Tran, Nguyen; Ambos, Henrik; Jung, Alexander

A Network Compatibility Condition for Compressed Sensing over Complex Networks

Published in:
2018 IEEE Statistical Signal Processing Workshop, SSP 2018

DOI:
10.1109/SSP.2018.8450811

Published: 29/08/2018

Please cite the original version:
A NETWORK COMPATIBILITY CONDITION FOR COMPRESSED SENSING OVER COMPLEX NETWORKS

Nguyen Tran, Henrik Ambos, Alexander Jung

Department of Computer Science, Aalto University

ABSTRACT
This paper continues our recently initiated line of work on analyzing the network Lasso (nLasso), which has been proposed as an efficient learning algorithm for massive network-structured data sets (big data over networks). The nLasso extends the well-known Lasso estimator to network-structured datasets. In this paper we consider the nLasso using squared error loss and provide sufficient conditions on the network structure and available label information such that nLasso accurately recovers a clustered (piece-wise constant) graph signal from the information provided by the labels of a few data points.

Index Terms— big data over networks, compressed sensing, complex networks, network compatibility condition, network Lasso

1. INTRODUCTION

The datasets arising in many important applications, ranging from image processing to social network analysis, carry an intrinsic network structure which we refer to as the data graph. The nodes of the data graph represent individual data points (e.g., users in a social network), which typically carry label information called graph signal [1]. These nodes are connected by edges according to some notion of similarity. The network structure might be induced naturally by the application at hand (e.g., friendship in a social network) or obtained by learning a probabilistic graphical model for the dataset when interpreted as the realization of a random process [2–5].

On top of the network structure of, the datasets often carry additional information (e.g., labels) which we will represent by a graph signal. Since the acquisition of graph signal values (labels) is often expensive, recovery the entire graph signal from a (small) subset of nodes (sampling set) is a crucial task. by , The graph signal recovery problem is possible by exploiting the tendency of natural graph signals to conform to the underlying network structure. More precisely, the clustering hypothesis, which underlies most (semi-) supervised machine learning methods [6, 7], requires the graph signal to be nearly constant over well connected subset of nodes (clusters).

In this paper, we further develop the program initiated in [8, 9] revolving around conditions which guarantee convex optimization methods, such as nLasso [10] or sparse label propagation [11], to accurately recover clustered graph signals from a small number of signal samples. In particular, we apply the network compatibility condition (NCC) introduced in [8] to analyze the nLasso squared error loss. We verify that the NCC ensures accurate recovery of a clustered (piece-wise constant) graph signal from only few signal values (initial labels) using nLasso.

This paper is organized as follows: In Section 2, we formalize the problem setup. In Section 3, we present our main result, i.e., that the NCC guarantees accuracy of nLasso using squared error loss if the true underlying graph signal is clustered. The results of some illustrative numerical experiments are discussed in Section 4, which also contains the ADMM update rules nLasso with squared error loss.

2. PROBLEM FORMULATION

In what follows we consider a dataset which is represented by its data graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{W})$. The nodes $\mathcal{V} = \{1, \ldots, N\}$ represent individual data points. The undirected edges $\mathcal{E}$ encode domain-specific notions of similarity between data points. The non-negative entries $W_{i,j}$ of the weight matrix $\mathbf{W} \in \mathbb{R}^{N \times N}_+$ quantify the level of similarity between connected nodes. In particular, the weight $W_{i,j}$ is non-zero only if nodes $i, j \in \mathcal{V}$ are connected by an edge $\{i, j\} \in \mathcal{E}$.

In addition to the graph structure $G$, datasets typically convey additional information, such as labels associated with individual data points $i \in \mathcal{V}$. In what follows, we represent any such label information using a graph signal $x[\cdot] : \mathcal{V} \to \mathbb{R}$, which assigns a real value $x[i] \in \mathbb{R}$ to node the $i \in \mathcal{V}$.

