Graph Signal Recovery via Primal-Dual Algorithms for Total Variation Minimization

Peter Berger, Gabor Hannak, and Gerald Matz

Institute of Telecommunications, Technische Universität Wien
Gusshausstrasse 25/389, 1040 Wien, Austria
email: firstname.lastname@nt.tuwien.ac.at

***Abstract*—We consider the problem of recovering a smooth graph signal from noisy samples taken on a subset of graph nodes. The smoothness of the graph signal is quantified in terms of total variation. We formulate the signal recovery task as a convex optimization problem that minimizes the total variation of the graph signal while controlling its global or node-wise empirical error. We propose a first-order primal-dual algorithm to solve these total variation minimization problems. A distributed implementation of the algorithm is devised to handle large-dimensional applications efficiently. We use synthetic and real-world data to extensively compare the performance of our approach with state-of-the art methods.**

I. Introduction

A. Background

Several new branches in signal processing have recently emerged in response to the need of dealing with huge amounts of data. Such datasets, characterized by high volume, variety, and velocity, and the associated storage and computation schemes are nowadays subsumed under the term "Big Data" [3]. Many big data problems can be tackled successfully by modeling the data in terms of graphs, which have the advantage of being flexible and computationally efficient and of featuring attractive scaling properties. As a result, graph signal processing (GSP) has evolved into a particularly promising engineering paradigm in this field [4]–[7]. Graph signal models are of particular interest since they are inherently well-suited for distributed storage and processing (e.g., in the form of message passing algorithms), which is a key aspect in handling big data. Furthermore, they capture similarity relations within the dataset in a straightforward and versatile manner and thus provide an efficient means to cope with data of heterogeneous nature.

GSP has been used in a wide range of real-world applications. In online social networks, users can be represented as nodes in a graph whose edges connect users with similar attributes or friends and followers. The resulting graph can be used to find influential users or to track opinion propagation [8]–[10]. Applying a similar approach to the web and the blogosphere has helped to understand sociopolitical phenomena [11], [12]. In the context of online retailers and services, similar behavioral patterns establish the edges in a user graph, whose structure is instrumental for the development of recommender systems [13]. For further application examples of GSP, the reader is referred to [4] and [6].

This paper deals with the important GSP problem of recovering a graph signal from inaccurate and incomplete data [14]–[24]. This problem is also referred to as semi-supervised learning [25]–[28] or graph signal inpainting [29]. More specifically, the measurements correspond to noisy samples of the graph signal taken at a small subset of graph nodes (the additive noise subsumes potential measurement and modeling errors). The reconstruction of graph signals from noisy samples presupposes some form of smoothness model. Such a model amounts to the assumption that the graph signal is smooth with respect to the graph topology, i.e., the signal changes between neighboring nodes are small. In contrast to the bulk of existing work in GSP, we here model graph signal smoothness in terms of small total variation. Graph total variation was previously used e.g. for regularization in tomographic reconstructions [30] or for computing the balanced cut of a graph [31]. The total variation semi-norm [4], [32] is a generalization of the total variation of continuous functions; it is a convex function but has the handicap of not being differentiable.

B. Contributions

We formulate the graph signal recovery problem as an optimization problem that corresponds to finding a graph signal that has minimal total variation among all signals that lie within a prescribed distance to the measured signal samples. In this formulation, the (estimated) measurement noise level can be directly incorporated, which is a major advantage compared to the regularization terms used in most existing work. The resulting optimization problem is convex but non-smooth. We propose to solve this problem using a first-order primal-dual algorithm that is obtained by reformulating the original optimization task as a saddle point problem. The primal-dual algorithm requires the computation of orthogonal projections for which we derive simple explicit expressions. We derive a lower bound on the operator norm of the graph gradient, which allows us to choose the optimal step size (modulo a factor of at most two) in the primal-dual algorithm. We further develop an efficient distributed implementation of the proposed primal-dual algorithm that has favorable scaling behavior and is instrumental in employing the method in actual

Funding by WWTF Grant ICT15-119. Part of this work was performed while the first author was affiliated with Aalto University (Finland). Preliminary portions of this work have been presented at IEEE SPAWC 2016 (Edinburgh, UK) and Asilomar 2016 (Pacific Grove, CA) [1], [2].

big data applications. Finally, we corroborate the usefulness of our approach by providing extensive numerical experiments on synthetic and real-world data. More specifically, we assess the performance of our scheme using isotropic and anisotropic total variation on two types of synthetic community graphs [33] and compare our results with those achieved by state-of-the-art methods like Tikhonov regularization [26], kernel-based reconstruction [24], and graph variation minimization [22], [29]. A similar comparison is provided for a large-scale real-world graph (Amazon co-purchase graph). Our experiments confirm that our scheme is superior or at least competitive in terms of reconstruction performance and computational complexity.

C. Related Work

There is a number of papers on the reconstruction of smooth graph signals that build on different smoothness models.

In [26], [27] the authors propose recovery algorithms based on a Tikhonov regularization for graph signals. Several approaches build on the generalization of spectrum analysis to graph signals, which carries over many concepts known from classical signal processing to irregular graph topologies by using the eigendecomposition of the graph Laplacian or of the adjacency matrix as a generalization of the Fourier transform [14]–[21]. Specifically, smoothness of graph signals is modeled in terms of band-limitedness in the (eigen-)spectral domain. This approach has the advantage that it allows the formulation of actual sampling theorems. Another notion of smoothness, termed graph variation, quantifies the difference between the original graph signal and a filtered version of the signal obtained by applying a graph shift operator (e.g., the graph adjacency matrix) [22], [29]. Tikhonov regularization [26], [27] and [22, eq. (27)] as well as the least squares approach in [14]–[19] can be viewed as special cases of kernel-based methods [24], in which alternative smoothness kernels are obtained via non-negative functions on the spectrum of the graph Laplacian [23], [34], [35].

In our own previous work we tackled a Lagrangian version of the total variation minimization problem via a combination of denoising and ADMM [1]. In contrast to ℓ_1 regularization, the proximal operator in total variation denoising admits no closed form solution.

This entails that (i) in each iteration ADMM requires the solution of a subproblem whose effect on the accuracy of the overall scheme is unclear and (ii) fast first-order methods such as FISTA [36] cannot be directly applied (cf. also the discussion in Section III-B). Alternative ADMM approaches applicable to our problem setting have been considered in [37] (which appeared after we first submitted this manuscript). Specifically, an augmented ADMM algorithm has been proposed since standard ADMM requires numerically expensive matrix inversions [37, Section 4.3]. Interestingly, for a certain choice of parameters our primal-dual algorithm turns out to be equivalent to the augmented ADMM algorithm applied to the dual problem [37, Theorem 2].

In [2], we studied the solution of the total variation minimization problem via Nesterov's method [38]. The Nesterov scheme applies an optimal first-order method to a smoothed version of the objective function. The problem with Nesterov's method lies in tuning the smoothing parameter, which trades off the convergence speed and the accuracy of the algorithm. For large-scale graphs, a reasonable choice for the smoothing parameter is neither obvious nor computationally feasible. Alternative approaches are provided by subgradient methods, which usually have a slow rate of convergence of $\mathcal{O}(1/\sqrt{k})$, however [39, Section 3.2.3]. These drawbacks of the denoising, Nesterov, and subgradient schemes motivated us to follow the suggestion in [40, Section 4.3] and use splitting strategies such as the primal-dual algorithm for solving the total variation minimization problem.

D. Notation

Matrices are denoted by boldface uppercase letters and column vectors are denoted by boldface lowercase letters. The elements of a vector $\mathbf{x}$ or a matrix $\mathbf{A}$ are written as x_i and A_{ij}, respectively. For the inner product and the induced norm on a generic Hilbert space $\mathcal{H}$ we use the notation $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and $\| \cdot \|_{\mathcal{H}}$, respectively. Specifically, for vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ we have $\langle \mathbf{x}, \mathbf{y} \rangle_2 \triangleq \sum_i x_i y_i$ and $\|\mathbf{x}\|_2 \triangleq \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle_2}$. Furthermore, for matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{N \times N}$ we have $\langle \mathbf{X}, \mathbf{Y} \rangle_{\mathrm{F}} \triangleq \sum_{i,j} X_{ij} Y_{ij}$ and $\|\mathbf{X}\|_{\mathrm{F}} \triangleq \sqrt{\langle \mathbf{X}, \mathbf{X} \rangle_{\mathrm{F}}}$. Given a linear operator $\mathbf{B}$ that maps a Hilbert space $\mathcal{H}_1$ to a Hilbert space $\mathcal{H}_2$, its adjoint is denoted by $\mathbf{B}^*$ and its operator norm is defined by $\|\mathbf{B}\|_{\mathrm{op}} \triangleq \sup_{\|\mathbf{x}\|_{\mathcal{H}_1} \leq 1} \|\mathbf{B}\mathbf{x}\|_{\mathcal{H}_2}$. The symbol $\mathbf{I}_M$ is used for the identity matrix of dimension M and $\mathbf{0}_{M \times N}$ and $\mathbf{1}_{M \times N}$ are all-zeros and all-ones matrices, respectively, of dimension $M \times N$ (if there is no danger of confusion, the dimensions will be omitted). The orthogonal projection of a point $\mathbf{x}$ onto a closed convex subset $\mathcal{C} \subseteq \mathcal{H}$ of a Hilbert space is denoted by $\pi_{\mathcal{C}}(\mathbf{x}) = \arg\min_{\mathbf{z} \in \mathcal{C}} \|\mathbf{z} - \mathbf{x}\|_{\mathcal{H}}$. For the Kronecker delta, we use the symbol δ_{ij} and $\mathrm{sign}(\cdot)$ denotes the sign function. The expectation of random variables is written as $\mathrm{E}\{\cdot\}$.

E. Paper Organization

In Section II, we describe the graph signal sampling model and the total variation smoothness metric. Section III introduces the convex recovery problems and discusses their relation to denoising. In Section IV we derive a saddlepoint reformulation of the optimization problem and present the primal-dual algorithm proposed to solve the recovery problems. A distributed implementation is devised in Section V. Numerical experiments with synthetic and real-world data are discussed in Section VI and conclusions are provided in Section VII.

II. Sampling and Smoothness Model

A. Graph Signal Model

We consider signals on weighted directed graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ with vertex set $\mathcal{V} = \{1, \ldots, N\}$, edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and weight matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$. The entries $W_{ij} \geq 0$ of the weight matrix are non-zero if and only if $(i, j) \in \mathcal{E}$. The weights W_{ij} describe the strength of the connection from node i to j. We assume the graph has no loops, i.e., $W_{ii} = 0$

for all $i \in \mathcal{V}$. A graph signal is a mapping from the vertex set $\mathcal{V}$ to $\mathbb{R}$, i.e., it associates with each node $i \in \mathcal{V}$ a real number $x_i \in \mathbb{R}$. These real numbers can be conveniently arranged into a length-N vector $\mathbf{x} \triangleq (x_1, \ldots, x_N)^T \in \mathbb{R}^N$.

