Alexander Jung, Ayelet Heimowitz and Yonina C. Eldar

1Department of Computer Science, Aalto University, Finland; firstname.lastname(at)aalto.fi
2Department of Electrical Engineering, Technion, Haifa, Israel

ABSTRACT

We study compressed sensing of graph signals defined over complex networks. In particular, we propose and analyze a convex optimization method for recovering smooth graph signals from a small number of samples. Assuming the true underlying graph signal to be constant over well connected subset of nodes (clusters), we derive a sufficient condition on the sampling set and network structure such that the proposed convex method is accurate. This condition, which we coin the network nullspace property, characterizes which nodes of the graph should be sampled in order to retain the full information about the underlying graph signal.

Index Terms— compressed sensing, big data, semi-supervised learning, complex networks, convex optimization

I. INTRODUCTION

In many applications such as image processing, social networks or bioinformatics, the observed datasets carry an intrinsic network structure. Such datasets can be represented conveniently by signals defined over the underlying graph or network [19].

Processing of graph signals often rests on a smoothness hypothesis, e.g., as in (semi-)supervised learning [2], [4], requiring the signal to be nearly constant over well connected subset of nodes (clusters). Many convex optimization based approaches for recovering smooth graph signals from noisy observations have been proposed recently [11], [13], [14].

However, the characterization of conditions which guarantee accurate recovery is still in its infancy. The authors of [5], [16] derive sufficient conditions for recoverability, i.e., sampling theorems, for band-limited graph signals. The class of band-limited graph signals is spanned by a small number of eigenvectors of the graph Laplacian.

We present recovery conditions for clustered (“piece-wise constant”) graph signals defined over complex networks which are formulated directly in terms of network structure or topology. In particular, we show that if cluster boundaries are well-connected (in a sense made precise below) to the sampled nodes, then accurate recovery is possible by solving a convex optimization problem. Our second main contribution, beside stating a recovery condition, is the formulation of an efficient convex recovery method, i.e., the sparse label propagation algorithm, which is scalable to massive datasets.

The problem setup considered in this work is very similar to those of [20], [21], which provide sufficient conditions such that a variant of the LASSO method accurately recovers smooth graph signals from noisy observations. However, these works require access to fully labeled datasets, while we consider datasets which are only partially labeled.

Outline. We formalize the problem of recovering smooth graph signals from observing its values at few sampled nodes in Section II. In particular, we show how to formulate this recovery as a convex optimization problem which is equivalent to a linear program. Our main result, stated in Section III, is a sufficient condition on the network structure and sampling set such that accurate recovery is possible. Loosely speaking, this condition requires to sample nodes which are well-connected to the boundaries of clusters. The detailed derivation of our main result can be found in a follow-up journal paper.

Notation. We denote non-negative real numbers as $\mathbb{R}_+$. Vectors and matrices are denoted by boldface lower-case and upper-case letters, respectively. The vector with all entries equal to one (zero) is denoted $\mathbf{1}$ ($\mathbf{0}$). For a matrix $A$, we write its transpose and nullspace as $A^T$ and $\mathcal{N}(A):=\{x: Ax=0\}$. The $\ell_1$, $\ell_2$ and $\ell_\infty$ norm of a vector $u=(u_1,...,u_N)^T$ are $\|u\|_1:=\sum_i |u_i|$, $\|u\|_2:=\sqrt{\sum_i |u_i|^2}$ and $\|u\|_\infty:=\max_i |u_i|$, respectively. For a vector $x\in\mathbb{R}^N$ and index set $B\subseteq\{1,...,N\}$, we denote by $x_B\in\mathbb{R}^N$ the vector obtained by zeroing all entries $x$ whose indices are not in $B$. We denote the support of a vector $x\in\mathbb{R}^N$ as $\text{supp}(x):=\{i: x_i \neq 0\}$. The spectral norm of a matrix $D$ is denoted $\|D\|_2:=\max_{\|x\|_2=1} \|Dx\|_2$.

A weighted undirected graph is denoted $G=(\mathcal{V},\mathcal{E},W)$ with nodes $\mathcal{V}=\{1,...,N\}$ which are connected by undirected edges $\{i,j\}\in\mathcal{E}$ with weights $W_{i,j}\in\mathbb{R}_+$. Given a subset $C\subseteq\mathcal{V}$ of nodes, its boundary $\partial C\subseteq\mathcal{E}$ is made up of all edges $\{i,j\}\in\mathcal{E}$ with $i\in C$ and $j\in\mathcal{V}\setminus C$. Given a subset $S\subseteq\mathcal{E}$ of edges, by a slight abuse of notation, we use the same symbol $\mathcal{S}$ to indicate the subset of nodes connected by the edges in $\mathcal{S}$. The neighbourhood of a node $i\in\mathcal{V}$ is denoted $\mathcal{N}(i):=\{j\in\mathcal{V}: \{i,j\}\in\mathcal{E}\}$. 

