

$$\begin{aligned}\hat{\mathbf{x}} &= (\mathbf{I} + \mathbf{\Gamma}\partial h)^{-1}(\hat{\mathbf{x}} - \mathbf{\Gamma}\mathbf{B}^T\hat{\mathbf{y}}) \\ \hat{\mathbf{y}} - 2(\mathbf{I} + \mathbf{\Lambda}\partial g^*)^{-1}\mathbf{\Lambda}\mathbf{B}\hat{\mathbf{x}} &= (\mathbf{I} + \mathbf{\Lambda}\partial g^*)^{-1}(\hat{\mathbf{y}} - \mathbf{\Lambda}\mathbf{B}\hat{\mathbf{x}}).\end{aligned}\quad (32)$$

The fixed point characterization (32) of nLasso solutions suggests the following coupled fixed-point iterations:

$$\begin{aligned}\hat{\mathbf{y}}^{(k+1)} &:= (\mathbf{I} + \mathbf{\Lambda}\partial g^*)^{-1}(\hat{\mathbf{y}}^{(k)} + \mathbf{\Lambda}\mathbf{B}(2\hat{\mathbf{x}}^{(k)} - \hat{\mathbf{x}}^{(k-1)})) \\ \hat{\mathbf{x}}^{(k+1)} &:= (\mathbf{I} + \mathbf{\Gamma}\partial h)^{-1}(\hat{\mathbf{x}}^{(k)} - \mathbf{\Gamma}\mathbf{B}^T\hat{\mathbf{y}}^{(k+1)}).\end{aligned}\quad (33)$$

The fixed-point iterations (33) are obtained as a special case of the iterations [20, Eq. (4)] when choosing  $\theta = 1$  (using the notation in [20]).

The updates in (33) allow for simple closed-form expressions (see [8, Sec. 6.2.] for more details). Inserting these expressions into (33) yields the updates (22)-(27) for iterative solving of nLasso (7).

**Bounding Sub-Optimality.** The identity (13) allows to bound the sub-optimality  $\mathcal{L}(\tilde{\mathbf{x}}) - \mathcal{L}(\hat{\mathbf{x}})$  of a given candidate  $\tilde{\mathbf{x}}$  for the solution of nLasso (8). Inserting an arbitrary dual vector  $\mathbf{y}$  into (13),

$$\mathcal{L}(\tilde{\mathbf{x}}) - \mathcal{L}(\hat{\mathbf{x}}) \leq \mathcal{L}(\tilde{\mathbf{x}}) - \mathcal{D}(\mathbf{y}). \quad (34)$$

Note that the right hand side in (34) can be evaluated for any given pair  $\tilde{\mathbf{x}}$ ,  $\mathbf{y}$  of primal and dual vectors.

**Stopping Criteria.** Possible stopping criteria for the updates (22)-(27) include a fixed number of iterations or testing for a sufficiently small sub-optimality gap  $\mathcal{L}(\tilde{\mathbf{x}}^{(k)}) - \mathcal{L}(\hat{\mathbf{x}})$ . We can ensure a maximum sub-optimality gap using the bound (34). When using a fixed number  $K$  of iterations, one can use well-known results on the convergence rate of primal-dual methods [8]. Roughly speaking these results imply that the sub-optimality of the iterates  $\tilde{\mathbf{x}}^{(K)}$  decrease according to  $\propto 1/K$ . The convergence rate  $1/K$  is tight among all message passing methods to solve (7). It is attained in chain-structured graphs (see [12]).

**Proof of Proposition 2** We first note that (10) is equivalent to

$$\min_{\hat{\mathbf{y}} \in \mathbb{R}^{\mathcal{E}}} \sum_{i \in \mathcal{M}} v_i((1/2)v_i - x_i), \quad (35)$$

$$\text{s.t. } \sum_{j \in \mathcal{N}^+(i)} \hat{y}_{(i,j)} - \sum_{j \in \mathcal{N}^-(i)} \hat{y}_{(j,i)} = \begin{cases} v_i & \text{for } i \in \mathcal{M} \\ 0 & \text{otherwise.} \end{cases} \quad (36)$$

$$|\hat{y}_{(i,j)}| \leq \lambda W_e \text{ for all } e \in \mathcal{E}.$$

The definition (11) and (12) for the components of (10) enforce implicit constraints on the dual vector that are identical with the constraints (36). Thus, any optimal dual vector  $\hat{\mathbf{y}}$  solving (10), must satisfy the constraints (36)). However, the objective functions in (35) and (10) coincide when evaluated for vectors  $\mathbf{y}$  satisfying (35).

The final step of the proof is to verify equivalence of the optimization problems (35) and (15). To this end, we note that the additional edges  $(i, \star)$  in  $\tilde{\mathcal{G}}$  (see Fig. 1-(b)) have no capacity constraints, or “unbounded” capacity, which allows them to “discharge” the node demands  $v_i$ , for  $i \in \mathcal{M}$ .

Consider an optimal flow  $\hat{\mathbf{y}}$  which solves (35). We then construct a flow  $\tilde{\mathbf{y}}_e$  on the extended graph  $\tilde{\mathcal{G}}$  by setting  $\tilde{y}_e := \hat{y}_e$  for all intra-cluster edges  $e \in \mathcal{E} \setminus \partial\mathcal{F}$  and  $\tilde{y}_{(i,\star)} := v_i$  for all sampled nodes  $i \in \mathcal{M}$ .

The accumulating node “ $\star$ ” has only inward edges resulting in the total demand  $\sum_{i \in \mathcal{M}} v_i$ . However, this sum is zero since the demands  $v_i$  of the flow in (35) must sum to zero (see [5, Chap. 1]).