We consider the problem of recovering a graph signal $\mathbf{x}$ from $M < N$ noisy samples. Without loss of generality, we assume that the samples are taken on the vertex set $\{1, \ldots, M\}$ (this can always be achieved by appropriately relabeling the vertices). Our linear noisy sampling model on the sampling set thus reads

$$y_i = x_i + u_i, \quad i = 1, \ldots, M. \tag{1}$$

Here, the additive noise $u_i \in \mathbb{R}$, $i = 1, \ldots, M$, captures any measurement and modeling errors. Stacking the measurements y_i and the noise u_i, $i = 1, \ldots, M$, into the length-M vectors $\mathbf{y} = (y_1, \ldots, y_M)^T$ and $\mathbf{u} = (u_1, \ldots, u_M)^T$, respectively, we obtain the vectorized measurement model

$$\mathbf{y} = \mathbf{S}\mathbf{x} + \mathbf{u}, \tag{2}$$

where the elements of the sampling matrix $\mathbf{S} \in \{0,1\}^{M \times N}$ are given by $S_{ij} = \delta_{ij}$ and thus

$$\mathbf{S} = (\mathbf{I}_M \ \mathbf{0}_{M \times (N-M)}). \tag{3}$$

B. Total Variation

In order to recover the unknown graph signal $\mathbf{x}$, we assume the graph signal to be smooth in the sense that it varies little over strongly connected nodes. In order to define a precise metric for the smoothness of a graph signal, we define the local gradient $\nabla_i \mathbf{x} \in \mathbb{R}^N$ of a graph signal $\mathbf{x}$ at node $i \in \mathcal{V}$ as the length-N column vector whose jth element equals

$$\big(\nabla_i \mathbf{x}\big)_j \triangleq (x_j - x_i) W_{ij}. \tag{4}$$

The smoothness of a graph signal is then quantified in terms of the (isotropic) graph total variation, defined as [4], [32]

$$\|\mathbf{x}\|_{\mathrm{TV}} \triangleq \sum_{i \in \mathcal{V}} \|\nabla_i \mathbf{x}\|_2 = \sum_{i \in \mathcal{V}} \sqrt{\sum_{j \in \mathcal{V}} (x_j - x_i)^2 W_{ij}^2}. \tag{5}$$

An alternative definition is given by the anisotropic total variation $\|\mathbf{x}\|_{\mathrm{TV}}^A \triangleq \sum_{i=1}^N \sum_{j=1}^N |x_j - x_i| W_{ij} = \sum_{i \in \mathcal{V}} \|\nabla_i \mathbf{x}\|_1$. While the isotropic total variation is the ℓ_1 norm of the ℓ_2 norms of the local gradients, the anisotropic total variation is the ℓ_1 norm of the overall graph gradient $\nabla \mathbf{x} = (\nabla_1 \mathbf{x}, \ldots, \nabla_N \mathbf{x})^T \in \mathbb{R}^{N \times N}$ (equivalently, the ℓ_1 norm of the ℓ_1 norms of the local gradients). Thus, the isotropic definition favors sparsity in the local gradients (i.e., a small number of smooth signal changes), whereas the anisotropic definition favors sparsity in the overall graph gradient (i.e., a possibly larger number of abrupt signal changes). While we limit our discussion to the isotropic case, our results carry over to the anisotropic case with minor modifications (the only change is in the constraint set (17), see also [41]). The difference between isotropic and anisotropic total variation will be further explored in our simulations (cf. Section VI). The graph total variation is a generalization of the total variation used for image denoising [42]. More specifically, the definition of total variation used in [42] is re-obtained (modulo boundary effects) from (5) with an "image graph" consisting of $N = K^2$ pixels at coordinates $(k_i, l_i) \in \{1, \ldots, K\}^2$ and edges with weight $W_{ij} = 1$ if node j is the northern or eastern neighbor of node i, i.e.,

$$(i,j) \in \mathcal{E} \quad \Longleftrightarrow \quad (k_j, l_j) \in \big\{(k_i, l_i{+}1), (k_i{+}1, l_i)\big\}.$$

Note that the total variation defined in (5) depends on the weights W_{ij} and thus is a measure of the smoothness of a graph signal $\mathbf{x}$ relative to the graph topology. Changing the graph by adding or removing edges will therefore lead to a different total variation for the same graph signal. Furthermore, the graph total variation in (5) can be shown to be a seminorm, i.e., it is homogeneous and satisfies the triangle inequality but $\|\mathbf{x}\|_{\mathrm{TV}} = 0$ for any constant signal $\mathbf{x} = c\,\mathbf{1}$ (for the special case of images this was stated already in [41]). Being a seminorm, $\|\mathbf{x}\|_{\mathrm{TV}}$ is also a convex function of $\mathbf{x}$.

C. Alternative Smoothness Models

In most existing work, graph signal reconstruction is based on different smoothness models. Many papers [14]–[21], [23]–[28] use the graph Laplacian

$$\mathbf{L} \triangleq \mathrm{diag}\,\big(\sum_i W_{1i}, \ldots, \sum_i W_{Ni}\big) - \mathbf{W},$$

to quantify the amount of graph signal variation. Specifically, a signal is considered smooth if the quadratic form $\mathbf{x}^T g(\mathbf{L}) \mathbf{x}$ is small, with $g(\cdot)$ denoting a particular matrix function. The most common case corresponds to $g(\mathbf{L}) = \mathbf{L}$ [26], which leads to

$$\mathbf{x}^T \mathbf{L} \mathbf{x} = \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} (x_j - x_i)^2 W_{ij}.$$

Other metrics are obtained by choosing $g(\cdot)$ as higher-order polynomial or exponential function (e.g., [23], [24]).

Alternative approaches assume that the graph signal lives in a "bandlimited" subspace corresponding to eigenspaces of $\mathbf{L}$ associated to the smallest eigenvalues [18]–[20]. Bandlimited graph signals have $\mathbf{x}^T g(\mathbf{L}) \mathbf{x} = 0$ with $g(\cdot)$ being a unit step function.

Finally, the graph signal variation used in [22], [29] is defined as

$$s_p(\mathbf{x}) \triangleq \|\mathbf{x} - \mathbf{A}\mathbf{x}\|_p, \tag{6}$$

where $\mathbf{A} = \mathbf{W}/\|\mathbf{W}\|_{\mathrm{op}}$ is the graph's normalized adjacency matrix. As opposed to all other smoothness measures, in general $s_p(\mathbf{x})$ does not necessarily vanish for constant graph signals and may equal zero for non-constant signals. As an example consider the graph with normalized adjacency matrix $\mathbf{A} = \frac{1}{\sqrt{2}} \left(\begin{smallmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{smallmatrix}\right)$. Here, the graph variation $s_1(\mathbf{x})$ of the constant graph signal $\mathbf{x} = (1,1,1)^T$ equals 1 whereas it equals zero for the less smooth signal $\mathbf{x} = (1, \sqrt{2}, 1)^T$.

In our numerical experiments, we extensively compare graph signal reconstruction schemes based on the various smoothness models (see Section VI).

III. Recovery Strategies

A. Direct Formulation

We propose two strategies to recover the unknown graph signal $\mathbf{x}$ from the noisy samples $\mathbf{y}$ in (2). The first approach aims at finding the graph signal with minimal total variation subject to a single side constraint that controls the total empirical error $\|\mathbf{y}-\mathbf{S}\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{M}(y_i-x_i)^2}$, i.e.,

$$\min_{\mathbf{x}\in\mathbb{R}^N} \|\mathbf{x}\|_{\mathrm{TV}} \quad \text{s.t.} \quad \|\mathbf{y}-\mathbf{S}\mathbf{x}\|_2 \leq \varepsilon. \tag{7}$$

In some scenarios, the reliability of the individual measurements y_i may be different. This can be accounted for by allowing different empirical error levels at the various sampling nodes, which amounts to a variation of (7) with M side constraints:

$$\min_{\mathbf{x}\in\mathbb{R}^N} \|\mathbf{x}\|_{\mathrm{TV}} \quad \text{s.t.} \quad |y_i-x_i| \leq \varepsilon_i,\ i=1,\ldots,M. \tag{8}$$

Problems (7) and (8) can be written in a unifying manner as

$$\min_{\mathbf{x}\in\mathcal{Q}_i} \|\mathbf{x}\|_{\mathrm{TV}},$$

where the constraint sets are respectively defined as

$$\mathcal{Q}_1 \triangleq \{\mathbf{x}\in\mathbb{R}^N : \|\mathbf{y}-\mathbf{S}\mathbf{x}\|_2 \leq \varepsilon\},$$
$$\mathcal{Q}_2 \triangleq \{\mathbf{x}\in\mathbb{R}^N : |y_i-x_i| \leq \varepsilon_i,\ i=1,\ldots,M\}.$$

These constraints do not involve x_i, $i=M+1,\ldots,N$, and constrain $x_1,\ldots,x_M$ to lie within a hypersphere or a hyperrectangle centered at $\mathbf{y}$. The constraint sets $\mathcal{Q}_1$ and $\mathcal{Q}_2$ thus represent hypercylinders in $\mathbb{R}^N$ whose M-dimensional base is a hypersphere and a hyperrectangle, respectively. We note that the set $\mathcal{Q}_2$ is useful even for identical error levels $\varepsilon_i=\varepsilon$, $i=1,\ldots,M$; in fact, in this case the constraints amount to $\|\mathbf{y}-\mathbf{S}\mathbf{x}\|_\infty \leq \varepsilon$, which will be seen to have the advantage of facilitating per-node processing schemes.

The optimization problems (7) and (8) can also be rephrased in Lagrangian form,

$$\min_{\mathbf{x}\in\mathbb{R}^N} \|\mathbf{x}\|_{\mathrm{TV}} + \frac{\lambda}{2}\|\mathbf{y}-\mathbf{S}\mathbf{x}\|_2^2, \tag{9a}$$

$$\min_{\mathbf{x}\in\mathbb{R}^N} \|\mathbf{x}\|_{\mathrm{TV}} + \sum_{i=1}^{M}\frac{\lambda_i}{2}|y_i-x_i|^2. \tag{9b}$$

Since $\|\cdot\|_{\mathrm{TV}}$ is a convex function and the constraint sets are convex, problems (7), (8), and (9) are all convex optimization problems. However, $\|\cdot\|_{\mathrm{TV}}$ is not differentiable and hence the optimization problems are non-smooth. Furthermore, (7) and (8) in general do not have a unique solution. Consider the simple chain graph defined by $\mathcal{V}=\{1,2,3\}$ and $\mathcal{E}=\{(1,3),(3,2)\}$ with weights $W_{1,3}=W_{3,2}=1$. Here, $\|\mathbf{x}\|_{\mathrm{TV}}=|x_1-x_3|+|x_3-x_2|$. Assume $M=2$ and noise-free samples $y_1<y_2$. Solving (7) with $\varepsilon=0$ then amounts to choosing $x_3\in\mathbb{R}$ since the sampling constraint enforces $x_1=y_1$ and $x_2=y_2$. Here, any choice for x_3 in the interval $[y_1,y_2]$ leads to the same total variation and therefore corresponds to an optimal solution of (7).

A main virtue of (7) and (8) compared with the Lagrangian form problems (9) is the fact that knowledge of the measurement noise level can be directly incorporated by an appropriate choice of the parameters ε and $\varepsilon_1,\ldots,\varepsilon_M$. In fact, this advantage was one of the motivations for the work in [43], which partly inspired our graph signal recovery ideas.

B. Relation to Denoising

In Section IV, we propose to solve (7) and (8) using the primal-dual hybrid gradient (PDHG) method [44]–[46]. This method has a guaranteed convergence rate of $\mathcal{O}(1/k)$ in the objective function. In theory, there exist methods for solving (9) with a convergence rate of $\mathcal{O}(1/k^2)$ [36]. The FISTA Algorithm in [36] requires that in each iteration a denoising problem of the form

$$\min_{\mathbf{x}\in\mathbb{R}^N} \|\mathbf{x}\|_{\mathrm{TV}} + \frac{\lambda}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 \tag{10}$$

is solved. This problem is re-obtained from (9a) by setting $M=N$ which implies $\mathbf{S}=\mathbf{I}_N$. The optimization problem (10) has no simple closed-form expression and hence FISTA requires the repeated numerical solution of a subproblem (10) in each iteration. Such an approach was pursued in [41, Section V] for image deblurring.

We emphasize that there is a fundamental difference between the sampling problem (9) and the denoising problem (10) since $\|\mathbf{y}-\mathbf{S}\mathbf{x}\|_2^2$ is strongly convex for $M=N$ (denoising) but not for $M<N$ (sampling). There exist special methods for the minimization of the sum of a strongly convex function and a convex function [41], [47], [48]. It is straightforward to apply these methods to (10) by using the results from the present paper.