THE NETWORK NULLSPACE PROPERTY FOR COMPRESSED SENSING OVER NETWORKS
II. PROBLEM FORMULATION

We begin by discussing how graph signals arise when performing semi-supervised learning from massive heterogeneous datasets, i.e., big data over networks. In particular, we show that the learning problem can be formulated as a graph signal recovery problem. The recovery is based on the availability of a few labeled data points, which amounts to sampling the graph signal at the corresponding nodes.

II-A. Graph Signals for Learning from Big Data

Consider a massive heterogeneous dataset comprised of different types of data, e.g., mixtures of audio, video and text data. Moreover, datasets typically contain mostly unlabeled data points; only a small fraction is labeled data. An efficient strategy to handle such datasets is to organize them as a networks or graph $G = (V, E)$ whose nodes $V$ represent individual data points. For example, the node $i \in V$ might represent a (super-)pixel in image processing, a neuron of a neural network [10] or a social network user profile [6].

Many applications naturally suggest a notion of similarity between individual data points, e.g., the profiles of befriended social network users or greyscale values of neighbouring image pixels. These domain-specific notions of similarity are represented by the edges of the graph $G$, i.e., the nodes $i, j \in V$ representing similar data points are connected by an undirected edge $\{i, j\} \in E$. In some applications it is possible to quantify the extent to which data points are similar (e.g., via the physical distance in a sensor network application). Formally, given two similar data points $i, j \in V$ we quantify the strength of their connection by the non-negative edge weight $W_{i,j} \geq 0$ which we collect in the symmetric weight matrix $W \in \mathbb{R}^{N \times N}$.

In order to characterize the structure of a given graph, we need the notion of the strength (degree) of a node $i \in V$, defined as the overall weight of connections to its neighbours $N(i) := \{j \in V : \{i, j\} \in E\}$, i.e.,

$$d_i := \sum_{j \in N(i)} W_{i,j}. \quad (1)$$

In what follows, we only consider connected simple graphs $G$, i.e., we have $\{i, j\} \notin E$ and $d_i > 0$ for any node $i \in V$. The maximum node degree of $G$ is denoted $d_{\text{max}} := \max_{i \in V} d_i$.

We orient the graph $G$ by declaring for each edge $e = \{i, j\}$ one node as the head $e^+$ and the other node as the tail $e^-$. Given an edge set $S$ in the undirected graph $G$, we denote the set of directed edges obtained by orienting $G$ as $\overrightarrow{S}$. For a graph $G$ and a particular orientation, we define the weighted incidence matrix $D \in \mathbb{R}^{E \times |V|}$ [20]

$$D_{e,i} := \begin{cases} W_{i,j} & \text{if } i = e^+ \\ -W_{i,j} & \text{if } i = e^- \\ 0 & \text{else} \end{cases} \quad (2)$$

If we number the nodes and orient the edges in the chain graph in Fig. 1-(a) from left to right its weighted incidence matrix is $D = \begin{pmatrix} W_{1,2} & -W_{1,2} & 0 \\ 0 & W_{2,3} & -W_{2,3} \end{pmatrix}$. Note that

$$\|Dx\|_2^2 \leq 2 \sum_{\{i,j\} \in E} W_{i,j}^2 \sum_{i = 1}^{|V|} x_i^2 \leq 2 \sum_{i = 1}^{|V|} d_i^2 \|x_i\|_2^2, \quad (3)$$

where step (a) is due to $(x+y)^2 \leq 2(x^2 + y^2)$ and step (b) is due to $\sum_{j \in N(i)} W_{i,j}^2 \leq \left( \sum_{j \in N(i)} W_{i,j} \right)^2$. Thus,

$$\|D\|_2 \leq \max_{\|x\|_2 \neq 0} \|Dx\|_2 \leq \sqrt{2 \sum_{i \in V} d_i^2 \|x_i\|_2^2} \leq \sqrt{2} d_{\text{max}}. \quad (4)$$

The bound (4) will be used for analyzing the convergence of the recovery algorithm presented in Section II-B.

Beside the graph structure $G$, a dataset also conveys label information which induces a graph signal defined over $G$. We define a graph signal $x$ over the graph $G = (V, E)$ as a mapping $V \to \mathbb{R}$, which associates (labels) every node $i \in V$ with the signal value $x[i] \in \mathbb{R}$. For a supervised machine learning application, the signal values $x[i]$ might represent class membership in a classification problem or the target (output) value in a regression problem. The space of all graph signals, which is also known as the vertex space (cf. [7]), is denoted by $\mathbb{R}^V$.

