ON THE DUALITY BETWEEN NETWORK FLOWS AND NETWORK LASSO

Alexander Jung

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

ABSTRACT

The data arising in many application domains can have an intrinsic network structure. Such network structure is computationally appealing due to the availability of highly scalable graph algorithms. An important class of graph algorithms is related to optimizing network flows. This paper explores the duality of network flow methods and the recently proposed network Lasso. Network Lasso extends the Lasso method from sparse linear models to clustered graph signals. It turns out that the computational and statistical properties of network Lasso crucially depend on the existence of sufficiently large network flows. Using elementary tools from convex analysis, we offer a precise characterization of the duality between network Lasso and a minimum cost network flow problem. This duality provides a strong link between network Lasso methods and network flow algorithms.

1. INTRODUCTION

The data arising in many important application domains can be modeled efficiently using some network structure. The network structure of data might arise from physical proximity (in time or space), physical connection (communication networks) or statistical dependency (probabilistic graphical models) [3]. Examples of such networked data are found in signal processing where signal samples can be arranged as a chain, in image processing with pixels arranged on a grid and wireless sensor networks where measurements conform to sensor proximity [5][13][14][19]. Organizing data using networks is also used in knowledge bases (graphs) whose items are linked by relations [18][20].

Efficient methods to process networked data are offered by graph algorithms such as clustering or network flow optimization [2][9][12]. While these graph algorithms only use the network structure, joint clustering and optimization methods such as network Lasso also take additional information into account [10]. This additional information is conveyed in signal values associated with individual data points.

Contribution. We show the a dual problem of network Lasso is equivalent to a minimum-cost flow problem. The solutions of this minimum-cost flow problem can be used to partially characterize solutions of network Lasso. A primal-dual method is implemented to jointly solve the network Lasso and its dual. We provide an interpretation of this method as a network flow optimization.

Notation. The sub-differential of a function $g(x)$ at $x_0 \in \mathbb{R}^n$ is the set

$$\partial g(x_0) := \{y \in \mathbb{R}^n : g(x_0) + y^T(x - x_0) \leq g(x) \text{ for any } x\}.$$ 

The convex conjugate function of $g(x)$ is [3]

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

2. RECOVERING CLUSTERED GRAPH SIGNALS

We represent networked data as an undirected empirical graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ [6]. The nodes $i \in \mathcal{V}$ of the empirical graph represent 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 $i, j \in \mathcal{V}$.

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

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

The maximum (weighted) node degree is

$$d_{\max} := \max_{i \in \mathcal{V}} d_i = \sum_{j \in \mathcal{N}(i)} W_{i,j}. \quad (3)$$

For a given undirected empirical graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, we orient the undirected edge $(i, j)$ by defining the head as $e^+ = \min\{i, j\}$ and the tail as $e^- = \max\{i, j\}$. We use $\mathbf{G}$ and $\mathcal{E}$ to also denote the oriented empirical graph and its directed edges, respectively. The incidence matrix $\mathbf{B} \in \mathbb{R}^{\mathcal{E} \times n}$ of the empirical graph $G$ is

$$B_{e,i} = \begin{cases} W_e & \text{if } i = e^+ \\ -W_e & \text{if } i = e^- \\ 0 & \text{else.} \end{cases} \quad (4)$$

In many applications, the goal is to determine (or infer) some relevant property beside the network structure, datasets carry additional information which we represent by a graph signal $\mathbf{x} = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$. 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 \( i \in \mathcal{V} \setminus \mathcal{M} \).

Given the signal samples \( x_i \) for data points \( i \in \mathcal{M} \) in the training set, we want to recover the entire graph signal \( x \in \mathbb{R}^n \). This learning (or recovery) problem is feasible, if the underlying graph signal \( x \) has a known structure. As mentioned above, a particular structure is obtained if the labels \( x_i \) conform with the cluster structure of the empirical graph \( \mathcal{G} \).

