COORDINATE DESCENT ACCELERATIONS FOR SIGNAL RECOVERY ON SCALE-FREE GRAPHS BASED ON TOTAL VARIATION MINIMIZATION

Peter Berger, Gabor Hannak, and Gerald Matz

Institute of Telecommunications, TU Wien (Vienna, Austria)
Email: `firstname.lastname@nt.tuwien.ac.at`

ABSTRACT

We extend our previous work on learning smooth graph signals from a small number of noisy signal samples. Minimizing the signal's total variation amounts to a non-smooth convex optimization problem. We propose to solve this problem using a combination of Nesterov's smoothing technique and accelerated coordinate descent. The resulting algorithm converges substantially faster, specifically for graphs with vastly varying node degrees (e.g., scale-free graphs).

I. INTRODUCTION

Background. Graphs are flexible and powerful models for many massive data sets (e.g., sensor networks and online social networks) [1]–[4]. In graph signal processing (GSP), each graph node is associated with a data point and the graph edges reflect data dependencies or similarity relations. Applications of GSP include social networks [5], [6], news sites and blog spaces [7], [8], and proteomics [9], [10].

In this paper we consider the problem of recovering a graph signal from noisy samples taken on a small subset of graph nodes. This problem is also referred to as semi-supervised learning [11] or inpainting [12] on graphs. Graph signal recovery requires some kind of smoothness, which can be quantified in terms of a generalized notion of band-limitation [13], Tikhonov regularization [11], graph variation [12], and the graph total variation (TV) [14]. Building on TV, we formulate graph signal recovery as a non-smooth convex optimization problem. In our previous work, we solved this problem using a combination of ADMM with denoising [15], a primal-dual algorithm [16], and Nesterov's method [17] (cf. [18]). An augmented ADMM algorithm for this problem was recently proposed in [19].

Contributions. In this paper, we propose an efficient graph signal learning algorithm that combines Nesterov's smoothing technique [20] with accelerated coordinate descent [21], [22]. Numerical comparisons with FISTA [23], [24] and with the augmented ADMM algorithm [19] confirm that our new algorithm excels in terms of convergence for strongly irregular graphs, i.e., graphs having vastly differing node degrees.

Funding by WWTF Grant ICT15-119. Part of this work was performed while the first author was affiliated with Aalto University (Finland).

II. PROBLEM FORMULATION

We consider a weighted directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ with node set $\mathcal{V} = \{1, \ldots, N\}$, edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and nonnegative weight matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$. We have $W_{i,j} = 0$ unless there is an edge from node i to node j; the larger the weight $W_{i,j}$, the stronger the connection from node i to j.

A graph signal is a mapping that associates to each node $i \in \mathcal{V}$ a real value x_i. We can represent a graph signal by a vector $\mathbf{x} \triangleq (x_1, \ldots, x_N)^T \in \mathbb{R}^N$. Assume we only have access to noisy signal samples on the node set $\{1, \ldots, M\}$ (this can always be achieved by relabeling the nodes). Our noisy sampling model can thus be expressed as

$$y_i = x_i + n_i, \quad i = 1, \ldots, M.$$

The additive noise n_i, $i = 1, \ldots, M$, subsumes measurement and modeling errors.

Our goal is to recover the full graph signal $\mathbf{x}$ from the observations y_i, $i = 1, \ldots, M$. To that end, the graph signal $\mathbf{x}$ is required to be smooth, i.e., to vary little between strongly connected nodes. We quantify the smoothness of $\mathbf{x}$ via its (anisotropic) TV, defined as [25]

$$\|\mathbf{x}\|_{\mathrm{TV}} = \sum_{i=1}^{N} \sum_{j=1}^{N} |x_j - x_i| \sqrt{W_{i,j}}. \tag{1}$$

With minor modifications, our results apply to the isotropic TV $\|\mathbf{x}\|_{\mathrm{TV}}^{I} = \sum_{i=1}^{N} \sqrt{\sum_{j=1}^{N} (x_j - x_i)^2 W_{i,j}}$ (cf. [25]). Our recovery method amounts to the optimization problem

$$\min_{\mathbf{x} \in \mathbb{R}^N} \; \frac{1}{2} \sum_{i=1}^{M} (y_i - x_i)^2 + \lambda \|\mathbf{x}\|_{\mathrm{TV}}, \tag{2}$$

in which the parameter $\lambda > 0$ balances the empirical error and the total variation of the recovered signal.