IV. Graph Signal Recovery

A. Saddle Point Formulation

We will next derive an alternative formulation of the optimization problems (7) and (8), which enables us to use the PDHG method [44]–[46] for graph signal recovery. For this purpose we introduce the graph gradient operator ∇ as a mapping from the Hilbert space $\mathbb{R}^N$ with inner product $\langle\cdot,\cdot\rangle_2$ to the the Hilbert space $\mathbb{R}^{N\times N}$ with inner product $\langle\cdot,\cdot\rangle_{\mathrm{F}}$; this mapping combines all local gradients $\nabla_i\mathbf{x}$, $i=1,\ldots,N$ (cf. (4)) into the matrix such that

$$\nabla\mathbf{x} = (\nabla_1\mathbf{x},\ldots,\nabla_N\mathbf{x})^T. \tag{11}$$

The negative adjoint of ∇ yields the divergence operator $\mathrm{div} = -\nabla^*$ that maps a matrix $\mathbf{Z}\in\mathbb{R}^{N\times N}$ to a vector $\mathrm{div}\,\mathbf{Z}\in\mathbb{R}^N$ whose ith element equals [32]

$$(\mathrm{div}\,\mathbf{Z})_i \triangleq \sum_{j\in\mathcal{V}} W_{ij}Z_{ij} - W_{ji}Z_{ji}. \tag{12}$$

We can reformulate the objective function $\|\mathbf{x}\|_{\mathrm{TV}}$ as

$$\|\mathbf{x}\|_{\mathrm{TV}} = \sum_{i\in\mathcal{V}} \|\nabla_i\mathbf{x}\|_2 = \|\nabla\mathbf{x}\|_{2,1}, \tag{13}$$

where we used

$$\|\mathbf{Z}\|_{2,1} \triangleq \sum_{i\in\mathcal{V}} \sqrt{\sum_{j\in\mathcal{V}} Z_{ij}^2}. \tag{14}$$

Furthermore, let us define the characteristic function of a set $\mathcal{Q} \subset \mathbb{R}^N$ as

$$\chi_{\mathcal{Q}}(\mathbf{x}) \triangleq \begin{cases} 0 & \text{for } \mathbf{x} \in \mathcal{Q}, \\ +\infty & \text{for } \mathbf{x} \notin \mathcal{Q}. \end{cases} \tag{15}$$

The problems (7) and (8) can then be rewritten as

$$\min_{\mathbf{x}\in\mathbb{R}^N} \|\nabla \mathbf{x}\|_{2,1} + \chi_{\mathcal{Q}_i}(\mathbf{x}). \tag{16}$$

This minimization can be cast as a saddle point problem using the generic procedure described in Appendix A. Observe that (16) is of the form (33) with $\mathcal{H}_1 = \mathbb{R}^N$, $\mathcal{H}_2 = \mathbb{R}^{N\times N}$, $f(\mathbf{Z}) = \|\mathbf{Z}\|_{2,1}$, $\mathbf{B}\mathbf{x} = \nabla\mathbf{x}$ and $g(\mathbf{x}) = \chi_{\mathcal{Q}_i}(\mathbf{x})$. To obtain a saddle point formulation of (16), all we need to do is determine the convex conjugate of $\|\cdot\|_{2,1}$. For this purpose we define a closed convex subset of $\mathbb{R}^{N\times N}$ that consists of all matrices whose rows all have norm less than 1, i.e.,

$$\mathcal{P} \triangleq \{\mathbf{P} = (\mathbf{p}_1, \dots, \mathbf{p}_N)^T \colon \|\mathbf{p}_i\|_2 \le 1,\ i = 1, \dots, N\}. \tag{17}$$

For any $\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_N)^T$, the convex conjugate of the indicator function $\chi_{\mathcal{P}}(\mathbf{X})$ of $\mathcal{P}$ can be written as

$$\chi^*_{\mathcal{P}}(\mathbf{Z}) = \sup_{\mathbf{X}\in\mathbb{R}^{N\times N}} \langle \mathbf{Z}, \mathbf{X}\rangle_{\mathrm{F}} - \chi_{\mathcal{P}}(\mathbf{X}) = \sup_{\mathbf{P}\in\mathcal{P}} \langle \mathbf{Z}, \mathbf{P}\rangle_{\mathrm{F}}.$$

Writing out the inner product of $\mathbf{Z}$ and $\mathbf{P}$ in terms of their rows and using the relation $\|\mathbf{z}_i\|_2 = \sup_{\|\mathbf{p}_i\|_2\le 1} \langle \mathbf{z}_i, \mathbf{p}_i\rangle_2$, we further obtain

$$\chi^*_{\mathcal{P}}(\mathbf{Z}) = \sum_{i\in\mathcal{V}} \sup_{\|\mathbf{p}_i\|_2\le 1} \langle \mathbf{z}_i, \mathbf{p}_i\rangle_2 = \sum_{i\in\mathcal{V}} \|\mathbf{z}_i\|_2 = \|\mathbf{Z}\|_{2,1}.$$

As a result, we have

$$\|\mathbf{Z}\|^*_{2,1} = \chi^{**}_{\mathcal{P}}(\mathbf{Z}) = \chi_{\mathcal{P}}(\mathbf{Z}), \tag{18}$$

which leads to the following saddle point formulation of (16):

$$\min_{\mathbf{x}\in\mathbb{R}^N} \max_{\mathbf{Z}\in\mathbb{R}^{N\times N}} \langle \nabla\mathbf{x}, \mathbf{Z}\rangle_{\mathrm{F}} - \chi_{\mathcal{P}}(\mathbf{Z}) + \chi_{\mathcal{Q}_i}(\mathbf{x}). \tag{19}$$

B. First-Order Primal-Dual Algorithm

We will now describe how to apply the PDHG method [44]–[46] to the graph signal recovery problems (7) and (8) in their saddlepoint formulation (19). A brief summary of the PDHG method and some comments regarding its application to graph signal recovery can be found in Appendix B. The main effort is to specialize the proximal operators (steps 5 and 7 in Algorithm 3 from Appendix B) to the graph signal recovery problem.

Let us first consider the proximal operator for the indicator function $\chi_{\mathcal{P}}(\mathbf{Y})$. We observe that

$$\begin{aligned}\operatorname{prox}_{\sigma\chi_{\mathcal{P}}}(\mathbf{Z}) &= \operatorname*{arg\,min}_{\mathbf{Z}'\in\mathbb{R}^{N\times N}} \frac{1}{2\sigma}\|\mathbf{Z}' - \mathbf{Z}\|^2_{\mathrm{F}} + \chi_{\mathcal{P}}(\mathbf{Z}') \\ &= \operatorname*{arg\,min}_{\mathbf{Z}'\in\mathcal{P}} \|\mathbf{Z}' - \mathbf{Z}\|^2_{\mathrm{F}} = \pi_{\mathcal{P}}(\mathbf{Z}),\end{aligned}$$

where $\pi_{\mathcal{P}}$ denotes the orthogonal projection onto the set $\mathcal{P}$. Consequently step 4 and 5 of Algorithm 3 applied to (19) can be written as $\mathbf{Z}^{(k+1)} = \pi_{\mathcal{P}}(\tilde{\mathbf{Z}})$ with $\tilde{\mathbf{Z}} = \mathbf{Z}^{(k)} + \sigma\nabla\overline{\mathbf{x}}^{(k)}$. Expressing the matrices involved in terms of their rows, $\mathbf{Z}^{(k)} = (\mathbf{z}_1^{(k)}, \dots, \mathbf{z}_N^{(k)})^T$ and $\tilde{\mathbf{Z}} = (\tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_N)^T$, we have $\tilde{\mathbf{z}}_i = \mathbf{z}_i^{(k)} + \sigma\nabla_i\overline{\mathbf{x}}^{(k)}$ and it is straightforward to verify that the orthogonal projection $\mathbf{Z}^{(k+1)}$ of $\tilde{\mathbf{Z}}$ onto the set $\mathcal{P}$ is given in terms of its rows as

$$\mathbf{z}_i^{(k+1)} = \frac{\tilde{\mathbf{z}}_i}{\max\{1, \|\tilde{\mathbf{z}}_i\|_2\}}. \tag{20}$$

Next consider the proximal operator for the indicator function $\chi_{\mathcal{Q}_i}(\mathbf{x})$. Here,

$$\begin{aligned}\operatorname{prox}_{\tau\chi_{\mathcal{Q}_i}}(\mathbf{x}) &= \operatorname*{arg\,min}_{\mathbf{x}'\in\mathbb{R}^N} \frac{1}{2\tau}\|\mathbf{x}' - \mathbf{x}\|^2_2 + \chi_{\mathcal{Q}_i}(\mathbf{x}') \\ &= \operatorname*{arg\,min}_{\mathbf{x}'\in\mathcal{Q}_i} \|\mathbf{x}' - \mathbf{x}\|^2_2 = \pi_{\mathcal{Q}_i}(\mathbf{x}).\end{aligned}$$

Using the fact that the dual of the graph gradient $\mathbf{B} = \nabla$ is the negative divergence, $\mathbf{B}^* = -\operatorname{div}$, steps 6 and 7 of Algorithm 3 amount to $\tilde{\mathbf{x}} = \mathbf{x}^{(k)} + \tau \operatorname{div}\mathbf{Z}^{(k+1)}$ and $\mathbf{x}^{(k+1)} = \pi_{\mathcal{Q}_i}\tilde{\mathbf{x}}$. Explicit expressions for the orthogonal projections $\pi_{\mathcal{Q}_i}(\tilde{\mathbf{x}})$ are obtained via the following Lemma.

Lemma IV.1. *For any matrix $\mathbf{S} \in \mathbb{R}^{M\times N}$ with orthonormal rows, i.e., $\mathbf{S}\mathbf{S}^T = \mathbf{I}$, the orthogonal projection*

$$\pi_{\mathcal{Q}_1}(\tilde{\mathbf{x}}) = \operatorname*{arg\,min}_{\mathbf{x}\in\mathcal{Q}_1} \big\|\mathbf{x} - \tilde{\mathbf{x}}\big\|^2_2 \tag{21}$$

of $\tilde{\mathbf{x}}$ onto $\mathcal{Q}_1 = \{\mathbf{x} \in \mathbb{R}^N \colon \|\mathbf{y} - \mathbf{S}\mathbf{x}\|_2 \le \varepsilon\}$ is given by

$$\pi_{\mathcal{Q}_1}(\tilde{\mathbf{x}}) = \begin{cases} \tilde{\mathbf{x}} + c\mathbf{S}^T\mathbf{r}, & \text{if } \|\mathbf{r}\|_2 > \varepsilon, \\ \tilde{\mathbf{x}}, & \text{if } \|\mathbf{r}\|_2 \le \varepsilon. \end{cases} \tag{22}$$

Here, $\mathbf{r} \triangleq \mathbf{y} - \mathbf{S}\tilde{\mathbf{x}}$ and $c \triangleq 1 - \frac{\varepsilon}{\|\mathbf{r}\|_2}$.