The optimization problem (7) is a special case of capacitated minimum (flow) optimization \[2\]. 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

\[
\hat{x} \in \arg \min_{x \in \mathbb{R}^n} \mathcal{L}(x) := g(D\hat{x}) + h(\hat{x}),
\]

with the scaled incidence matrix \( D := \lambda B \) (see (4)) and

\[
g(\hat{y}) := \|\hat{y}\|_1, \quad \text{and} \quad h(\hat{x}) := (1/2) \sum_{i \in \mathcal{M}} (\hat{x}_i - x_i)^2.
\]

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

\[
\hat{y} \in \arg\max_{y \in \mathbb{R}^{|E|}} \mathcal{D}(\hat{y}) := -h^*( -D^T \hat{y} ) - g^*(\hat{y}). \tag{9}
\]

The objective function \( f(y) \) of the dual problem (9) is composed of the convex conjugates (see (1)) of the components \( h(x) \) and \( g(y) \) of the primal problem (8). These convex conjugates are given explicitly by

\[
h^*(\hat{x}) = \sup_{z \in \mathbb{R}^n} \hat{z}^T \hat{x} - h(z) \tag{10}
\]

and

\[
g^*(y) = \sup_{z \in \mathbb{R}^{|E|}} z^T y - g(z) \tag{11}
\]

The optimization problem (7) is a special case of the network Lasso (nLasso) \[10\]. Since the objective function and the constraints in (7) are convex, the optimization problem (7) is a convex optimization problem \[3\]. In fact, (7) can be reformulated as a linear program \[3\, \text{Sec. 1.2.2}\]. The solutions \( \hat{x} \) of (7) optimally balance between consistency with observed signal samples \( x_i \), for \( i \in \mathcal{M} \), and small TV \( \|\hat{x}\|_TV \).

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 \( \|\hat{x}\|_TV \) of the recovered graph signal \( \hat{x} \). 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 function 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 \[2\]. Moreover, by jointly considering (7) and its dual, we obtain an efficient method for simultaneously solving both problems (see Section 5).

The relation between the primal problem (8) and the dual problem (9) is composed explicitly by

\[
\mathcal{L}(\hat{x}) = \inf_{y \in \mathbb{R}^{|E|}} \mathcal{D}(\hat{y}) = \mathcal{D}(\hat{y})
\]

The identity (12) allows to bound the sub-optimality \( \mathcal{L}(\hat{x}) - \mathcal{L}(\hat{x}) \) of a given candidate \( \hat{x} \) for the solution of nLasso (8).
Indeed, according to (12), given any dual vector \( y \) we can bound its sub-optimality as

\[
\mathcal{L}(\hat{x}) - \mathcal{L}(\hat{x}) \leq \mathcal{L}(\hat{x}) - \mathcal{D}(\hat{y}).
\]

(13)

Note that the right hand side in (13) can be evaluated for any given pair \( x, y \) of primal and dual vectors.

Another consequence of (16), (31.3) is a characterization of the solutions of the primal (8) and dual problem (9). A pair of vectors \( \hat{x} \in \mathbb{R}^{|V|}, \hat{y} \in \mathbb{R}^{|E|} \) are solutions to the primal (8) and dual problem (9), respectively, if and only if

\[
-D^T \hat{y} \in \partial h(\hat{x}), \text{ } \hat{D} \hat{x} \in \partial g^*(\hat{y}).
\]

(14)

Given any dual solution \( \hat{y} \in \mathbb{R}^E \) to (9), every primal nLasso solution \( \hat{x} \) must satisfy (14). Condition (14) is the launching point for a primal-dual method to solve (7) (see Section 5).

Our main result is the equivalence of the nLasso dual (9) to maximizing network flows on the empirical graph \( G \). After making the notion of a network flow precise, we show how to associate a particular network flow with a dual vector \( y \).