III. SMOOTH MINIMIZATION OF NON-SMOOTH FUNCTIONS

The problem in (2) is difficult since the TV is a convex but non-smooth (non-differentiable) function. To resolve this difficulty, we apply the smoothing technique from [20].

Let $\mathbf{B} : \mathcal{H}_1 \to \mathcal{H}_2$ be a linear operator from a finite-dimensional Hilbert space $\mathcal{H}_1$ to a finite-dimensional Hilbert space $\mathcal{H}_2$, both defined over the real numbers. Let $\mathcal{Q}_1 \subseteq \mathcal{H}_1$ be a closed convex set and let $\mathcal{Q}_2 \subset \mathcal{H}_2$ be a bounded, closed

ISBN 978-0-9928626-7-1 © EURASIP 2017

and convex set. Consider two continuous and convex functions $\hat{f}$ and $\hat{g}$ on $\mathcal{Q}_1$ and $\mathcal{Q}_2$, respectively. The function $\hat{f}$ is assumed differentiable with Lipschitz-continuous gradient $\nabla\hat{f}(\mathbf{x})$ with Lipschitz constant $L \geq 0$, i.e.,

$$\|\nabla\hat{f}(\mathbf{y}) - \nabla\hat{f}(\mathbf{x})\|_{\mathcal{H}_1} \leq L\|\mathbf{y} - \mathbf{x}\|_{\mathcal{H}_1}. \tag{3}$$

The method in [20] is designed for convex optimization problems of the form $\min_{\mathbf{x}\in\mathcal{Q}_1} f(\mathbf{x})$ with

$$f(\mathbf{x}) \triangleq \hat{f}(\mathbf{x}) + \max_{\mathbf{u}\in\mathcal{Q}_2}\{\langle\mathbf{u}, \mathbf{B}\mathbf{x}\rangle_{\mathcal{H}_2} - \hat{g}(\mathbf{u})\}. \tag{4}$$

Instead of $\min_{\mathbf{x}\in\mathcal{Q}_1} f(\mathbf{x})$, we solve the smooth problem $\min_{\mathbf{x}\in\mathcal{Q}_1} f_\mu(\mathbf{x})$ with the proxy $f_\mu(\mathbf{x}) \triangleq \hat{f}(\mathbf{x}) + h_\mu(\mathbf{x})$, where

$$h_\mu(\mathbf{x}) \triangleq \max_{\mathbf{u}\in\mathcal{Q}_2} \langle\mathbf{u}, \mathbf{B}\mathbf{x}\rangle_{\mathcal{H}_2} - \hat{g}(\mathbf{u}) - \frac{\mu}{2}\|\mathbf{u}\|^2_{\mathcal{H}_2} \tag{5}$$

and $\mu > 0$ is a smoothing parameter. It can be shown that

$$f_\mu(\mathbf{x}) \leq f(\mathbf{x}) \leq f_\mu(\mathbf{x}) + \frac{\mu}{2} \max_{\mathbf{u}\in\mathcal{Q}_2} \|\mathbf{u}\|^2_{\mathcal{H}_2},$$

and hence $f_\mu(\mathbf{x})$ is indeed a uniform smooth approximation of the objective function $f(\mathbf{x})$. The function $h_\mu(\mathbf{x})$ in (5) is differentiable with gradient $\nabla h_\mu(\mathbf{x}) = \mathbf{B}^*\mathbf{u}_\mu(\mathbf{x})$, where $\mathbf{u}_\mu(\mathbf{x})$ is the maximizer in (5). Furthermore, $\nabla h_\mu(\mathbf{x})$ is Lipschitz continuous with constant $\frac{1}{\mu}\|\mathbf{B}\|^2_{\text{op}}$ [20, Theorem 1]. Due to the Lipschitz continuity of $\nabla\hat{f}(\mathbf{x})$ (cf. (3)), the function $f_\mu(\mathbf{x})$ has a Lipschitz-continuous gradient $\nabla f_\mu(\mathbf{x}) = \nabla\hat{f}(\mathbf{x}) + \mathbf{B}^*\mathbf{u}_\mu(\mathbf{x})$ with Lipschitz constant

$$L_\mu \triangleq L + \frac{1}{\mu}\|\mathbf{B}\|^2_{\text{op}}. \tag{6}$$

IV. Smoothing the Graph Signal Recovery Problem

We will now cast the graph signal recovery problem (2) in the form (4). For this purpose, we introduce the local gradient vector $\nabla_i\mathbf{x}$ at node $i \in \mathcal{V}$ with elements

$$\big(\nabla_i\mathbf{x}\big)_j \triangleq (x_j - x_i)\sqrt{W_{i,j}}, \quad j = 1, \ldots, N.$$