The orthogonal projection $\mathbf{v} \triangleq \pi_{\mathcal{Q}_2}(\tilde{\mathbf{x}})$ of $\tilde{\mathbf{x}}$ onto the set $\mathcal{Q}_2 = \{\mathbf{x} \in \mathbb{R}^N \colon |y_i - x_i| \le \varepsilon_i,\ i = 1, \dots, M\}$ is given element-wise by (here, $r_i = y_i - \tilde{x}_i$)

$$v_i = \begin{cases} y_i - \varepsilon_i \operatorname{sign}(r_i) & \text{if } i = 1, \dots, M \text{ and } |r_i| > \varepsilon_i, \\ \tilde{x}_i, & \text{else}. \end{cases} \tag{23}$$

Proof: The Lagrangian of (21) is

$$\mathcal{L}(\mathbf{v}, \lambda) = \frac{1}{2}\|\tilde{\mathbf{x}} - \mathbf{v}\|^2_2 + \frac{\lambda}{2}(\|\mathbf{y} - \mathbf{S}\mathbf{v}\|^2_2 - \varepsilon^2).$$

The corresponding KKT conditions for λ and $\mathbf{v}$ to be a primal-dual optimal pair are

$$\tilde{\mathbf{x}} - \mathbf{v} + \lambda\mathbf{S}^T(\mathbf{y} - \mathbf{S}\mathbf{v}) = \mathbf{0}, \tag{24}$$

$$\lambda(\|\mathbf{y} - \mathbf{S}\mathbf{v}\|^2_2 - \varepsilon^2) = 0, \tag{25}$$

$$\|\mathbf{y} - \mathbf{S}\mathbf{v}\|_2 \le \varepsilon, \tag{26}$$

$$\lambda \ge 0.$$

The complementary slackness condition (25) implies that either $\lambda = 0$ or $\|\mathbf{y} - \mathbf{S}\mathbf{v}\|_2 = \varepsilon$. Due to (24), the case $\lambda = 0$ yields $\mathbf{v} = \tilde{\mathbf{x}}$, which according to (26) requires $\|\mathbf{y} - \mathbf{S}\tilde{\mathbf{x}}\|_2 = \|\mathbf{r}\|_2 \le \varepsilon$. For $\lambda > 0$, the gradient condition (24) leads to

$$\big(\mathbf{I} + \lambda\mathbf{S}^T\mathbf{S}\big)\mathbf{v} = \tilde{\mathbf{x}} + \lambda\mathbf{S}^T\mathbf{y}. \tag{27}$$

Since $\mathbf{S}$ has orthonormal rows, $\mathbf{S}^T\mathbf{S}$ is an orthogonal projec-

tion matrix and hence

$$\left(\mathbf{I}+\lambda\mathbf{S}^T\mathbf{S}\right)^{-1}=\mathbf{I}-\frac{\lambda}{1+\lambda}\mathbf{S}^T\mathbf{S}. \tag{28}$$

Combining (28) with (27) yields

$$\mathbf{v}=\left(\mathbf{I}-\frac{\lambda}{1+\lambda}\mathbf{S}^T\mathbf{S}\right)\left(\tilde{\mathbf{x}}+\lambda\mathbf{S}^T\mathbf{y}\right). \tag{29}$$

For $\lambda>0$ complementary slackness requires $\|\mathbf{y}-\mathbf{S}\mathbf{v}\|_2=\varepsilon$; hence, with (29) and $\mathbf{S}\mathbf{S}^T=\mathbf{I}$ we conclude

$$\begin{aligned}\varepsilon&=\left\|\mathbf{y}-\mathbf{S}\Big(\mathbf{I}-\frac{\lambda}{1+\lambda}\mathbf{S}^T\mathbf{S}\Big)\left(\tilde{\mathbf{x}}+\lambda\mathbf{S}^T\mathbf{y}\right)\right\|_2\\&=\frac{1}{1+\lambda}\|\mathbf{y}-\mathbf{S}\tilde{\mathbf{x}}\|_2=\frac{\|\mathbf{r}\|_2}{1+\lambda},\end{aligned} \tag{30}$$

and hence $\|\mathbf{r}\|_2>\varepsilon$ due to $\lambda>0$. Solving (30) for λ leads to $\lambda=\|\mathbf{r}\|_2/\varepsilon-1$. Inserting this value for λ into (29), we obtain (22). The proof of (23) is similar to the proof of (22) and therefore omitted. ∎

In the special case of graph signal recovery, the projections $\pi_{\mathcal{Q}_i}(\mathbf{x})$ can be efficiently computed since they only involve the signal values on the sampling nodes (in our case $i=1,\dots,M$). The residual on those nodes is given by $r_i=y_i-\tilde{x}_i$ and hence (for $\|\mathbf{r}\|_2>\varepsilon$ respectively $|r_i|>\varepsilon_i$) the elements of $\mathbf{v}=\pi_{\mathcal{Q}_i}(\tilde{\mathbf{x}})$ specialize to $v_i=\tilde{x}_i+c(y_i-\tilde{x}_i)=(1-c)\tilde{x}_i+cy_i$, i.e., a convex combination of $\tilde{x}_i$ and y_i with c chosen such that $\mathbf{v}$ lies on the boundary of $\mathcal{Q}_i$.

C. Choice of Stepsize

For any $\tau,\sigma>0$, the PDHG Algorithm 3 is guaranteed to converge with rate $\mathcal{O}(1/k)$ as long as $\tau\sigma\|\nabla\|_{\mathrm{op}}^2<1$ (cf. [46, Theorem 1]). To satisfy this requirement, we need (an estimate of) the operator norm of the graph gradient. From a practical perspective, determining the exact operator norm in large-scale graphs is computationally too expensive; however, the following result provides simple upper and lower bounds. For this purpose, we define the degree of a vertex i as

$$d_i\triangleq\sum_{j\in\mathcal{V}}(W_{ij}^2+W_{ji}^2).$$

Note that the summation here is actually only over the neighbors of node i (for which W_{ij} or W_{ji} is non-zero). The degree d_i is a metric for how strongly node i is connected to other nodes in the graph. For non-weighted graphs with $W_{ij}\in\{0,1\}$, the degree d_i reduces to the number of neighbors (incident edges) of node i. We further define the maximum vertex degree of the graph $\mathcal{G}$ as

$$\rho_{\mathcal{G}}\triangleq\max_i d_i.$$

Lemma IV.2. *For any weighted directed graph $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathbf{W})$, the squared operator norm of the gradient operator ∇ in* (11) *is bounded as*

$$\rho_{\mathcal{G}}\le\|\nabla\|_{\mathrm{op}}^2\le 2\rho_{\mathcal{G}}. \tag{31}$$

Proof: For convenience, we rephrase the proof for the upper bound from [32]. Specifically, using (11) and the inequality $(a-b)^2\le 2(a^2+b^2)$ we obtain

$$\begin{aligned}\|\nabla\|_{\mathrm{op}}^2&=\sup_{\mathbf{x}\in\mathbb{R}^N\setminus\{\mathbf{0}\}}\frac{\|\nabla\mathbf{x}\|_{\mathrm{F}}^2}{\|\mathbf{x}\|_2^2}\\&=\sup_{\mathbf{x}\in\mathbb{R}^N\setminus\{\mathbf{0}\}}\frac{1}{\|\mathbf{x}\|_2^2}\sum_{i,j\in\mathcal{V}}(x_j-x_i)^2W_{ij}^2\\&\le\sup_{\mathbf{x}\in\mathbb{R}^N\setminus\{\mathbf{0}\}}\frac{1}{\|\mathbf{x}\|_2^2}\sum_{i,j\in\mathcal{V}}2(x_j^2+x_i^2)W_{ij}^2\\&=\sup_{\mathbf{x}\in\mathbb{R}^N\setminus\{\mathbf{0}\}}\frac{2}{\|\mathbf{x}\|_2^2}\sum_{i\in\mathcal{V}}x_i^2\,d_i\\&\le 2\max_i d_i=2\rho_{\mathcal{G}}.\end{aligned}$$

We next derive the lower bound in (31). Let $l\in\mathcal{V}$ be a vertex with maximum degree, $d_l=\rho_{\mathcal{G}}$, and consider the graph signal $\mathbf{x}'\in\mathbb{R}^N$ defined by $x_i'=c\,\delta_{il}$ (thus, $\|\mathbf{x}'\|_2^2=c^2$). With this choice we obtain

$$\begin{aligned}\|\nabla\|_{\mathrm{op}}^2&\ge\frac{\|\nabla\mathbf{x}'\|_{\mathrm{F}}^2}{\|\mathbf{x}'\|_2^2}=\frac{1}{\|\mathbf{x}'\|_2^2}\sum_{i,j\in\mathcal{V}}\left(x_j'-x_i'\right)^2W_{ij}^2\\&=\frac{1}{c^2}\sum_{i,j\in\mathcal{V}}c^2\left(\delta_{jl}-\delta_{il}\right)^2W_{ij}^2\\&=\sum_{j\in\mathcal{V}}(W_{lj}^2+W_{jl}^2)=\rho_{\mathcal{G}}.\end{aligned}$$

∎

D. Algorithm Statement

We now have all ingredients required for the adaptation of Algorithm 3 to our graph signal recovery problems (7) and (8), see Algorithm 1. In these and subsequent algorithms, it should be understood implicitly that statements made for a generic vertix i are to be performed for all nodes in the vertex set $\mathcal{V}$. Furthermore, Algorithm 1 potentially takes a vector argument $\boldsymbol{\varepsilon}=(\varepsilon_1,\dots,\varepsilon_M)$ for the error constraint. If the length of $\boldsymbol{\varepsilon}$ equals 1, the algorithm uses the projection on $\mathcal{Q}_1$ to control the total empirical error in (7), otherwise the projection on $\mathcal{Q}_2$ is employed to control the node-wise empirical error in (8).

Based on the upper bound in Lemma IV.2, we choose the stepsize parameters according to $\sigma=\frac{1}{2\tau\rho_{\mathcal{G}}}$. We observe that according to the lower bound in (31), this choice entails a loss of at most a factor 2 for the stepsize σ relative to the optimal value of $\frac{1}{\tau\|\nabla\|_{\mathrm{op}}^2}$, which is an acceptable penalty for avoiding the difficult computation of $\|\nabla\|_{\mathrm{op}}^2$. While the parameter τ in principle can still be chosen arbitrarily, we advocate a choice that balances the stepsize for both proximal steps, which leads to $\sigma=\tau=1/\sqrt{2\rho_{\mathcal{G}}}$ and unburdens the user from the need to specify a stepsize.

There is an intuitive interpretation of the updates of $\mathbf{Z}$, $\mathbf{x}$ and $\overline{\mathbf{x}}$ in Algorithm 1. Consider the first term $h(\mathbf{Z},\mathbf{x})\triangleq\langle\nabla\mathbf{x},\mathbf{Z}\rangle_{\mathrm{F}}$ in the saddlepoint formulation (19). Its gradient with respect to $\mathbf{Z}$ is given by $\nabla_{\mathbf{Z}}h(\mathbf{P},\mathbf{x})=\nabla\mathbf{x}$. Therefore the update of $\mathbf{Z}$ (steps 6 and 7) can be written in the form

$$\mathbf{Z}^{(k+1)}=\pi_{\mathcal{P}}\left(\mathbf{Z}^{(k)}+\sigma\nabla_{\mathbf{Z}}h(\mathbf{Z}^{(k)},\overline{\mathbf{x}}^{(k)}\right),$$

which is a gradient ascent step in $\mathbf{Z}$ with stepsize σ followed by a projection onto the constraint set $\mathcal{P}$. Furthermore, the gradient of h with respect to $\mathbf{x}$ equals $\nabla_{\mathbf{x}}h(\mathbf{Z},\mathbf{x})=-\operatorname{div}\mathbf{Z}$.

Algorithm 1 PDHG algorithm for solving (7) or (8)

input: $\mathbf{y}$, $\mathbf{W}$, τ, ε, $\mathbf{x}^{(0)}$, $\mathbf{Z}^{(0)}$

1: $\rho_{\mathcal{G}} = \max_i \sum_{j\in\mathcal{V}} \left(W_{ij}^2 + W_{ji}^2\right)$
2: $\sigma = \dfrac{1}{2\tau\rho_{\mathcal{G}}}$
3: $\overline{\mathbf{x}}^{(0)} = \mathbf{x}^{(0)}$
4: $k = 0$
5: **repeat**
6: $\quad \tilde{\mathbf{z}}_i = \mathbf{z}_i^{(k)} + \sigma\nabla_i\overline{\mathbf{x}}^{(k)}$
7: $\quad \mathbf{z}_i^{(k+1)} = \tilde{\mathbf{z}}_i / \max\{1, \|\tilde{\mathbf{z}}_i\|_2\}$
8: $\quad \tilde{\mathbf{x}} = \mathbf{x}^{(k)} + \tau \operatorname{div}\mathbf{Z}^{(k+1)}$
9: $\quad$ **if** length(ε) $= 1$ **then**
10: $\quad\quad c = 1 - \varepsilon/\sqrt{\sum_{i=1}^{M}(y_i - \tilde{x}_i)^2}$
11: $\quad\quad x_i^{(k)} = \begin{cases} \tilde{x}_i + c(y_i - \tilde{x}_i) & \text{if } i \le M \text{ and } c > 0 \\ \tilde{x}_i & \text{else} \end{cases}$
12: $\quad$ **else**
13: $\quad\quad r_i = y_i - \tilde{x}_i$
14: $\quad\quad x_i^{(k)} = \begin{cases} y_i - \varepsilon_i \operatorname{sign}(r_i) & \text{if } i \le M \text{ and } |r_i| > \varepsilon_i \\ \tilde{x}_i & \text{else} \end{cases}$
15: $\quad$ **end if**
16: $\quad \overline{\mathbf{x}}^{(k+1)} = 2\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$
17: $\quad k = k + 1$
18: **until** stopping criterion is satisfied

output: $\mathbf{x}^{(k)}$

Therefore the update of $\mathbf{x}$ can be written in the form

$$\mathbf{x}^{(k+1)} = \pi_{\mathcal{Q}_i}\left(\mathbf{x}^{(k)} - \tau\nabla_{\mathbf{x}} h(\mathbf{Z}^{(k+1)}, \mathbf{x}^{(k)})\right).$$

This is a gradient descent step in $\mathbf{x}$ with stepsize τ followed by a projection onto the constraint set $\mathcal{Q}_i$. Finally, the computation of $\overline{\mathbf{x}}^{(k+1)}$ in step 16 is a simple linear extrapolation based on the current and previous estimate of $\mathbf{x}$.