**Definition 1.** A network flow \( f : E \rightarrow \mathbb{R} \) with supplies \( v_e \), at the nodes \( i \in V \), assigns each directed edge \( e = (i,j) \in E \) the value \( f_e \). The flow has to satisfy

- the capacity constraints:
  \[
  |f_e| \leq \Lambda W_e \text{ for each } e \in E,
  \]
  (15)

- and the conservation law:
  \[
  \sum_{j \in N^+(i)} f_{(i,j)} - \sum_{j \in N^-(i)} f_{(j,i)} = v_i \text{ for each } i \in V.
  \]
  (16)

Any dual vector \( y \in \mathbb{R}^E \) defines a flow \( f^{(y)}_e := \Lambda W_y e \). The flow \( f^{(y)} \) satisfies the capacity constraints (15) and the conservation law (16) with supplies \( v_i \) if and only if

\[
||y||_\infty \leq 1, D^T y = v \text{ with } v = (v_1, \ldots, v_n)^T \in \mathbb{R}^n.
\]

(17)

**Proposition 2.** The dual problem (9) of nLasso (7) is equivalent to the network flow optimization

\[
\max_{f : \mathbb{E} \rightarrow \mathbb{R}} \sum_{i \in \mathbb{M}} v_i \left( x_i + (1/2)v_i \right),
\]

(18)

\[
s.t. \sum_{j \in N^+(i)} f_{(i,j)} - \sum_{j \in N^-(i)} f_{(j,i)} = v_i \text{ for } i \in \mathbb{M}.
\]

\[
|f_{(i,j)}| \leq \Lambda W_e \text{ for all } e \in E.
\]

A vector \( y \) solves (9) if and only if \( f^{(y)}_e = \Lambda W_y e \) solves (18).

Proof. The implicit constraints arising from the definitions (10) and (11) of the functions that constitute the dual problem (9), enforce the dual vector \( y \) to satisfy (17) with supplies \( v_i = 0 \) for every unobserved node \( i \in V \setminus \mathbb{M} \). Any optimal dual vector \( \hat{y} \) solving (9), defines a valid flow \( f^{(y)} \) (satisfying the constraints in (18)). The objective functions in (18) and (9) coincide for dual vectors \( y \) such that \( f^{(y)} \) is feasible.

The problem (18) is an instance of a minimum-cost flow problem with convex separable cost functions (see [2, Ch. 8]). Efficient methods for such flow problems are presented in [2]. However, we will directly solve nLasso using a primal-dual method (see Section 5).

4. STATISTICAL ASPECTS

Combining Theorem [2] with the optimality condition (14) allows to characterize nLasso solutions via network flows.

**Corollary 3.** Given networked data with empirical graph \( G \) and labels \( \{x_i\}_{i \in \mathbb{M}}, \) consider some flow \( \hat{f} \) which solves the minimum-cost flow problem (18). Let us denote the set of edges which are not saturated in \( \hat{f} \) by

\[
U := \{ (i, j) \in E : |f_e| < \Lambda W_e \}.
\]

Then, any nLasso solution \( \hat{x} \) satisfies

\[
\hat{x}_i = \hat{x}_j \text{ for each } e = (i, j) \in U.
\]

(19)

Given any optimal flow \( \hat{f} \) (which solves (18)), each nLasso solution is constant along edges which are not saturated by \( \hat{f} \).

Proof. Given the optimal flow \( \hat{f} \), we define the dual vector \( \hat{y} = \hat{f}/(\Lambda W_e \hat{x}) \). By Theorem [2] \( \hat{y} \) solves the dual problem (9). For this particular (optimal) dual vector \( \hat{y} \), any solution \( \hat{x} \) to TV minimization has to satisfy the optimality condition (14). Combining (14) with elementary properties of the subdifferential \( \partial g^*(\hat{y}) \) (see (11) and (16), Sec. 32)) yields (19).

Once we find at least one optimal flow \( \hat{f} \) such that a particular edge \( e = (i, j) \in E \) is not saturated, i.e., \( |f_e| < \Lambda W_e \), we are assured that every nLasso solution does not change over this edge \( e \). In order to apply Corollary [3] we need to be able construct or characterize solutions of (18). Since the network flow problem (18) involves a differentiable cost function, the optimality conditions presented in [2, Ch. 8] can be used.