The graph gradient is then given by

$$\nabla_{\mathcal{G}} : \mathbb{R}^N \to \mathbb{R}^{N\times N}, \quad \nabla_{\mathcal{G}}\mathbf{x} = (\nabla_1\mathbf{x}, \ldots, \nabla_N\mathbf{x})^T.$$

$\mathbb{R}^N$ and $\mathbb{R}^{N\times N}$ are Hilbert spaces with respective inner product $\langle\mathbf{x}, \mathbf{y}\rangle_2 \triangleq \sum_i x_i y_i$ and $\langle\mathbf{X}, \mathbf{Y}\rangle_{\text{F}} \triangleq \sum_{i,j} X_{i,j}Y_{i,j}$. The negative adjoint of $\nabla_{\mathcal{G}}$ equals the divergence operator, $\operatorname{div}_{\mathcal{G}} = -\nabla^*_{\mathcal{G}}$, which maps a matrix $\mathbf{P}$ to a vector $\operatorname{div}_{\mathcal{G}}\mathbf{P}$ with entries (cf. [14])

$$(\operatorname{div}_{\mathcal{G}}\mathbf{P})_i \triangleq \sum_{j\in\mathcal{V}} \sqrt{W_{i,j}}P_{i,j} - \sqrt{W_{j,i}}P_{j,i}, \quad i = 1, \ldots, N.$$

The TV term in (1) can now be written as $\lambda\|\mathbf{x}\|_{\text{TV}} = \max_{\mathbf{P}\in\mathcal{P}_\lambda} \langle\mathbf{P}, \nabla_{\mathcal{G}}\mathbf{x}\rangle_{\text{F}}$ with the closed convex set

$$\mathcal{P}_\lambda \triangleq \{\mathbf{P} : |P_{i,j}| \leq \lambda, \quad 1 \leq i, j \leq N\}.$$

Furthermore, (2) can be reformulated as

$$\min_{\mathbf{x}\in\mathbb{R}^N} \frac{1}{2}\sum_{i=1}^{M}(y_i - x_i)^2 + \max_{\mathbf{P}\in\mathcal{P}_\lambda} \langle\mathbf{P}, \nabla_{\mathcal{G}}\mathbf{x}\rangle_{\text{F}}. \tag{7}$$

The optimization problem (7) is of the form (4) with $\mathbf{B} = \nabla_{\mathcal{G}}$, $\hat{f}(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{M}(y_i - x_i)^2$ with Lipschitz constant $L = 1$, $\mathbf{u} = \mathbf{P}$, $\hat{g}(\mathbf{u}) \equiv 0$, $\mathcal{Q}_1 = \mathbb{R}^N$, and $\mathcal{Q}_2 = \mathcal{P}_\lambda$. The smoothed version of (7) is given by $\min_{\mathbf{x}\in\mathbb{R}^N} f_\mu(\mathbf{x})$ with

$$f_\mu(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{M}(y_i - x_i)^2 + \max_{\mathbf{P}\in\mathcal{P}_\lambda} \Big(\langle\mathbf{P}, \nabla_{\mathcal{G}}\mathbf{x}\rangle_{\text{F}} - \frac{\mu}{2}\|\mathbf{P}\|^2_{\text{F}}\Big). \tag{8}$$

The gradient $\nabla f_\mu(\mathbf{x})$ is obtained as

$$\nabla f_\mu(\mathbf{x}) = \mathbf{z} - \operatorname{div}_{\mathcal{G}}\mathbf{P}_\mu(\mathbf{x}) \tag{9}$$

with $\mathbf{z} = (x_1 - y_1, \ldots, x_M - y_M, 0, \ldots, 0)^T$ and

$$\begin{aligned}\mathbf{P}_\mu(\mathbf{x}) &= \operatorname*{arg\,max}_{\mathbf{P}\in\mathcal{P}_\lambda} \Big(\langle\mathbf{P}, \nabla_{\mathcal{G}}\mathbf{x}\rangle_{\text{F}} - \frac{\mu}{2}\|\mathbf{P}\|^2_{\text{F}}\Big) \\ &= \operatorname*{arg\,min}_{\mathbf{P}\in\mathcal{P}_\lambda} \Big\|\mathbf{P} - \frac{1}{\mu}\nabla_{\mathcal{G}}\mathbf{x}\Big\|^2_{\text{F}}\end{aligned}$$

is the orthogonal projection of $\frac{1}{\mu}\nabla_{\mathcal{G}}\mathbf{x}$ onto $\mathcal{P}_\lambda$, which is given by the element-wise clipping

$$\big(\mathbf{P}_\mu(\mathbf{x})\big)_{i,j} = \begin{cases}(x_j - x_i)\frac{\sqrt{W_{i,j}}}{\mu}, & \text{if } |x_j - x_i|\sqrt{W_{i,j}} \leq \mu\lambda, \\ \lambda\operatorname{sgn}(x_j - x_i), & \text{else.}\end{cases}$$