V. Distributed Implementation

In this section we use message passing and consensus schemes to develop a distributed implementation of Algorithm 1, which is summarized in Algorithm 2. With such a distributed implementation, even huge graphs can be handled efficiently by distributing and parallelizing the computation and memory requirements. For simplicity of exposition, we assume that there is a network of N separate computers, with one computer for each graph node $i \in \mathcal{V}$. The computational network has the same topology as the graph $\mathcal{G}$. This amounts to an implementation with maximum communication overhead but minimal computational and storage requirements for the processing units. It is straightforward to modify the implementation to a network of K computers, where each of the computers is dedicated to one of K disjoint subgraphs of $\mathcal{G}$. Algorithm 2 comprises all computation and communication tasks to be performed by node i. In contrast to the previously stated algorithms, it uses in-place computation (overwriting variables with new values) to minimize memory usage. Furthermore, the algorithms use a variable $Q \in \{1, 2\}$ to switch between projections onto the constraint sets $\mathcal{Q}_1$ and $\mathcal{Q}_2$ in (7) and (8), respectively.

In the distributed algorithm, messages are sent over the edges of the graph, i.e., node i sends messages to its children $\operatorname{ch}(i) \triangleq \{j \in \mathcal{V} : W_{ij} \neq 0\}$ and to its parents $\operatorname{pa}(i) \triangleq \{j \in \mathcal{V} : W_{ji} \neq 0\}$. We denote the set of all neighbors of node i by $\mathcal{N}(i) = \operatorname{ch}(i) \cup \operatorname{pa}(i)$. The gradient ascent and descent steps in Algorithm 1 work almost on a per-node basis, with the graph gradient $\nabla\bar{\mathbf{x}}^{(k)}$ and the divergence $\operatorname{div}\mathbf{Z}^{(k+1)}$ involving only neighboring nodes; the corresponding data (elements of $\bar{\mathbf{x}}^{(k)}$ and $\mathbf{Z}^{(k+1)}$) is obtained via message passing between neighboring computers rather than by accessing the local memory.

The remaining computational steps to be distributed are the projection onto $\mathcal{Q}_1$ (in case a total error constraint is used) and the computation of the global stepsize parameter $\rho_{\mathcal{G}}$. We emphasize that the projection onto $\mathcal{Q}_2$ is already performed on a per-node basis. With the projection onto $\mathcal{Q}_1$, the key non-local operation is the computation of the error norm $\sqrt{\sum_{i=1}^{M}(y_i - \tilde{x}_i)^2}$ (cf. step 10 in Algorithm 1). To obtain a fully distributed algorithm, we propose to use for that purpose a fast version of the average consensus algorithm (e.g., [49], [50]). With the definitions $b_i^{(0)} = (y_i - \tilde{x}_i)^2$, $i \le M$, and $b_i^{(0)} = 0$, $i > M$, the average consensus algorithm uses the consensus weights

$$u_{ij} \triangleq \frac{1}{\max\{d_i, d_j\} + 1} \quad \text{and} \quad \bar{u}_i = \sum_{j\in\mathcal{N}(i)} u_{ij}$$

to perform repeated local node updates according to

$$b_i^{(l)} = (1 - \bar{u}_i) b_i^{(l-1)} + \sum_{j\in\mathcal{N}(i)} u_{ij}\, b_j^{(l-1)}.$$

These updates require a message passing exchange of $\{b_j^{(l-1)}\}_{j\in\mathcal{N}(i)}$ between neighboring nodes in each iteration. The consensus iterations converge to the arithmetic mean,

$$\lim_{l\to\infty} b_i^{(l)} = \frac{1}{N}\sum_{i\in\mathcal{V}} b_i^{(0)} = \frac{1}{N}\sum_{i=1}^{M}(y_i - \tilde{x}_i)^2,$$

and hence for sufficiently large l an accurate approximation for $\|\mathbf{y} - \tilde{\mathbf{x}}\|_2$ is given by $\sqrt{N b_i^{(l)}}$. As a rule of thumb, the number of consensus iterations should be in the order of the graph diameter. The overall average consensus procedure is represented by step 26 of Algorithm 2.

It remains to find a distributed algorithm for the computation of the maximum graph degree $\rho_{\mathcal{G}}$. This can be achieved in a finite number of steps via a maximum consensus algorithm [51] that uses the initialization $\rho_i^{(0)} = d_i$ and performs repeated local maximizations according to

$$\rho_i^{(l)} = \max\left\{\rho_j^{(l-1)}\right\}_{j\in\mathcal{N}(i)\cup\{i\}},$$

thereby allowing the global maximum to propagate through the network. This procedure is represented by step 4 of Algorithm 2. Each maximum consensus step requires a message passing exchange of $\rho_i^{(l)}$ between neighboring computing nodes.

We close this section with a few remarks regarding the com-

Algorithm 2 Distributed PDHG algorithm

input: y_i, $\{W_{ij}\}_{j\in\mathrm{ch}(i)}$, $\{W_{ji}\}_{j\in\mathrm{pa}(i)}$, τ, ε_i, Q

initialize x_i, $\{Z_{ij}\}_{j\in\mathrm{ch}(i)}$
$\overline{x}_i \leftarrow x_i$
$d_i \leftarrow \sum_{j\in\mathrm{ch}(i)} W_{ij}^2 + \sum_{j\in\mathrm{pa}(i)} W_{ji}^2$
$\rho_{\mathcal{G}} \leftarrow$ **max-consensus**$\{d_1,\ldots,d_N\}$
$\sigma \leftarrow 1/(2\tau\rho_{\mathcal{G}})$
repeat
  $x_i^{\mathrm{old}} \leftarrow x_i$
  broadcast $\overline{x}_i$ to parents $\mathrm{pa}(i)$
  collect $\{\overline{x}_j\}_{j\in\mathrm{ch}(i)}$ from children
  for $j \in \mathrm{ch}(i)$ **do**
    $Z_{ij} \leftarrow Z_{ij} + \sigma(\overline{x}_j - \overline{x}_i)W_{ij}$
  end for
  $\zeta_i \leftarrow \sqrt{\sum_{j\in\mathrm{ch}(i)} Z_{ij}^2}$
  for $j \in \mathrm{ch}(i)$ **do**
    $Z_{ij} \leftarrow Z_{ij}/\max\{1,\zeta_i\}$
    send Z_{ij} to child node j
  end for
  collect $\{Z_{ji}\}_{j\in\mathrm{pa}(i)}$ from parents
  $x_i \leftarrow x_i + \tau\Big(\sum_{j\in\mathrm{ch}(i)} W_{ij}Z_{ij} - \sum_{j\in\mathrm{pa}(i)} W_{ji}Z_{ji}\Big)$
  if $Q = 1$ **then**
    if $i \le M$ **then**
      $b_i \leftarrow (y_i - x_i)^2$
    else
      $b_i \leftarrow 0$
    end if
    $\beta_i \leftarrow$ **average-consensus**$\{b_1,\ldots,b_N\}$
    $c_i \leftarrow 1 - \varepsilon/\sqrt{N\beta_i}$
    if $i \le M$ **and** $c_i > 0$ **then**
      $x_i \leftarrow x_i + c_i(y_i - x_i)$
    end if
  else if $Q = 2$ **then**
    if $i \le M$ **and** $|y_i - x_i| > \varepsilon_i$ **then**
      $x_i \leftarrow y_i - \varepsilon_i \operatorname{sign}(y_i - x_i)$
    end if
  end if
  $\overline{x}_i \leftarrow 2x_i - x_i^{\mathrm{old}}$
until stopping criterion is satisfied

output: x_i

plexity and performance of the distributed PDHG algorithm. The per-node complexity of the method is dominated by the computation and communication of Z_{ij}, x_i, and $\bar{x}_i$, and by the consensus stage for the error norm β_i (the other steps in the algorithm are local scalar multiplications and additions). All of these steps scale with the size of the neighbor set $\mathcal{N}(i)$. Since $\sum_{i\in\mathcal{V}} |\mathcal{N}(i)| = 2|\mathcal{E}|$, the complexity of one iteration of the algorithm scales linearly with the number of edges of the graph $\mathcal{G}$. Since the PDHG method converges at rate $\mathcal{O}(\frac{1}{k})$ we thus require $\mathcal{O}(\frac{1}{\delta}|\mathcal{E}|)$ operations to achieve a primal dual gap smaller than δ [46, Theorem 1].

The accuracy of the recovered graph signal is essentially the same as that of the centralized implementation. In fact, there are only two possible causes for differences in the outputs of the distributed and the centralized scheme:

1) In Algorithm 2, the maximum graph degree and the total residual error for the $\mathcal{Q}_1$ projection are computed using consensus schemes. The accuracy of the results can be controlled via the number of consensus iterations, which in turn affects the computational complexity of the overall method.
2) The message passing scheme for exchanging the iterates $\bar{x}_i$ and Z_{ij} (and the consensus variables) between neighboring computer nodes may be affected by transmission and quantization errors. These errors are controlled by designing suitable message compression and coding schemes.

A more detailed analysis of these error sources and the design of suitable communication protocols is left for future work.

VI. Numerical Experiments

In this section, we provide numerical simulations to illustrate the graph signal recovery performance of our algorithm and compare it to recent state-of-the-art methods that build on different smoothness models (cf. Subsection II-C). More specifically, we show graph signal reconstruction results obtained with the following five algorithms:

1) Our primal-dual method for isotropic total variation, i.e., Algorithm 1 based on a total error constraint with step size $\sigma = \tau = 1/\sqrt{2\rho_{\mathcal{G}}}$ (labeled "isotropic").
2) An adaption of our primal-dual algorithm using the anisotropic TV $\|\hat{\mathbf{x}}\|_{\mathrm{TV}}^A$ with step size $\sigma = \tau = 1/\sqrt{2\rho_{\mathcal{G}}}$) (labeled "anisotropic").
3) The graph signal inpainting scheme from [22], [29] but using the graph variation (6) with $p = 1$ instead of $p = 2$, solved via a primal-dual algorithm with step size $\sigma = \tau = 1/2$ (labeled "graph variation").
4) Graph signal recovery via [26, Algorithm 2] using the graph Laplacian $\mathbf{L}$ as kernel matrix (labeled "Tikhonov" since it corresponds to Tikhonov regularization with regularization parameter $\gamma \to 0$).
5) Recovery via the kernel method described in [24] (implemented again via [26, Algorithm 2]) using a (non-sparse) diffusion kernel with σ^2 equal to 8 times the reciprocal of the maximum eigenvalue of the Laplacian (labeled "kernel").

We point out that there is a sign error in [26, Algorithm 2] that we corrected in our simulations.