Our approach is based on the assumption that the graph signals $x[\cdot]$ encountered in applications are typically clustered, i.e., have small total variation (TV) [12]

$$
\|x[\cdot]\|_\mathcal{E} := \sum_{\{i,j\} \in \mathcal{E}} W_{i,j} |x[j] - x[i]|. 
$$

We will use the following simple model for clustered (piece-wised constant) graph signals [12]:

$$
x[i] = \sum_{\mathcal{C} \in \mathcal{F}} a_\mathcal{C} I_\mathcal{C}[i],
$$
with \( a_C \in \mathbb{R} \) and the indicator signal \( I_C[i] \in \{0, 1\} \) for subset \( C \subset V \), i.e., \( I_C[i] = 1 \) if and only if \( i \in C \). The model (2) involves a partition \( F = \{C_1, \ldots, C_M\} \) of \( V \) into disjoint subsets \( C_i \).

In principle our analysis applies to any partition \( F = \{C_1, \ldots, C_M\} \) used to define the model (2). However, our results are most useful if the partition conforms with the “intrinsic (cluster) structure” of the data graph \( G \). In particular, we consider a partition \( F \) such that the total weight of the cluster boundaries

\[
\partial F = \{ \{i, j\} \in E : i \in C, j \in C' (\neq C) \}
\]

is small compared to the total weight of intra-cluster edges \( \partial F = E \setminus \partial F \), i.e., \( \sum_{(i,j) \in \partial F} W_{i,j} \ll \sum_{(i,j) \in \partial F} W_{i,j} \). As can verified easily, for such a partition, any signal of the form (2) has a small TV \( \|x[\cdot]\|_E \).

We consider the case of having access to the graph signal values \( x[i] \) only for (small) subset of nodes, i.e., the sampling set \( M := \{i_1, \ldots, i_M\} \subseteq V \) (typically \(|M| \ll |V|\)). In particular, we observe

\[
y[i] = x[i] + \varepsilon[i] \quad \text{for a sampled node } i \in M. \tag{3}
\]

The error or noise component \( \varepsilon[i] \) in (3) subsumes any data curation or labelling errors. We do not assume any probabilistic model for the noise \( \varepsilon[i] \). However, for our analysis in what follows we have to require the noise is not excessively large. In particular, we require an upper bound

\[
\sum_{i \in M} \tilde{x}[i] \varepsilon[i] \leq \lambda \|\tilde{x}[\cdot]\|_E \tag{4}
\]

to hold for any non-constant graph signal \( \tilde{x}[\cdot] \in \mathbb{R}^N \), i.e., for which \( \|\tilde{x}\|_E \neq 0 \), and some positive constant \( \nu > 0 \).

The recovery of the entire graph signal \( x[\cdot] \) from the noisy measurements \( \{y[i]\}_{i \in M} \) rests on the assumption that the true underlying graph signal has small TV \( \|x[\cdot]\|_E \). Moreover, the recovered graph signal \( \hat{x}[\cdot] \) should agree well with the observed samples, i.e., it should incur a small empirical error

\[
\tilde{E}(\hat{x}[\cdot]):= \sum_{i \in M} (\hat{x}[i] - y[i])^2. \tag{5}
\]

Our approach to recovering the underlying graph signal is based on balancing a small TV \( \|\tilde{x}[\cdot]\|_E \) with a small empirical error \( \tilde{E}(\hat{x}[\cdot]) \), i.e.,

\[
\hat{x}[\cdot] \in \arg \min_{\tilde{x}[\cdot] \in \mathbb{R}^N} \tilde{E}(\tilde{x}[\cdot]) + \lambda \|\tilde{x}[\cdot]\|_E. \tag{6}
\]

The recovery problem (6) is a convex optimization problem [13]. Indeed, the objective function of (6) is the sum of a convex quadratic function (5) and a semi-norm \( \lambda \|\tilde{x}[\cdot]\|_E \) (which is convex) [13]. The regularization parameter \( \lambda \) in the recovery problem (6) allows to trade-off a small empirical error against a small TV of the recovered graph signal. Our analysis of the accuracy achieved by the recovery method (6) will focus on a particular choice for \( \lambda \) (see Theorem 2 below).