According to (6), the Lipschitz constant for $\nabla f_\mu(\mathbf{x})$ equals

$$L_\mu = 1 + \frac{1}{\mu}\|\nabla_{\mathcal{G}}\|^2_{\text{op}} = 1 + \frac{1}{\mu}\|\operatorname{div}_{\mathcal{G}}\|^2_{\text{op}}. \tag{10}$$

The operator norm in this expression (and hence the Lipschitz constant) is essentially determined by the maximum (weighted) node degree [17].

V. Reconstruction Algorithm

When applying classical gradient-based algorithms (e.g., [20], [24], [26]) to minimize the convex, continuously differentiable proxy $f_\mu(\mathbf{x})$ in (8), the step size is determined by the reciprocal of the Lipschitz constant L_μ in (10). When this Lipschitz constant is large (which happens if there are nodes with large weighted degree), the convergence of the algorithms becomes slow. To mitigate this problem, we propose to apply the accelerated coordinate descent (ACD) method from [21]. Here, the convergence is determined by the distinct (and hopefully smaller) coordinate-wise Lipschitz constants $L_1, L_2, \ldots, L_N$ defined by

$$|\nabla_i f_\mu(\mathbf{x} + s\mathbf{e}_i) - \nabla_i f_\mu(\mathbf{x})| \leq L_i|s|.$$

Here, $\mathbf{e}_i \in \mathbb{R}^N$ denotes the ith standard unit vector and the inequality must hold for all $\mathbf{x} \in \mathbb{R}^N$ and $s \in \mathbb{R}$.

For this scenario, the ACD method from [21] achieves an asymptotic rate of convergence of $\mathcal{O}(\frac{1}{k^2})$ in the function value. For each iteration of this ACD method, only one

ISBN 978-0-9928626-7-1 © EURASIP 2017

element (coordinate) of the gradient needs to be computed but a relatively expensive summation of two N-dimensional vectors is required. The latter drawback has been resolved in [22] via a change of variables. Using the results from Section IV for the gradient of $f_\mu(\mathbf{x})$ we obtain Algorithm 1 as an adaptation of the efficient implementation [22] of the ACD method to our (smoothed) signal recovery problem. This algorithm uses in-place computations and (for better clarity) the child set $\mathrm{ch}(i) \triangleq \{j \in \mathcal{V} : W_{i,j} > 0\}$ and the parent set $\mathrm{pa}(i) \triangleq \{j \in \mathcal{V} : W_{j,i} > 0\}$ of node i. Moreover, it uses the following coordinate-wise Lipschitz constants for the gradient $\nabla f_\mu(\mathbf{x})$ (see the Appendix for a proof).

Lemma 1. *The coordinate-wise Lipschitz constants L_i, $i = 1, \ldots, N$, of the gradient $\nabla f_\mu(\mathbf{x})$ in* (9) *are given by*

$$L_i = 1 + \frac{2}{\mu} d_i, \quad d_i \triangleq \sum_{j=1}^{N} (W_{i,j} + W_{j,i}). \tag{11}$$

This result says that the Lipschitz constant for the ith coordinate is essentially determined by the degree d_i of node i. In many graphs (e.g., scale-free), the majority of nodes has small degree and only few nodes have large degrees. For those types of graphs, we expect our ACD algorithm to converge particularly fast.

VI. Numerical Experiments

We next assess the performance and the convergence speed of Algorithm 1 using a clustered scale-free graph that is a good model e.g. for online social networks that consist of multiple communities (clusters) with opinion-leaders (hubs) in each community. All edges in the graph are undirected and unweighted, i.e., we ensure $W_{i,j} = W_{j,i} \in \{0, 1\}$.