With the first three approaches we stopped the primal-dual iterations as soon as the relative change between two successive graph signal estimates was small enough, i.e.,

$$\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_2 \le \epsilon \|\mathbf{x}^{(k-1)}\|_2. \tag{32}$$

In the noisy regime, regularized formulations of methods 4) and 5) could be obtained via [26, Algorithm 1]. However, since

an appropriate choice of the regularization parameter depends on the noise level in a nontrivial way, we here only considered the primal-dual variation minimization schemes 1), 2), and 3).

A. Synthetic Data

We first show results from numerical experiments with synthetic data.

1) Graph and Signal Models: To investigate the difference between the isotropic and anisotropic total variation we used two random models for the graph $\mathcal{G}$ and the graph signal $\mathbf{x}$. In both models we partitioned $N = 2000$ nodes into 10 disjoint subsets $\mathcal{V}_r$ of equal size $|\mathcal{V}_r| = 200$, $r = 1, \ldots, 10$. The partitions represent clusters (communities) in which the nodes are well connected and carry identical graph signal values, i.e., $x_i = \xi_r$ for $i \in \mathcal{V}_r$. The cluster signal values ξ_r were chosen randomly according to independent standard Gaussian distributions. The undirected edges of the graph were placed randomly, with two nodes within the same partition $\mathcal{V}_r$ being connected with a high probability of $P_{\rm i} = 0.2$ (all edge weights were set to $W_{ij} = 1$).

Model $\mathcal{I}$. This model is supposed to be matched to the *i*sotropic total variation. To this end, we randomly selected $|\mathcal{V}_r|/20 = 10$ boundary nodes from each cluster. Two boundary nodes in different clusters were connected by an edge with probability $P_{\rm b} = 0.5$. Smooth transitions of the graph signal along the boundary nodes in different clusters were enforced by running a single iteration of (standard) average consensus [50], [52] with generalized MH weights (this leaves the signal values on non-boundary nodes inside the clusters unchanged). As a result, the nonzero elements of the gradient matrix $\nabla\mathbf{x}$ are concentrated in the rows (columns) corresponding to the boundary nodes, leading to a small number of non-zero local gradients (whose ℓ_2 norm is small due to the smooth transitions across boundaries).

Model $\mathcal{A}$. This model is tailored to the *a*nisotropic total variation. Specifically, we induced a sparse nonzero pattern in the global graph signal gradient by connecting nodes in different clusters uniformly at random with probability $P_{\rm o} \ll P_{\rm i}$. Unless stated otherwise the inter-cluster edge probability was chosen as $P_{\rm o} \approx 3.7 \cdot 10^{-4}$, which amounts to $\eta = 60$ times more edges on average within clusters than between clusters. Note that the graph signal changes abruptly across these inter-cluster edges. Since the signal values are identical within clusters, each inter-cluster edge corresponds to two nonzero elements in the global signal gradient matrix.

2) Results: Unless stated otherwise, the graph signal was sampled at $M = 600$ randomly chosen nodes ($M/N = 0.3$). We used the stopping criterion (32) with $\epsilon = 10^{-3}$. Since the average graph signal power is equal to 1, we define the the signal-to-noise ratio (SNR) as $\mathrm{SNR} = 1/\sigma_u^2$, where σ_u^2 is the average power of the noise u_i in (1). The recovery performance is quantified in terms of the normalized mean squared reconstruction error (NMSE) $e^2 = \frac{1}{N} \mathrm{E}\{\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2\}$. All results described below have been obtained by averaging over 100 independent realizations of the graph signal, the graph topology, the sampling set, and the noise.

Experiment 1. We first study the impact of the sampling rate M/N on the recovery performance in the noise-free case (SNR $= \infty$) by varying the number of samples in the range $M = 50, \ldots, 500$. The three signal-variation based primal-dual algorithms ("isotropic", "anisotropic", "graph variation") used a total error level of $\varepsilon = 0$. Figs. 1(a) and (b) show the NMSE e^2 versus the sampling rate M/N for model $\mathcal{I}$ and model $\mathcal{A}$, respectively. It is seen that the recovery performance of all schemes improves with increasing sampling rate for both graph signal models. However, our total variation schemes outperform the three state-of-the art methods by orders of magnitude, with "isotropic" performing best on model $\mathcal{I}$ and "anisotropic" performing best on model $\mathcal{A}$ at sampling rates below 12%.

Experiment 2. Fig. 1(b) shows that at sampling rates larger than 12% "isotropic" performs better than "anisotropic" even on model $\mathcal{A}$. This is due to the fact that there are very few inter-cluster edges ($\eta = 60$ times fewer than intra-cluster edges). Fig. 1(c) shows the reconstruction error versus η (the ratio of the average number of intra-cluster and inter-cluster edges, which was varied via the inter-cluster edge probability $P_{\rm o}$) for a sampling rate of $M/N = 0.3$ on model $\mathcal{A}$. It is seen that anisotropic total variation reconstruction is almost unaffected by the inter-cluster connectivity (a degradation occurs only for η below 10) and substantially outperforms all other methods. Reconstruction based on isotropic total variation deteriorates noticeably for small η and eventually becomes even worse then kernel-based reconstruction.

Experiment 3. Next we compare "isotropic", "anisotropic", and "graph variation" with i.i.d. zero-mean Gaussian measurement noise of variance σ_u^2. The total empirical ℓ_2 error was controlled with $\varepsilon = \sigma_u \sqrt{M}$. The number of samples was again $M = 600$. Figs. 1(d) and (e) show the reconstruction NMSE versus SNR for models $\mathcal{I}$ and $\mathcal{A}$, respectively. It is seen that for model $\mathcal{I}$ again "isotropic" performs best and for model $\mathcal{A}$ "anisotropic" performs best. The "graph variation" method performs way worse than the two total variation methods and even appears to saturate at high SNR.

Experiment 4. We repeat experiment 3 with model $\mathcal{I}$, but this time half of the samples are acquired noiseless, i.e., $y_i = x_i$, $i = 1, \ldots, M/2$, whereas the other half is affected by independent noise with uniform distribution on the interval $[-b, b]$. The variance of the noise equals $\sigma_u^2 = b^2/3$. This particular noise model lends itself to a per-node constraint on the empirical error, i.e., we set $\varepsilon_i = 0$, $i = 1, \ldots, M/2$, and $\varepsilon_i = \sigma_u = \frac{b}{\sqrt{3}}$, $i = M/2 + 1, \ldots, M$. The reconstruction NMSE as a function of SNR is depicted in Fig. 1(f). Isotropic total variation reconstruction is superior on model $\mathcal{I}$ also with per-node error constraints on this particular noise model. We found that for this noise model reconstruction with a global error constraint (not shown here) is worse by about 6 dB.

B. Real-World Data

The Amazon co-purchase dataset is a publicly accessible collection of product information from the online retailer Amazon [53]. It contains a list of different products, their average user rating, and, for each product, a list of products that are frequently co-purchased. The average user rating is an

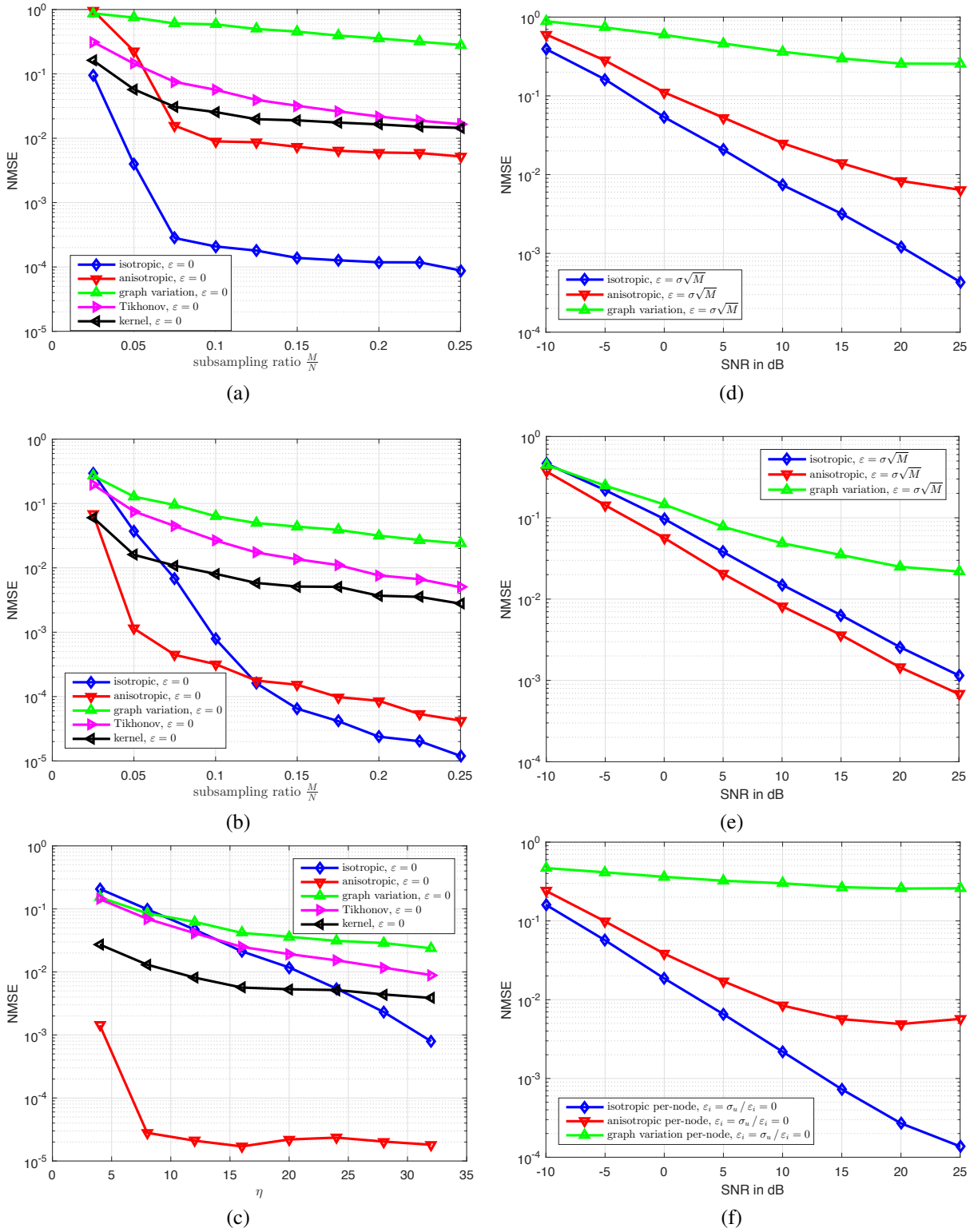

Fig. 1: Performance of different graph signal reconstruction schemes on synthetic graph signals: (a) NMSE versus sampling rate M/N on model $\mathcal{I}$ without noise; (b) NMSE versus sampling rate M/N on model $\mathcal{A}$ without noise; (c) NMSE versus inter-cluster connectivity on model $\mathcal{A}$ without noise; (d) NMSE versus SNR with Gaussian noise on model $\mathcal{I}$; (e) NMSE versus SNR with Gaussian noise on model $\mathcal{A}$; (f) NMSE versus SNR on model $\mathcal{I}$ with uniform noise on half of the samples.

integer or half integer between 1 and 5; for products without actual user ratings the value is set to 0.

We generated a preliminary graph in which each node is identified with one product and there is an edge (with weight 1) from node i to node j if product j is co-purchased with product i. The value of the graph signal at node i equals the average rating of product i. The rationale is that co-purchased products tend to have similar quality and thus similar ratings. This preliminary graph is composed of one large connected subgraph that contains about 90% of the products and many disconnected small subgraphs. For our experiments, we only retained the large subgraph and discarded all other components. The resulting graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ has $N = 334\,859$ nodes and $1\,851\,720$ edges. The distribution of the node degrees d_i and of the average rating x_i is shown in Figures 2(a) and (b). The majority of nodes is seen to have degrees below 10 and very few nodes have degrees larger than 100. Regarding the ratings, almost 15% of the products have no actual rating, while close to 30% have a maximal rating of 5.