\section{ADMM updates for nLasso}

By introducing, for each edge \( \{i,j\} \in E \), the auxiliary variables \( z_{ij}, z_{ji} \) (which act as local copies of \( \hat{x}[i], \hat{x}[j] \)), the recovery problem (6) can be rewritten as (cf. (5), (1))

\[
\begin{align*}
\min_{\{x[i]\}_{i \in V}} & \sum_{(i,j) \in E} (\hat{x}[i] - y[i])^2 + \lambda \sum_{(i,j) \in E} W_{i,j} |z_{ij} - z_{ji}| \\
\text{s.t.} & \quad x[i] = z_{ij}, \quad \text{for } i \in V, \quad j \in N(i). \tag{7}
\end{align*}
\]

We solve the convex optimization (7) using the alternating direction method of multipliers (ADMM) [14]. The use of ADMM is appealing since it naturally lends to a distributed implementation in the form of message passing over the underlying data graph \( G \). We have summarized this message passing formulation of the ADMM iterations in Alg. 1.

In particular, ADMM is an iterative method generating a sequence \( x^{(k)}[i], k = 0, 1, \ldots \) of signal estimates which eventually converge to a solution of (7) [14]. For a detailed derivation of the update rules we refer to [8]. In particular, for each node \( i \in V \), ADMM amounts to computing the update \( x^{(k+1)}[i] \) by solving

\[
x^{(k+1)}[i] = \arg \min_{\tilde{x}[i]} \left( I_M[i](\hat{x}[i] - y[i])^2 + (\rho/2) \sum_{j \in N(i)} (\hat{x}[i] - z_{ij}^{(k)} + u_{ij}^{(k)})^2 \right). \tag{8}
\]

Solving (8) for a sampled node \( i \notin M \), yields

\[
x^{(k+1)}[i] = (1/|N(i)|) \sum_{j \in N(i)} (z_{ij}^{(k)} - u_{ij}^{(k)}), \tag{9}
\]

whereas for a non-sampled node \( i \in M \),

\[
x^{(k+1)}[i] = \frac{2}{\rho |N(i)| + 2} \left( y[i] + \frac{\rho}{2} \sum_{j \in N(i)} (z_{ij}^{(k)} - u_{ij}^{(k)}) \right). \tag{10}
\]

As the notation in (6) suggests, there might be several solutions \( \hat{x} \) which achieve the minimum objective value. However, our analysis applies to any of those optimal solutions so that this non-uniqueness poses no difficulty for our approach. The tuning parameter \( \lambda \) in (6) trades off a small empirical error \( \tilde{E}(\hat{x}[\cdot]) \) (cf. (5)) against a small total variation \( \|\hat{x}[\cdot]\|_E \) (cf. (1)) of the recovered signal \( \hat{x}[\cdot] \). A small value of \( \lambda \) enforces the solutions of (6) to achieve a small empirical error, whereas, a large value of \( \lambda \) enforces the solutions of (6) to have a small TV.

The recovery problem (6) is a convex problem and can be approached by modern convex optimization methods [11, 15, 16]. These optimization methods take advantage of a particular structure which is presented in the nLasso problem (6). In particular, the nLasso amounts to minimizing the sum of two functions, each of which could be minimized easily when considered isolated.
Algorithm 1 nLasso via ADMM

Input: data graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, sampling set $\mathcal{M}$, and signal samples $\{y[i]\}_{i \in \mathcal{M}}$

Initialize: for all nodes $i \in \mathcal{V}$ choose $x^{(0)}[i]$ i.i.d. uniformly from $\{1, \ldots, 5\}$; for all edges $i,j \in \mathcal{E}$ set $n_{ij}^{(0)} = 0, z_{ij}^{(0)} = 0$; iteration counter $k = 0$

repeat
  for each node $i \in \mathcal{V}$ do
    update $x^{(k+1)}[i]$ by
      
      $\begin{cases} (10), & i \in \mathcal{M} \\ (9), & \text{otherwise} \end{cases}$
  end for
  for each edge $i,j \in \mathcal{E}$ do
    $\theta = \max(0.5, 1 - \frac{\lambda W_{ij}}{\rho ||x^{(k+1)}[i] + u^{(k)}_{ij}||_2 - ||y[i]||_2})$
    $z_{ij}^{(k+1)} = \theta(x^{(k+1)}[i] + u^{(k)}_{ij}) + (1 - \theta)(x^{(k+1)}[j] + u^{(k)}_{ji})$
    $u^{(k+1)}_{ij} = u^{(k)}_{ij} + (x^{(k+1)}[i] - z_{ij}^{(k+1)})$
    $u^{(k+1)}_{ji} = u^{(k)}_{ji} + (x^{(k+1)}[j] - z_{ji}^{(k+1)})$
  end for
  $k = k + 1$
until convergence