II-B. Graph Signal Recovery

We aim at recovering a graph signal $x \in \mathbb{R}^V$ defined over the empirical graph $G$, from observing its values $\{x[i]\}_{i \in \mathcal{M}}$ on a (small) sampling set

$$\mathcal{M} := \{i_1, \ldots, i_M\} \subseteq V, \quad (5)$$

where typically $M \ll N$.

Our approach to graph signal recovery is based on a smoothness assumption, which is somewhat similar to the
Algorithm 1 Sparse Label Propagation

**Input:** incidence matrix $D \in \mathbb{R}^{E \times V}$ of the graph $G$, sampling set $\mathcal{M}$, initial labels $\{x[i]\}_{i \in \mathcal{M}}$.

**Initialize:** $k := 0, z^{(0)} := 0, x^{(0)} := x, \hat{x}^{(0)} := 0, y^{(0)} := 0$

repeat

1. $y^{(k+1)} := T(y^{(k)}) + (1/(2d_{\text{max}}))Dz^{(k)}$
2. $r := x^{(k)} - (1/(2d_{\text{max}}))D^T y^{(k+1)}$
3. $x^{(k+1)} := \begin{cases} x[i] & \text{for } i \in \mathcal{M} \\ r[i] & \text{else} \end{cases}$
4. $z^{(k+1)} := 2x^{(k+1)} - x^{(k)}$
5. $x^{(k+1)} := \hat{x}^{(k)} + x^{(k+1)}$
6. $k := k + 1$

until stopping criterion is satisfied

**Output:** $\hat{x}^{(k)} := (1/k)\hat{x}^{(k)}$

Algorithm 1 we use the clipping operator $T : \mathbb{R}^E \rightarrow \mathbb{R}^E$ for edge signals, defined elementwise as $(T(y))[e] = (1/\max\{|y[e]|, 1\})|y[e]|$. There are various options for the stopping criterion in Algorithm 1, e.g., using a fixed number of iterations or sufficient decrease of the objective function (cf. [11]).

We highlight the fact that Algorithm 1 can be easily converted into a scalable message passing algorithm. Indeed, the application of the graph incidence matrix $D$ and its transpose $D^T$ amounts to local updates involving only the neighbourhoods of a particular node or edge in the graph $G$. In particular, it is not required that each node has full knowledge of the entire graph. This characteristic makes Algorithm 1 scalable to massive datasets.

Combining [3, Thm. 1] with the bound (4), the output $\hat{x}^{(k)}$ of Algorithm 1 after $k$ iterations satisfies

$$\|\hat{x}^{(k)}\|_{TV} - \|\hat{x}\|_{TV} \leq c_1/k,$$

with $c_1$ depending on the underlying graph signal $x$. The convergence rate is optimal for first-order gradient methods applied to problems of the type (8) (cf. [12]).

**III. WHEN IS RECOVERY ACCURATE?**

For Algorithm 1 to be accurate, we have to verify that the solutions $\hat{x}$ of (8) are close to the true underlying graph signal $x \in \mathbb{R}^V$. In what follows, we present a condition which guarantees any solution $\hat{x}$ of (8) to be close to $x$. To this end, we introduce the notion of a circulation with demands [15].

**Definition 1.** Consider an empirical graph $G$ with an arbitrary but fixed orientation. A circulation with demands $d[i] \in \mathbb{R}$, for $i \in \mathcal{V}$, is a mapping $f[\cdot] : \overrightarrow{E} \rightarrow \mathbb{R}_+$ satisfying

- the conservation law
  $$\sum_{j \in \mathcal{N}^+(i)} f[j,i] - \sum_{j \in \mathcal{N}^-(i)} f[i,j] = d[i], \text{ for any } i \in \mathcal{V}$$
- the capacity constraints
  $$f[e] \leq W_e \text{ for any oriented edge } e \in \overrightarrow{E}.$$

Here, we used the directed neighbourhood $\mathcal{N}^+(i) := \{j \in \mathcal{N}(i) : (i,j) \in E\}$ and analogous $\mathcal{N}^-(i)$.

Using the notion of circulations with demands, we now rephrase the nullspace property of compressed sensing [8], [9] for signals defined over networks.

**Definition 2.** Consider a partition $\mathcal{F} = \{C_1, \ldots, C_{|\mathcal{F}|}\}$ of pairwise disjoint subsets of nodes (clusters) $C_i \subseteq \mathcal{V}$ and a set of sampled nodes $\mathcal{M} \subseteq \mathcal{V}$. The sampling set $\mathcal{M}$ satisfies the network nullspace property w.r.t. $\mathcal{F}$, denoted $\text{NNSP}((\mathcal{M},\mathcal{F}))$, if for any orientation of the edges in the boundaries $B := \bigcup_i \partial C_i$, we can orient the remaining edges in $E \setminus B$ such
Theorem 4. Consider a clustered graph signal $x_c \in \mathcal{X}$ (cf. (13)) which is observed only at the sampling set $M \subseteq V$. If $\text{NNSP}(\mathcal{M}, \mathcal{F})$ holds, then the unique solution of (8) coincides with $x_c$.