We chose $M = 28600$ samples uniformly at random from the set of nodes with nonzero rating and attempted to recover the remaining known ratings as well the unknown ratings. No sampling noise was considered. We compared our PDHG algorithm for isotropic total variation with 5000 iterations and stepsize parameters $\tau = \sigma = 1/\sqrt{2\rho_{\mathcal{G}}}$ to algorithms 4) and 5) from the beginning of this section, i.e., Tikhonov regularization and recovery via a diffusion kernel (due to the large problem dimension, we had to use a non-sparse third-order Chebyshev series approximation [14], [24]). Both with Tikhonov and diffusion kernel regularization, we preconditioned the data by subtracting the mean. The dimensionality of the problem prevents direct computation of the closed-form solution for both regularizers; we thus used the LSQR method [54] to compute approximate solutions of the corresponding linear system of equations. The per-iteration complexity of the LSQR method scales linearly with the number of nonzero elements in the kernel matrix, which (due to fill-in effects) is substantially larger for the diffusion kernel than for the sparse Laplacian kernel.

The output of all algorithms was rounded to the nearest (half-)integer. To assess the recovery accuracy, we considered the magnitude of the recovery error for the unobserved products with known rating. A histogram of the magnitude of the recovery error averaged over 10 random choices of the sampling set is shown in Figure 2(c) for all three recovery methods. PDHG achieves the largest fraction of exact recoveries, i.e., 34% versus 29% (kernel) and 30% (Tikhonov). For all three methods, the percentage of estimated ratings that deviate by at most 0.5 from the true rating are similar, namely 76% (PDHG), 83% (kernel), and 80% (Tikhonov), respectively. The slightly higher percentage of the kernel method comes at the price of a larger complexity and the lack of an efficient distributed implementation (since the kernel matrix is no longer sparse).

M/N	NESTA	PDHG
0.01	1712	1463
0.05	1132	423
0.1	722	306
0.5	484	194

TABLE I: Comparison of average number of iterations required by NESTA and PDHG at different sampling ratios M/N.

Next, we compare the convergence speed of our PDHG method with the NESTA algorithm for total variation minimization proposed in our previous work [2]. The stopping criterion of the PDHG algorithm was (32) with $\epsilon = 2 \cdot 10^{-3}$. The smoothing parameter for NESTA was chosen as $\mu = 1$. Since NESTA has nearly identical per-iteration complexity as our PDHG algorithm, the complexity of both schemes can be compared in terms of number of iterations. Specifically, we measured how many iterations it takes for NESTA to find a graph signal whose total variation is at least as small as that of the PDGH output. We averaged the required number of iterations over 10 independent realizations of the sampling set for four different sampling ratios. The results are summarized in Table I. It is seen that PDHG runs substantially faster than NESTA at all sampling ratios shown. The convergence behavior of both algorithms is illustrated in Figure 2(d) for one realization of the sampling set. By changing the smoothing parameter to $\mu = 0.1$, the accuracy of NESTA can be improved at the cost of a further decrease in convergence speed.

VII. Conclusions

In this paper, we considered the problem of reconstructing smooth graph signals from samples taken on a small number of graph nodes. Smoothness is enforced by minimizing the total variation in the recovered graph signal while controlling the empirical error between the recovered graph signal and the measurements at the sampling nodes. We considered constraint sets that reflect the total empirical error (l_2 norm) and a per-node error (weighted l_∞ norm). The latter is particularly useful in scenarios with different levels of measurement noise. Even though the total variation is a non-smooth function, we derived a fast algorithm based on the PDHG method. Furthermore, in order to render the algorithm applicable to huge graphs, we devised a distributed implementation that requires only a few message updates between neighboring nodes in the graph. We illustrated the performance of our method on synthetic data and on the Amazon co-purchasing dataset. For the Amazon co-purchasing dataset we obtained reconstruction accuracy comparable to kernel-based methods at favorable complexity. Our numerical experiments indicated that PDHG converges substantially faster than the NESTA scheme used in our previous work. We further observed that total variation based reconstruction is particularly well suited to cluster/community graphs, where it outperforms state-of-the-art methods by orders of magnitude. We also found that it is favorable when the graph structure is well matched to the graph signal's smoothness structure. In practice, this can be achieved by exploiting known features of the graph nodes to construct the graph topology and then use this graph for the reconstruction of other features. For example, follower relationships between users in social networks may be exploited

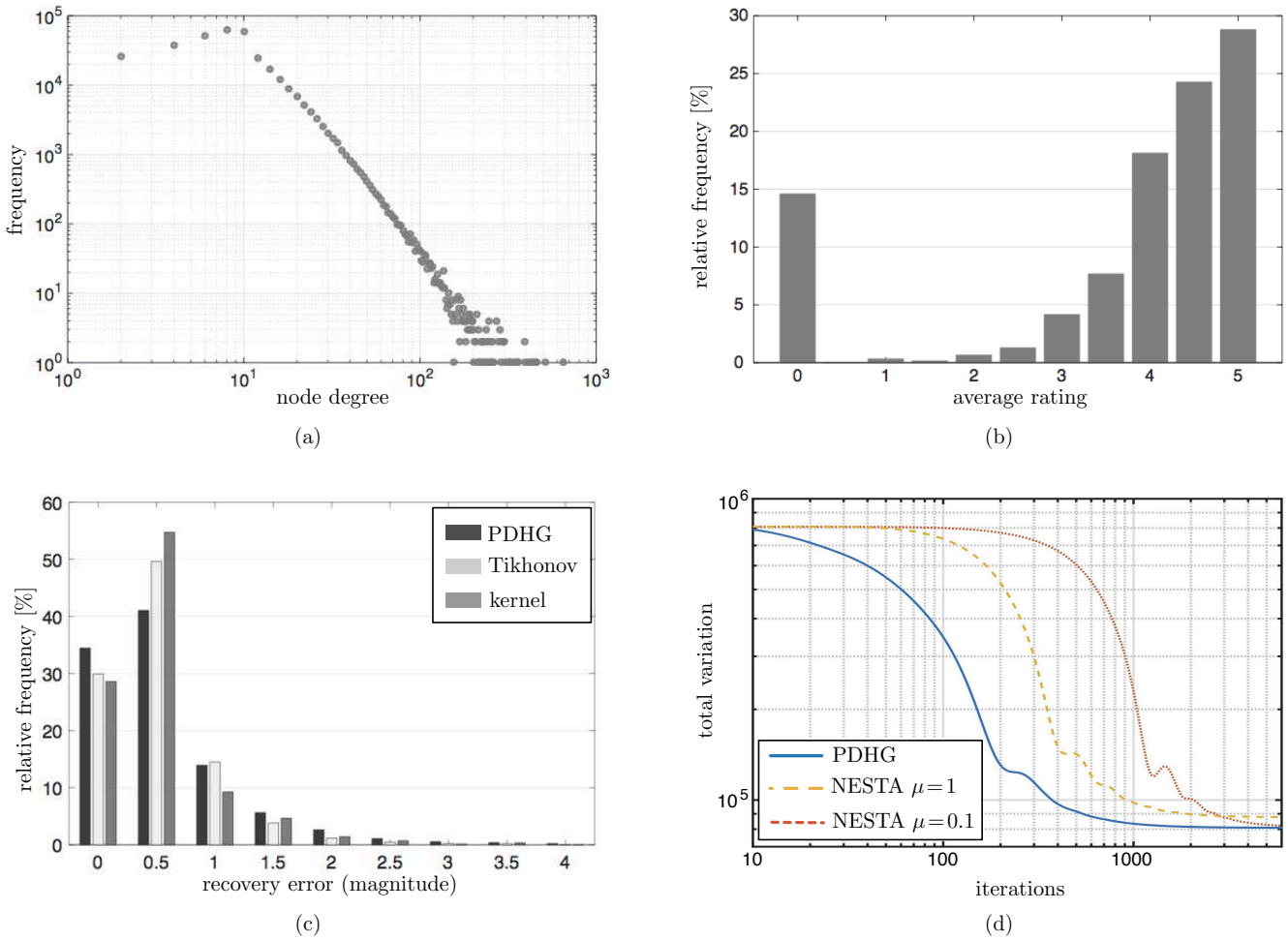

Fig. 2: Amazon co-purchasing dataset: (a) empirical distribution of node degrees; (b) histogram of product rating; (c) histogram of the recovery error magnitude achieved with PDHG, Tikhonov regularization, and kernel smoothing; (d) convergence of PDHG and NESTA.

to construct a directed graph which in turn can be used to recover graph signals capturing certain user preferences like political partisanship, musical taste, or preferred reading.

VIII. Acknowledgement

The authors are grateful to Alexander Jung for initiating this line of work, for pointing out [46], and for comments that helped improve the presentation. We also thank the reviewers for numerous constructive comments that led to substantial improvements of the paper.

Appendix A
Saddle Point Formulation

Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be finite dimensional Hilbert spaces, both defined over the real numbers. Let $f : \mathcal{H}_2 \to (-\infty, \infty]$ and $g : \mathcal{H}_1 \to (-\infty, \infty]$ be two lower semi-continuous convex functions and let $\mathbf{B} : \mathcal{H}_1 \to \mathcal{H}_2$ be a bounded linear operator. We consider convex optimization problems of the form

$$\min_{\mathbf{x}\in\mathcal{H}_1} f(\mathbf{B}\mathbf{x}) + g(\mathbf{x}). \tag{33}$$

Consider the convex conjugate of the function f, defined as

$$f^*(\mathbf{z}) \triangleq \sup_{\mathbf{x}\in\mathcal{H}_2} \langle \mathbf{x}, \mathbf{z}\rangle_{\mathcal{H}_2} - f(\mathbf{x}). \tag{34}$$

Since f is convex and lower semi-continuous we have

$$f(\mathbf{x}) = f^{**}(\mathbf{x}) = \sup_{\mathbf{z}\in\mathcal{H}_2} \langle \mathbf{x}, \mathbf{z}\rangle_{\mathcal{H}_2} - f^*(\mathbf{z}).$$

Using this relation to replace $f(\mathbf{B}\mathbf{x})$ in (33), we obtain the saddle point formulation

$$\min_{\mathbf{x}\in\mathcal{H}_1} \sup_{\mathbf{z}\in\mathcal{H}_2} \langle \mathbf{B}\mathbf{x}, \mathbf{z}\rangle_{\mathcal{H}_2} - f^*(\mathbf{z}) + g(\mathbf{x}). \tag{35}$$

Appendix B
Primal-Dual Algorithm

The saddle-point problem (35) is assumed to have at least one optimal point. The PDHG method for solving the generic saddlepoint problem (35) consists in alternatingly performing a proximal ascent step in $\mathbf{z}$ and a proximal descent step in $\mathbf{x}$ (see Algorithm 3). The corresponding proximal operator is

Algorithm 3 PDHG algorithm for solving (35)

input: f^*, g, $\mathbf{B}$, $\tau > 0$, $\sigma > 0$, $(\mathbf{x}^{(0)}, \mathbf{z}^{(0)}) \in \mathcal{H}_1 \times \mathcal{H}_2$

1: $\overline{\mathbf{x}}^{(0)} = \mathbf{x}^{(0)}$

2: $k = 0$

3: **repeat**

4: $\quad \tilde{\mathbf{z}} = \mathbf{z}^{(k)} + \sigma \mathbf{B} \overline{\mathbf{x}}^{(k)}$

5: $\quad \mathbf{z}^{(k+1)} = \underset{\mathbf{z} \in \mathcal{H}_2}{\arg\min} \frac{1}{2\sigma} \|\mathbf{z} - \tilde{\mathbf{z}}\|^2_{\mathcal{H}_2} + f^*(\mathbf{z})$

6: $\quad \tilde{\mathbf{x}} = \mathbf{x}^{(k)} - \tau \mathbf{B}^* \mathbf{z}^{(k+1)}$

7: $\quad \mathbf{x}^{(k+1)} = \underset{\mathbf{x} \in \mathcal{H}_1}{\arg\min} \frac{1}{2\tau} \|\mathbf{x} - \tilde{\mathbf{x}}\|^2_{\mathcal{H}_1} + g(\mathbf{x})$

8: $\quad \overline{\mathbf{x}}^{(k+1)} = 2\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$

9: $\quad k = k + 1$

10: **until** stopping criterion is satisfied

output: $(\mathbf{x}^{(k)}, \mathbf{z}^{(k)})$

defined as

$$\text{prox}_{\tau h}(\mathbf{x}) = \underset{\mathbf{x}'}{\arg\min} \frac{1}{2\tau} \|\mathbf{x}' - \mathbf{x}\|^2 + h(\mathbf{x}'), \qquad (36)$$

where τ is a stepsize parameter. The proximal operator is applied in steps 5 and 7 of Algorithm 3 with $h = f^*$ and $h = g$, respectively. For possible interpretations and properties of the proximal operator we refer the reader to [55].