Graph (Signal) Construction. Our iterative construction was initialized with a graph with $n = 10$ nodes, grouped into 5 disjoint subgraphs, each consisting of two nodes connected by an edge. The graph signal was obtained by assigning values from the alphabet $\{1, 2, 3, 4, 5\}$ to the nodes such that the values are identical within each subgraph but distinct for different subgraphs. We then iteratively grew the graph of size n using a modified preferential attachment scheme [27]. Specifically, in each step we added an additional node with signal value x_{n+1} drawn uniformly at random from $\{1, \ldots, 5\}$, and then placed 5 edges between the new node and existing nodes with probability

$$\mathrm{P}\{W_{i,n+1} = 1\} \propto d_i^a(n) \exp\left(-5|x_{n+1} - x_i|\right),$$

where $d_i(n)$ is the current degree of node i and $a > 0$ is a parameter. Edges therefore were preferably added for nodes with large degree $d_i(n)$ (leading to a scale-free graph with power-law degree distribution) and for nodes that have the same signal value (enforcing a clustered graph structure with 5 communities and few edges between distinct communities). An example graph is shown in Fig. 1.

Algorithm 1 ACD graph signal recovery

```
Input: $\mathbf{x}_0$, $\mu > 0$
Initialize: $L_i = 1 + \frac{2}{\mu}\big(\sum_{j\in\mathrm{ch(i)}} W_{i,j} + \sum_{j\in\mathrm{pa(i)}} W_{j,i}\big)$,
            $\mathbf{u} = \mathbf{x}_0$, $\mathbf{w} = \mathbf{x}_0$, $t = \frac{1}{N}$, $B_{2,2} = 1$, $B_{2,1} = 0$
repeat
  $\tau = t$
  $t = \frac{1+\sqrt{1+4N^2\tau^2}}{2N}$
  choose $i$ with uniform probability from $\{1, \ldots, N\}$
  $z_i = B_{2,1}u_i + B_{2,2}w_i$
  for $j \in \mathrm{ch}(i)$ do
      $z_j = B_{2,1}u_j + B_{2,2}w_j$
      $G_{i,j} = (z_j - z_i)\sqrt{W_{i,j}}$
      $a_j = \begin{cases} \frac{1}{\mu} & \text{if } |G_{i,j}| \leq \lambda\mu, \\ \frac{\lambda}{|G_{i,j}|} & \text{else} \end{cases}$
  end
  $r = \sum_{j\in\mathrm{ch}(i)} a_j \sqrt{W_{i,j}} G_{i,j}$
  for $l \in \mathrm{pa}(i)$ do
      $z_l = B_{2,1}u_l + B_{2,2}w_l$
      $G_{l,i} = (z_i - z_l)\sqrt{W_{l,i}}$
      $d_l = \begin{cases} \frac{1}{\mu} & \text{if } |G_{l,i}| \leq \lambda\mu, \\ \frac{\lambda}{|G_{l,i}|} & \text{else} \end{cases}$
  end
  $b = \sum_{l\in\mathrm{pa}(i)} d_l \sqrt{W_{l,i}} G_{l,i}$
  $c = r - b$
  $g = \begin{cases} x_i - y_i - c & \text{if } 1 \leq i \leq M, \\ -c & \text{else} \end{cases}$
  $B_{2,1} \leftarrow B_{2,1} + \frac{1}{tN}(1 - B_{2,1})$
  $B_{2,2} \leftarrow B_{2,2}\big(1 - \frac{1}{tN}\big)$
  $u_i \leftarrow u_i - \frac{\tau}{L_i} g$
  $m = -\frac{B_{2,1}}{B_{2,2}}\tau + \frac{1}{B_{2,2}}\frac{\tau+tN-1}{tN}$
  $w_i \leftarrow w_i - \frac{m}{L_i} g$
until stopping criterion is satisfied
Output: $\hat{\mathbf{x}} = B_{2,1}\mathbf{u} + B_{2,2}\mathbf{w}$
```

Simulation Setup. We constructed a graph with $N = 5000$ nodes and 49910 undirected edges. The graph signal was sampled at $M = 500$ randomly chosen nodes (thus $M/N = 10\%$). The noise was i.i.d. Gaussian with zero mean and variance σ^2. We used Algorithm 1 with $\mu = 1$ and $\mu = 100$ to recover the graph signal. We compare our method with the following algorithms: 1) FISTA [23], [24] on the smoothed proxy f_μ with global Lipschitz constant $L_\mu = 1 + \frac{2}{\mu}\max_i d_i$; 2) the augmented ADMM algorithm [19, Section 4.2] with initial step size $\rho = \frac{1}{2\max d_i}$ and varying penalty strategy [19, Section 2.3]. The augmented ADMM algorithm