Thus, if the sampling set $\mathcal{M}$ is chosen such that $\text{NNSP}(\mathcal{M}, \mathcal{F})$ holds, then recovery algorithms based on solving (8), in particular Algorithm 1 can be expected to accurately recover graph signals $x$ of the form (13).

Since Algorithm 1 does not require knowledge of the partition $\mathcal{F} = \{ C_1, \ldots, C_{|\mathcal{F}|}\}$, it could be used for determining the clusters $C_i$, by assigning all nodes having the same graph signal value to the same cluster. This, of course, presupposes that the true underlying graph signal has different values on different clusters.

The scope of Theorem 4 is somewhat limited as it applies only to graph signals which are precisely of the form (13). We now state a more general result applying to any graph signal $x \in \mathbb{R}^V$.

**Theorem 5.** Consider a graph signal $x \in \mathbb{R}^V$ which is observed only at the sampling set $\mathcal{M}$. If $\text{NNSP}(\mathcal{M}, \mathcal{F})$ holds, then any solution $\hat{x}$ of (8) satisfies (cf. (13))

$$\|D(\hat{x} - x)\|_1 \leq 6d_{\max} \min_{x_c \in \mathcal{X}} \|x - x_c\|_1.$$  (14)

Thus, as long as the underlying graph signal $x$ can be well approximated by a clustered signal of the form (13), any solution $x_c$ of (8) accurately indicates edges $e \in \mathcal{E}$ over which the true signal $x$ changes significantly, i.e., for which the signal difference $\|D(x)\|_e$ is large.

**IV. FORE-/BACKGROUND SEGMENTATION**

We applied Algorithm 1 to the problem of fore-background (FB) segmentation [18]. Here, we consider images of varying size (around $512 \times 512$ pixels), with a particular pixel $i$ represented by the RGB vector $c[i] = (\text{red}[i], \text{green}[i], \text{blue}[i]) \in \{0, 1, \ldots, 255\}^3$. We associate a grid graph $\mathcal{G}$ shown in Fig. 1-(b) with a particular image. Each node represents an individual pixel and is connected to up to eight surrounding pixels (cf. Fig. 1-(b)). For two connected nodes $i$ and $j$, we use the edge weight $W_{ij} := \exp(-1/\sigma)\|c[i] - c[j]\|$ with $\sigma := \text{median}\{|c[i] - c[j]|\}_{i,j \in \mathcal{E}}$.

The problem of FB segmentation amounts to determining for each pixel $i$ the degree or our confidence of pixel $i$ belonging to the foreground or not. By representing this degree of node $i$ belonging to foreground as a real number $x[i]$, with $x[i] = 1$ if we are certain that it belongs to foreground and $x[i] = -1$ if we are certain that it belongs to background, we obtain a graph signal. The “grabCut” dataset [18] contains several images for which the values $x[i]$ are known for certain regions. In particular, images are divided into three regions, whose boundaries are indicated by white contours (cf. Fig. 2). Each region corresponds to a subset $\mathcal{R}_t$, for $t \in \{1, 2, 3\}$ of graph nodes. The outer (inner) region $\mathcal{R}_1 (\mathcal{R}_3)$ has been manually verified to be background (foreground), i.e., we have initial labels $x[i] = -1$ for $i \in \mathcal{R}_1$ ($x[i] = 1$ for $i \in \mathcal{R}_3$). For the nodes in the middle region $\mathcal{R}_2$, we do not know if they belong to background or not.

To find the values $x[i]$ for $i \in \mathcal{R}_2$, we run Algorithm 1 with initial labels $x[i] \in \{-1, 1\}$ for nodes $i \in \mathcal{M}$ in the sampling set $\mathcal{M} = \cup \mathcal{R}_1 \setminus \mathcal{R}_3$. We used a fixed number of 1000 iterations resulting in the recovered signal $\hat{s} = \hat{x}^{(1000)}$. The signs of the recovered graph signal values $\hat{s}[i]$ are then used for deciding if node $i \in \mathcal{R}_2$ belongs to foreground ($\hat{s}[i] > 0$) or not ($\hat{s}[i] \leq 0$). The obtained results are shown in Fig. 2.
Fig. 2. FB segmentation for two images of the “grabcut” dataset [18]. (a) Original images containing the boundaries between labeled ($R_1$, $R_3$) and unlabeled regions ($R_2$). (b) Extracted foreground obtained from the signs of the output of Algorithm 1 after 1000 iterations.

V. REFERENCES