Output: $x^{(k)}[\cdot]$

3. WHEN IS NETWORK LASSO ACCURATE?

Our main contribution is to show that the NCC guarantees any solutions of (6) allows to accurately recover the true underlying graph signal.

**Definition 1.** Consider a data graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ with a particular partition $\mathcal{F}$ of its nodes $\mathcal{V}$. A sampling set $\mathcal{M} \subseteq \mathcal{V}$ is said to satisfy NCC with constants $K, L > 0$, if

$$L ||z[\cdot]||_{\mathcal{F}} \leq K \sqrt{\sum_{i \in \mathcal{M}} (z[i])^2 + ||z[\cdot]||_{\mathcal{F}}}$$

for any graph signal $z[\cdot] \in \mathbb{R}^N$.

It turns out that, if the sampling set satisfies the NCC, any solution of (6) provides an accurate estimate of the true underlying graph signal (2).

**Theorem 2.** Consider a data set with data graph $\mathcal{G}$ and a graph signal $x[\cdot]$ of the form (2) with underlying partition $\mathcal{F}$. We observe noisy signal value on the sampling set $\mathcal{M}$. The sampling set $\mathcal{M}$ is chosen such that it satisfies NCC with parameters $L > 4$ and $K > 0$. The noise is sufficiently small such that (4) holds with some noise level $\nu < 1/(4K)$. Then any solution $\hat{x}[\cdot]$ of nLasso (6) with $\lambda := 1/K$, satisfies

$$||\hat{x}[\cdot] - x[\cdot]||_2 \leq 12K.$$  

**Proof of Theorem 2.** Consider an arbitrary solution $\hat{x}[\cdot]$ of (6) and denote the difference between $\hat{x}[\cdot]$ and the true underlying clustered signal $x[\cdot]$ as $\hat{x}[\cdot] := \hat{x}[\cdot] - x[\cdot]$, which we assume to be non-constant, i.e., $||\hat{x}[\cdot]||_2 \neq 0$. By (6),

$$\sum_{i \in \mathcal{M}} (\hat{x}[i] - y[i])^2 + \lambda ||\hat{x}[\cdot]||_2 \leq \sum_{i \in \mathcal{M}} \epsilon[i]^2 + \lambda ||x[\cdot]||_2.$$  

Using the decomposition property for the semi-norm $||\cdot||_2$, we have

$$\sum_{i \in \mathcal{M}} (\hat{x}[i] - y[i])^2 + \lambda ||\hat{x}[\cdot]||_{\mathcal{F}} \leq \sum_{i \in \mathcal{M}} \epsilon[i]^2 + \lambda(\|x[\cdot]\|_{\mathcal{F}} - \|\hat{x}[\cdot]\|_{\mathcal{F}}).$$

Since $x[\cdot]$ is assumed clustered (cf. (2)),

$$\|x[\cdot]\|_{\mathcal{F}} = 0, \text{ and } \|\hat{x}[\cdot]\|_{\mathcal{F}} = \|\hat{x}[\cdot]\|_{\mathcal{F}}.$$  

Thus, inserting (15) into (14), by the triangle inequality

$$\sum_{i \in \mathcal{M}} (\hat{x}[i] - y[i])^2 + \lambda ||\hat{x}[\cdot]||_{\mathcal{F}} \leq \sum_{i \in \mathcal{M}} \epsilon[i]^2 + \lambda ||\hat{x}[\cdot]||_{\mathcal{F}}.$$  