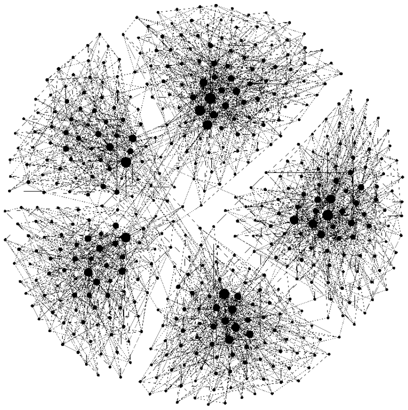


Fig. 1: Example graph (with $N = 500$) from our construction with $a = 1$ (node size is proportional to node degree).

is of particular interest for scale-free graphs since it involves a scaling matrix that takes nonuniform degree distribution into account. All experiments were repeated 10 times with different realizations of the graph signal, the graph topology, the sampling set, and the noise. We quantify the recovery performance in terms of the normalized mean squared error (NMSE) $e_k^2 = \mathrm{E}\{\|\hat{\mathbf{x}}_k - \mathbf{x}\|_2^2\} / \mathrm{E}\{\|\mathbf{x}\|_2^2\}$. The signal-to-noise ratio (SNR) was $\mathrm{SNR} = \mathrm{E}\{x_i^2\}/\sigma^2 = 20\,\mathrm{dB}$.

Results. On average, N iterations of Algorithm 1 require $20|\mathcal{E}| + \mathcal{O}(N)$ operations whereas one iteration of FISTA and the augmented ADMM (ignoring the operation count of the penalty strategy) requires $10|\mathcal{E}| + \mathcal{O}(N)$ operations. Therefore, in our convergence plots we use a normalized iteration count that compares every Nth iteration of Algorithm 1 with every second iteration of FISTA and augmented ADMM.

Fig. 2(a) shows the results for graphs with $a = 1$ and regularization parameter $\lambda = 0.0001$. Since most coordinate-wise Lipschitz constants in the corresponding graphs are substantially smaller than the global Lipschitz constant, our method (labeled ACD) indeed converges faster than FISTA, and also significantly faster than the augmented ADMM. By applying stronger smoothing, the convergence speed can be improved but the reconstruction performance deteriorates in general, which is in agreement with [18], [20]. We highlight that Nestereov's smoothing strategy seems to be very effective for graphs with vastly varying node degrees, even though the additional smoothing step typically slows down the convergence on different graph models [16], [19]. For graphs with $a = 1.5$ and thus larger degree variations (stronger hubs), the convergence of FISTA is further slowed down so that the advantage of ACD becomes even more pronounced as can be seen in Fig. 2(b). In Fig. 2(c) we used graphs with $a = 1.5$ and regularization parameter $\lambda = 0.01$. Note that for a larger regularization parameter the superior convergence performance of ACD seems to diminish compared to the augmented ADMM algorithm.

VII. CONCLUSION

We considered the recovery of smooth graph signal from noisy samples. The smoothness of the graph signal is quantified in terms of the graph total variation. By combining Nesterov's smoothing technique with accelerated coordinate descent, we developed a learning algorithm particularly well suited for graphs with strongly skewed degree distribution. Numerical experiments confirmed the superior convergence of our method on graphs of this type. The convergence speed of our method could be further improved by using continuation techniques [18]; this is left to future work.

APPENDIX: PROOF OF LEMMA 1

Using (9), the linearity of the divergence operator, and the triangle inequality, we obtain

$$\begin{aligned}&|\nabla_l f_\mu(\mathbf{x} + s\mathbf{e}_l) - \nabla_l f_\mu(\mathbf{x})| \\ &\quad \leq |s| + \left|\big(\operatorname{div}_{\mathcal{G}}(\mathbf{P}_\mu(\mathbf{x} + s\mathbf{e}_l) - \mathbf{P}_\mu(\mathbf{x}))\big)_l\right|.\end{aligned} \tag{12}$$

The lth element of the graph divergence can be expressed as $(\operatorname{div}_{\mathcal{G}} \mathbf{P})_l = \sum_{j \in \mathcal{V}} \sqrt{W_{l,j}} P_{l,j} - \sqrt{W_{j,l}} P_{j,l} = (\operatorname{div}_{\mathcal{G}_l} \mathbf{P})_l$, where $\mathcal{G}_l$ is the graph defined by the weight matrix $\mathbf{W}_l \in \mathbb{R}^{N \times N}$ with elements

$$W_{i,j}^l \triangleq \begin{cases} W_{i,j} & \text{if } i = l \text{ or } j = l, \\ 0 & \text{else.} \end{cases}$$