5. COMPUTATIONAL ASPECTS

Section [4] characterized the solutions of nLasso (7). Here we consider methods for computing (approximate) solutions. This method is obtained as a fixed point iteration for a reformulation of the optimality conditions (14) as a fixed point equation. To this end, we rewrite (14) as

\[
\hat{x} - \Gamma D^T \hat{y} = \hat{x} + \Gamma \partial h(\hat{x})
\]

\[
2\Lambda D \hat{x} + \hat{y} \in \Lambda \partial g^*(\hat{y}) + \Lambda D \hat{x} + \hat{y},
\]

(20)

with the invertible diagonal matrices

\[
\Lambda := (1/2) \text{diag}(\lambda_{(i,j)} = 1/(\Lambda W_{e(i,j)})) \in \mathbb{R}^{E \times E}
\]

\[
\Gamma := \text{diag}(\gamma_i = 1/d_i)_{i=1}^n \in \mathbb{R}^{n \times n}.
\]

(21)
The particular choice (21) ensures that [15] Lemma 2
\[ \| \Gamma^{1/2} D^T \Lambda^{1/2} \|_2 < 1, \]
which ensure convergence of the proposed method. There are other choices than (21) that ensure convergence. Data-driven tuning of the matrices \( \Gamma, \Lambda \) is beyond the scope of this paper.

We further develop the characterization (20) using the resolvent operators for the (set-valued) operators \( \partial g^* (y) \) and \( \Gamma \partial h(x) \) (see [6] and [15] Sec. 1.1.),
\begin{align*}
(I + \Lambda \partial g^*)^{-1} (\tilde{y}) &:= \arg \min_{z \in \mathbb{R}^{|E|}} g^*(z) + (1/2) \| y - z \|_{\Lambda^{-1}}^2, \\
(I + \Gamma \partial h)^{-1} (\tilde{x}) &:= \arg \min_{z \in \mathbb{R}^{|V|}} h(z) + (1/2) \| x - z \|_{\Gamma^{-1}}^2.
\end{align*}

(22)

Applying [11] Prop. 23.2 and [11] Prop. 16.44 to the optimality condition (20) yields the equivalent condition (for \( \tilde{x}, \tilde{y} \) to be primal and dual optimal)
\begin{align*}
\tilde{x} &= (I + \Gamma \partial h)^{-1} (\tilde{x} - \Gamma D \tilde{y}), \\
\tilde{y} - 2(I + \Lambda \partial g^*)^{-1} \Lambda D \tilde{x} &= (I + \Lambda \partial g^*)^{-1} (\tilde{y} - \Lambda D \tilde{x}).
\end{align*}

(23)

The fixed point characterization (23) of nLasso solutions suggests the following coupled fixed-point iterations:
\begin{align*}
\mathbf{y}^{(k+1)} &= (I + \Lambda \partial g^*)^{-1} (\mathbf{y}^k + \Lambda D (2\mathbf{x}^k - \mathbf{x}^{(k-1)})), \\
\mathbf{x}^{(k+1)} &= (I + \Gamma \partial h)^{-1} (\mathbf{x}^k - \Gamma D^T \mathbf{y}^{(k+1)}).
\end{align*}

(24)

The convergence rate of Alg. 1 is tight among all message passing methods to solve (7). It is attained in chain-structured graphs (see [11]).

As indicated by [7] Thm. 3.2, Alg. 1 is robust to numerical errors arising during the updates, which can be a crucial property for high-dimensional problems.

The computational cost of one iteration in Alg. 1 is proportional to the number of edges in the empirical graph \( \mathcal{G} \). This can be verified by noting that Alg. 1 can be implemented as message passing on the empirical graph. Moreover, while Alg. 1 allows for rather straightforward implementation on modern big data computing frameworks, this is typically more challenging for maximum flow methods which are (partially) based on combinatorial search (see [15] Sec. 3.3.).