Combining (16) with

$$\sum_{i \in \mathcal{M}} (\hat{x}[i] - y[i])^2 \leq \sum_{i \in \mathcal{M}} (\hat{x}[i] - \epsilon[i])^2$$

and

$$\sum_{i \in \mathcal{M}} \epsilon[i]^2 = 2 \sum_{i \in \mathcal{M}} \hat{x}[i] \epsilon[i]$$

yields

$$\sum_{i \in \mathcal{M}} \hat{x}[i]^2 + \lambda ||\hat{x}[\cdot]||_{\mathcal{F}} \leq 2\nu ||\hat{x}[\cdot]||_2 + \lambda ||\hat{x}[\cdot]||_{\mathcal{F}}.$$  

Thus, since $\lambda = 1/K$ and $\nu < 1/(4K)$, inequality (18) implies that the difference signal $\hat{x}[\cdot]$ is approximately clustered according to $\mathcal{F}$, i.e.,

$$||\hat{x}[\cdot]||_{\mathcal{F}} \leq 3||\hat{x}[\cdot]||_{\mathcal{F}}.$$  

Since $\mathcal{M}$ satisfies NCC with $L > 4$, (11) and (19) yield

$$(1/K)^2 \|\hat{x}[\cdot]\|_{\mathcal{F}}^2 \leq \sum_{i \in \mathcal{M}} \hat{x}[i]^2.$$  

Combining (20) with (18) yields (12).  

4. NUMERICAL RESULTS

We now present numerical experiments to illustrate our theoretical results in Section 3. These experiments revolve around a synthetic dataset which is generated using the Barabási-Albert (BA) model [17]. The resulting data graph is partitioned into 5 disjoint clusters $\mathcal{F} = \{C_1, \ldots, C_5\}$, each cluster consisting of 100 nodes.

In particular, starting from an initially connected network of 20 nodes, new nodes are added to the network one at a time and are connected to three existing nodes until obtaining a cluster of 100 nodes. The process is repeated 5 times in order to obtain 5 different clusters. These clusters are connected by randomly adding a total of 250 inter-cluster edges (see Fig. 52).
1). The edge weights of those inter-cluster edges \( \{i,j\} \in \partial F \) are set uniformly as \( W_{ij} = 2 \). In contrast, an edge connecting two nodes \( i,j \in C_k \) from the same cluster \( C_k \) is assigned the weight \( W_{ij} = 10k \).

We then generate a clustered graph signal according to (2) using the signal coefficients \( a_{C_k} = k \) (see Fig. 1). A noisy version of the signal values are observed at the nodes in the sampling set \( \mathcal{M} \) which is constructed by randomly selecting 50 nodes (which are shown in Fig. 2). In particular, the observed signal values are generated according to observation model (3) with the noise values \( \varepsilon[i] \) being i.i.d. zero-mean Gaussian random variables with variance \( \sigma^2 = 0.2 \), i.e., \( \varepsilon[i] \sim \mathcal{N}(0,0.2) \). In order to recover the entire graph signal from the noisy signal samples \( \{y[i]\}_{i \in \mathcal{M}} \), we run Alg. 1 to solve the recovery problem (6).

![Fig. 1](image1.png)

**Fig. 1.** A data graph obtained from the BA model which is partitioned into 5 clusters. The nodes belonging to the same cluster are coloured likely.

![Fig. 2](image2.png)

**Fig. 2.** The coloured nodes form the sampling set \( \mathcal{M} \).

In particular, we run Alg. 1 with a manually tuned parameter \( \lambda = 10^{-3} \) and \( \rho = 5 \cdot 10^{-2} \), for a fixed number of 200 iterations. The result is post-processed by rounding the recovered signal to the nearest integer and is plotted in Fig. 3. According to Fig. 3, Alg. 1 is able to perfectly recover the clustered-structure of the underlying graph signal. In Fig. 4, we plot the objective value as a function of iteration number which indicates that the Alg. 1 converges in less than 50 iterations. We also computed, using 1000 i.i.d. simulation runs, the normalized mean square error yielding a value of \( 9.99 \cdot 10^{-3} \) (standard deviation \( 6.4 \cdot 10^{-3} \)).

![Fig. 3](image3.png)

**Fig. 3.** The recovered graph signal obtained from Alg. 1 (each colour corresponds to one particular signal value).

![Fig. 4](image4.png)

**Fig. 4.** The objective value of the ADMM iterates in Algorithm 1.

5. REFERENCES