Therefore

$$\begin{aligned}&\left|\big(\operatorname{div}_{\mathcal{G}}(\mathbf{P}_\mu(\mathbf{x} + s\mathbf{e}_l) - \mathbf{P}_\mu(\mathbf{x}))\big)_l\right| \\ &\quad \leq \|\operatorname{div}_{\mathcal{G}_l}(\mathbf{P}_\mu(\mathbf{x} + s\mathbf{e}_l) - \mathbf{P}_\mu(\mathbf{x}))\|_2 \\ &\quad \leq \|\operatorname{div}_{\mathcal{G}_l}\|_{\mathrm{op}} \|\mathbf{P}_\mu(\mathbf{x} + s\mathbf{e}_l) - \mathbf{P}_\mu(\mathbf{x}))\|_{\mathrm{F}}.\end{aligned} \tag{13}$$

Since $\mathbf{P}_\mu(\mathbf{x}))$ is the orthogonal projection of $\frac{1}{\mu}\nabla_{\mathcal{G}}(\mathbf{x})$ onto $\mathcal{P}_\lambda$, we further have

$$\begin{aligned}\|\mathbf{P}_\mu(\mathbf{x} + s\mathbf{e}_l) - \mathbf{P}_\mu(\mathbf{x}))\|_{\mathrm{F}} &\leq \frac{1}{\mu}\|\nabla_{\mathcal{G}}(s\mathbf{e}_l)\|_{\mathrm{F}} \\ &= \frac{1}{\mu}\|\nabla_{\mathcal{G}_l}(s\mathbf{e}_l)\|_{\mathrm{F}} \leq |s|\frac{1}{\mu}\|\nabla_{\mathcal{G}_l}\|_{\mathrm{op}},\end{aligned} \tag{14}$$

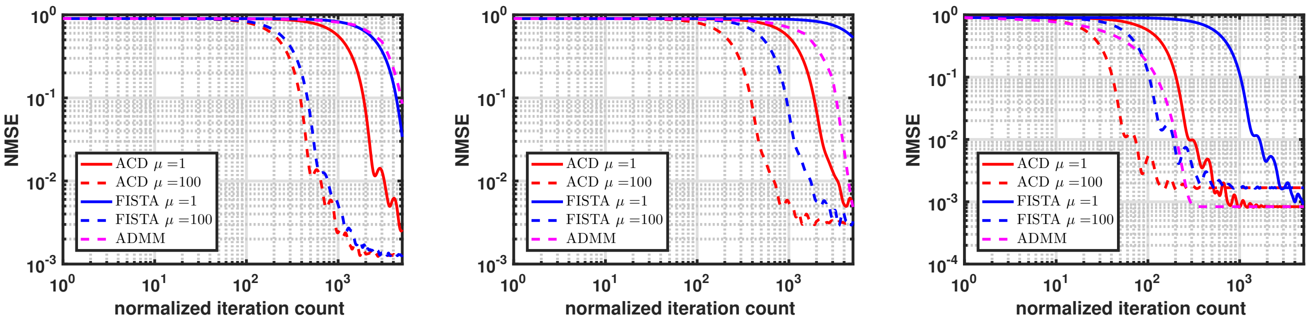
where $\nabla_{\mathcal{G}_l}$ is the gradient on $\mathcal{G}_l$. Since $\nabla_{\mathcal{G}_l}$ is the adjoint of $\operatorname{div}_{\mathcal{G}_l}$, combination of (12), (13), and (14) yields

$$|(\nabla f_\mu(\mathbf{x}+s\mathbf{e}_l))_l - (\nabla f_\mu(\mathbf{x}))_l| \leq |s|\Big(1 + \frac{1}{\mu}\|\nabla_{\mathcal{G}_l}\|_{\mathrm{op}}^2\Big). \tag{15}$$

According to [14, Proposition 5.2], the graph gradient is bounded as $\|\nabla_{\mathcal{G}}\|_{\mathrm{op}}^2 \leq 2 \max_i \sum_{j=1}^N (W_{i,j} + W_{j,i})$. Applying this bound to $\mathcal{G}_l$ entails $\|\nabla_{\mathcal{G}_l}\|_{\mathrm{op}}^2 \leq 2 \max_i \sum_{j=1}^N (W_{i,j}^l + W_{j,i}^l) = 2 \sum_{j=1}^N (W_{l,j} + W_{j,l}) = 2d_l$, which in combination with (15) leads to (11).