Alg. 1 implicitly solves the dual (2) of nLasso (7). We can interpret Alg. 1 as a message passing method for network optimization. In particular, associate the current approximation \( \mathbf{y}^k \) for the optimal dual vector \( \mathbf{y} \) (see (9)) with the flow \( f^k : \mathcal{E} \rightarrow \mathbb{R} \) having values \( f^k_e := W_e y^k_e \).

For each unobserved node \( i \in \mathcal{V} \setminus \mathcal{M} \), we can interpret the signal estimates \( \hat{x}_i^k \) as the (scaled) cumulative demand induced by the current flow \( f^k \).

For each unobserved node \( i \in \mathcal{V} \setminus \mathcal{M} \), we can interpret the signal estimates \( \hat{x}_i^k \) as the (scaled) cumulative demand induced by the current flow \( f^k \).

Algorithm 1: Primal-Dual Method for TV Minimization

Input: \( B \in \mathbb{R}^{E \times V}, \{ x_i \}_{i \in \mathcal{M}}, \lambda \)

Initialize: \( k := 0, \bar{x} = \hat{x}^{(-1)} = \hat{x}^{(0)} = \mathbf{y}^{(0)} = 0, \gamma_i := 1/d_i, \lambda_{(i,j)} := 1/(2\lambda W_{i,j}), \mathbf{D} := \lambda \mathbf{B} \)

\begin{enumerate}
\item \textbf{repeat}
\item \( \hat{x} := 2\hat{x}^k - \hat{x}^{(k-1)} \)
\item \( \mathbf{y}^{(k+1)} := \mathbf{y}^k + \Lambda D \hat{x} \) with \( \Lambda = \text{diag} \{ \lambda_{(i,j)} \}_{(i,j) \in \mathcal{E}} \)
\item \( \hat{y}_e^{(k+1)} := \hat{y}_e^{(k+1)} / \max \{ 1, |\hat{y}_e^{(k+1)}| \} \) for every edge \( e \in \mathcal{E} \)
\item \( \hat{x}^{(k+1)} := \hat{x}^k - \Gamma D^T \mathbf{y}^{(k+1)} \) with \( \Gamma = \text{diag} \{ \gamma_i \}_{i \in \mathcal{V}} \)
\item \( \hat{x}_i^{(k+1)} := \hat{x}_i^{(k+1)} / \gamma_i \) for every \( i \in \mathcal{M} \)
\item \( k := k + 1 \)
\item \( \hat{x} := (1 - 1/k) \hat{x}^{(k-1)} + (1/k) \hat{x}^k \)
\item \textbf{until} stopping criterion met
\end{enumerate}

Output: signal values \( \hat{x}_i := \hat{x}_i^k \) for all nodes \( i \in \mathcal{V} \)}
6. REFERENCES

[1] H. Bauschke and P. Combettes. Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer, New York, 2nd edition, 2017.

[2] Dimitri P. Bertsekas. Network Optimization: Continuous and Discrete Models. Athena Scientific, 1998.

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

[4] 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.

[5] L. F. O. Chamon and A. Ribeiro. Greedy sampling of graph signals. 2018, 66(1):34–47, 2018.

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

[7] 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.

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

[9] Andrew V. Goldberg and Robert E. Tarjan. Efficient maximum flow algorithms. Communications of the ACM, 57(8):82–89, 2014.

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

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

[12] J. Kleinberg and E. Tardos. Algorithm Design. Addison Wesley, 2006.

[13] S. G. Mallat. A Wavelet Tour of Signal Processing – The Sparse Way. Academic Press, San Diego, CA, 3 edition, 2009.

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

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

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

[17] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. Phys. D, 60(1-4):259–268, 1992.

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

[19] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Processing Magazine, 30(3):83–98, May 2013.

[20] D. Vrandečić and M. Krötzsch. Wikidata: A free collaborative knowledgebase. Commun. ACM, 57(10):78–85, Sep. 2014.

[21] 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.