REFERENCES

[1] 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 net-



(a) (b) (c)

Fig. 2: Reconstruction performance of ACD, FISTA, and augmented ADMM in terms of normalized mean-square error versus normalized iteration count for SNR $= 20\,\text{dB}$: (a) $a = 1$, $\lambda = 0.0001$, (b) $a = 1.5$, $\lambda = 0.0001$, (c) $a = 1.5$, $\lambda = 0.01$.

works and other irregular domains. *IEEE Signal Process. Mag.*, 30(3):83–98, May 2013.

[2] A. Sandryhaila and J. M. F. Moura. Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. *IEEE Signal Process. Mag.*, 31(5):80–90, Sept. 2014.

[3] A. Sandryhaila and J. M. F. Moura. Discrete signal processing on graphs. *IEEE Trans. Signal Process.*, 61(7):1644–1656, Apr. 2013.

[4] S. K. Narang and A. Ortega. Perfect reconstruction two-channel wavelet filter banks for graph structured data. *IEEE Trans. Signal Process.*, 60(6):2786–2799, June 2012.

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

[6] S. Aral and D. Walker. Identifying influential and susceptible members of social networks. *Science*, 337(6092):337–341, July 2012.

[7] L.-W. Ku, Y.-T. Liang, and H.-H. Chen. Opinion extraction, summarization and tracking in news and blog corpora. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 100–107, Palo Alto, CA, Mar. 2006.

[8] M. Tremayne, N. Zheng, J. K. Lee, and J. Jeong. Issue publics on the web: Applying network theory to the war blogosphere. *J. Computer-Mediated Commun.*, 12(1):290–310, Oct. 2006.

[9] R. Saidi, M. Maddouri, and E.M. Nguifo. Protein sequences classification by means of feature extraction with substitution matrices. *BMC Bioinformatics*, 11(1):1–13, Apr. 2010.

[10] P. Bühlmann and S. v. d. Geer. *Statistics for High-Dimensional Data*. Springer, 2011.

[11] M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised learning on large graphs. In *Proc. Annual Conf. Learning Theory (COLT)*, pages 624–638, Banff, Canada, July 2004.

[12] S. Chen, A. Sandryhaila, and J. Kovačević. Distributed algorithm for graph signal inpainting. In *Proc. IEEE ICASSP*, pages 3731–3735, Brisbane, Australia, Apr. 2015.

[13] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo. Signals on graphs: uncertainty principle and sampling. *IEEE Trans. Signal Process.*, 64(18):4845–4860, Sept. 2016.

[14] G. Gilboa and S. Osher. Nonlocal operators with applications to image processing. *Multiscale Model. Simul.*, 7(3):1005–1028, Nov. 2008.

[15] A. Jung, P. Berger, G. Hannak, and G. Matz. Scalable graph signal recovery for big data over networks. In *Proc. IEEE Workshop Signal Process. Advances in Wireless Commun.*, pages 1–6, Edinburgh, UK, July 2016.

[16] P. Berger, G. Hannak, and G. Matz. Graph signal recovery via primal-dual algorithms for total variation minimization. *to be published in IEEE Journal of Selected Topics in Signal Processing*, 2017.

[17] G. Hannak, P. Berger, A. Jung, and G. Matz. Efficient graph signal recovery over big networks. In *Proc. Asilomar Conf. Signals, Systems, Computers*, pages 1839–1843, Pacific Grove, CA, Nov. 2016.

[18] S. Becker, J. Bobin, and E. J. Candès. NESTA: a fast and accurate first-order method for sparse recovery. *SIAM J. Imaging Sci.*, 4(1):1–39, 2011.

[19] Y. Zhu. An Augmented ADMM Algorithm With Application to the Generalized Lasso Problem. *J. Comput. Graph. Statist.*, 26(1):195–204, Feb. 2017.

[20] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1, Ser. A):127–152, Dec. 2005.

[21] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM J. Optim.*, 22(2):341–362, Apr. 2012.

[22] Y. T. Lee and A. Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science—FOCS 2013*, pages 147–156, Berkeley, CA, USA, Oct. 2013.

[23] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[24] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, Mar. 2009.

[25] A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Trans. Image Process.*, 18(11):2419–2434, Nov. 2009.

[26] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vision*, 40(1):120–145, 2011.

[27] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, Oct. 1999.

ISBN 978-0-9928626-7-1 © EURASIP 2017