For $\tau\sigma\|\mathbf{B}\|^2_{\text{op}} < 1$ Algorithm 3 is guaranteed to converge with an (ergodic) rate of convergence of $\mathcal{O}(1/k)$ for the objective function [46, Theorem 1] (k is the iteration index). In [38], Nesterov showed that this convergence speed is optimal for convex optimization problems of the type (35).

The particular choice of the stepsize parameters τ and σ can heavily influence the actual convergence speed of the PDHG algorithm. An adaptive version of the PDHG algorithm that automatically tunes the stepsize parameters τ and σ can be found in [56]. This adaptive version tries to ensure that the l_1-norm of the primal and dual residuals have roughly the same magnitude in each iteration. We do not pursue this variant and rather stick to constant stepsize τ and σ since the extra calculation of the l_1 norms would significantly complicate our distributed implementation presented in Section V.

There also exist accelerated versions of the PDHG algorithm for the case where g or f^* in (35) is strongly convex, see [46, Section 5]. Unfortunately, neither $\chi_{\mathcal{Q}}(\mathbf{x})$ nor $\chi_{\mathcal{P}}(\mathbf{Z})$ in (19) is strongly convex, and therefore these accelerated schemes are not applicable to graph signal recovery.

REFERENCES

[1] 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.*, Edinburgh, UK, July 2016, pp. 1–6.

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

[3] A. Katal, M. Wazid, and R. H. Goudar, "Big data: Issues, challenges, tools and good practices," in *Proc. Int. Conf. Contemporary Computing (IC3)*, Noida, India, Aug. 2013, pp. 404–409.

[4] 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 Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[5] 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.*, vol. 31, no. 5, pp. 80–90, Sept. 2014.

[6] ——, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[7] S. K. Narang and A. Ortega, "Perfect reconstruction two-channel wavelet filter banks for graph structured data," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2786–2799, June 2012.

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

[9] P. S. Dodds and D. J. Watts, "Universal behavior in a generalized model of contagion," *Physical Review Letters*, vol. 92, no. 21, p. 218701, May 2004.

[10] S. Aral and D. Walker, "Identifying influential and susceptible members of social networks," *Science*, vol. 337, no. 6092, pp. 337–341, July 2012.

[11] 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*, Palo Alto, CA, Mar. 2006, pp. 100–107.

[12] 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.*, vol. 12, no. 1, pp. 290–310, Oct. 2006.

[13] W. Xing and A. Ghorbani, "Weighted pagerank algorithm," in *Proc. Annual Conf. Commun. Networks and Services Research*, Fredericton, Canada, May 2004, pp. 305–314.

[14] S. K. Narang, A. Gadde, E. Sanou, and A. Ortega, "Localized iterative methods for interpolation in graph structured data," in *Proc. IEEE GlobalSIP*, Austin, TX, Dec. 2013, pp. 491–494.

[15] S. K. Narang, A. Gadde, and A. Ortega, "Signal processing techniques for interpolation in graph structured data," in *Proc. IEEE ICASSP*, Vancouver, BC, Canada, May 2013, pp. 5445–5449.

[16] A. Gadde and A. Ortega, "A probabilistic interpretation of sampling theory of graph signals," in *Proc. IEEE ICASSP*, Brisbane, Australia, Apr. 2015, pp. 3257–3261.

[17] S. Chen, S. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6510–6523, Dec. 2015.

[18] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, "Signals on graphs: uncertainty principle and sampling," *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4845–4860, Sept. 2016.

[19] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3775–3789, July 2016.

[20] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Sampling of graph signals with successive local aggregations," *IEEE Trans. Signal Process.*, vol. 64, no. 7, pp. 1832–1843, Apr. 2016.

[21] X. Wang, P. Liu, and Y. Gu, "Local-set-based graph signal reconstruction," *IEEE Trans. Signal Process.*, vol. 63, no. 9, pp. 2432–2444, May 2015.

[22] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević, "Signal recovery on graphs: Variation minimization," *IEEE Trans. Signal Process.*, vol. 63, no. 17, pp. 4609–4624, Sep. 2015.

[23] G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, "Random sampling of bandlimited signals on graphs," *Applied and Computational Harmonic Analysis*, 2016.

[24] D. Romero, M. Ma, and G. B. Giannakis, "Kernel-based reconstruction of graph signals," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 764–778, Feb. 2017.

[25] O. Chapelle, B. Scholkopf, and A. Zien, Eds., *Semi-Supervised Learning*. MIT Press, 2006.

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

[27] ——, "Tikhonov regularization and semi-supervised learning on large graphs," in *Proc. IEEE ICASSP*, Montreal, Canada, May 2004.

[28] S. Chen, F. Cerda, P. Rizzo, J. Bielak, J. H. Garrett, Jr., and J. Kovačević, "Semi-supervised multiresolution classification using adaptive graph filtering with application to indirect bridge structural health monitoring," *IEEE Trans. Signal Process.*, vol. 62, no. 11, pp. 2879–2893, June 2014.

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

[30] F. Mahmood, N. Shahid, U. Skoglund, and P. Vandergheynst, "Adaptive graph-based total variation for tomographic reconstructions," *arXiv:1610.00893*, 2016.
[31] X. Bresson, T. Laurent, D. Uminsky, and J. Von Brecht, "Multiclass total variation clustering," in *In Advances in Neural Information Processing Systems 26*, Lake Tahoe, Nevada, USA, Dec. 2013, pp. 1421–1429.
[32] G. Gilboa and S. Osher, "Nonlocal operators with applications to image processing," *Multiscale Model. Simul.*, vol. 7, no. 3, pp. 1005–1028, Nov. 2008.
[33] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3–5, pp. 75–174, 2010.
[34] R. I. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete structures," in *Proc. Int. Conf. Machine Learning*, Sydney, Australia, Jul. 2002, pp. 315–322.
[35] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *Learning Theory and Kernel Machines: Proc. COLT/Kernel*, B. Schölkopf and M. K. Warmuth, Eds. Springer, 2003, pp. 144–158.
[36] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, Mar. 2009.
[37] Y. Zhu, "An Augmented ADMM Algorithm With Application to the Generalized Lasso Problem," *J. Comput. Graph. Statist.*, vol. 26, no. 1, pp. 195–204, Feb. 2017.
[38] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Program.*, vol. 103, no. 1, Ser. A, pp. 127–152, Dec. 2005.
[39] ——, *Introductory lectures on convex optimization*, ser. Applied Optimization. Boston, MA: Kluwer Academic Publishers, 2004, vol. 87.
[40] A. Chambolle and T. Pock, "An introduction to continuous optimization for imaging," *Acta Numer.*, vol. 25, pp. 161–319, May 2016.
[41] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, Nov. 2009.
[42] A. Chambolle, "An algorithm for total variation minimization and applications," *J. Mathematical Imaging and Vision*, vol. 20, no. 1-2, pp. 89–97, Jan. 2004.
[43] S. Becker, J. Bobin, and E. J. Candès, "NESTA: a fast and accurate first-order method for sparse recovery," *SIAM J. Imaging Sci.*, vol. 4, no. 1, pp. 1–39, 2011.
[44] M. Zhu and T. Chan, "An efficient primal-dual hybrid gradient algorithm for total variation image restoration," UCLA CAM Report 08-34, Tech. Rep., 2008.
[45] E. Esser, X. Zhang, and T. F. Chan, "A general framework for a class of first order primal-dual algorithms for TV minimization," UCLA CAM Report 09-67, Tech. Rep., 2009.
[46] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imaging Vision*, vol. 40, no. 1, pp. 120–145, 2011.
[47] A. Beck and M. Teboulle, "A fast dual proximal gradient algorithm for convex minimization and applications," *Oper. Res. Lett.*, vol. 42, no. 1, pp. 1–6, Oct. 2014.
[48] D. Kim and J. A. Fessler, "Fast dual proximal gradient algorithms with rate $O(1/k^{1.5})$ for convex minimization," *arXiv:1609.09441*, 2016.
[49] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.
[50] V. Schwarz, G. Hannak, and G. Matz, "On the convergence of average consensus with generalized Metropolis-Hasting weights," in *Proc. IEEE ICASSP*, Florence, Italy, May 2014, pp. 5442–5446.
[51] D. Bauso, L. Giarré, and R. Pesenti, "Non-linear protocols for optimal distributed consensus in networks of dynamic agents," *Systems & Control Letters*, vol. 55, no. 11, pp. 918–928, July 2006.
[52] L. Xiao, S. Boyd, and S. Lall, "Distributed average consensus with time-varying metropolis weights," *Automatica*, 2006.
[53] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.
[54] C. C. Paige and M. A. Saunders, "LSQR: an algorithm for sparse linear equations and sparse least squares," *ACM Trans. Math. Software*, vol. 8, no. 1, pp. 43–71, June 1982.
[55] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, Jan. 2014.
[56] T. Goldstein, M. Li, X. Yuan, E. Esser, and R. Baraniuk, "Adaptive primal-dual hybrid gradient methods for saddle-point problems," *arXiv:1305.0546*, 2013.

Peter Berger received the Dipl.-Ing. degree in technical mathematics from University of Innsbruck, Austria, in 2011 and his Dr. degree in mathematics from University of Vienna, Austria, in 2015. Since 2015, he has been with the Institute of Telecommunications, Vienna University of Technology, with a intermediate research period at Department of Computer Science, Aalto University, Finland. His research interests are in the areas of signal processing, inverse problems, sampling theory, and numerical analysis.

Gabor Hannak received his M.Sc. degree in electrical engineering and information technology from the Vienna University of Technology, Austria, in August 2013, with major in telecommunications. In October 2013 he joined the Institute of Telecommunications, Vienna University of Technology, and is currently a Ph.D. candidate. His research interests include Bayesian compressed sensing and signal processing over graphs.

Gerald Matz received the Dipl.-Ing. (1994) and Dr. techn. (2000) degrees in Electrical Engineering and the Habilitation degree (2004) for Communication Systems from Vienna University of Technology, Austria. He currently holds a tenured Associate Professor position with the Institute of Telecommunications, Vienna University of Technology. He has held visiting positions with the Laboratoire des Signaux et Systèmes at Ecole Supérieure d'Electricité (France, 2004), the Communication Theory Lab at ETH Zurich (Switzerland, 2007), and with Ecole Nationale Supérieure d'Electrotechnique, d'Electronique, d'Informatique et d'Hydraulique de Toulouse (France, 2011).

Prof. Matz has directed or actively participated in several research projects funded by the Austrian Science Fund (FWF), by the Viennese Science and Technology Fund (WWTF), and by the European Union. He has published some 200 scientific articles in international journals, conference proceedings, and edited books. He is co-editor of the book *Wireless Communications over Rapidly Time-Varying Channels* (New York: Academic, 2011). His research interests include wireless networks, statistical signal processing, information theory, and big data.

Prof. Matz served as as a member of the *IEEE SPS Technical Committee on Signal Processing Theory and Methods* and of the *IEEE SPS Technical Committee on Signal Processing for Communications and Networking*. He was an Associate Editor of the *IEEE Transactions on Information Theory* (2013–2015), of the *IEEE Transactions on Signal Processing* (2006-–2010), of the EURASIP journal *Signal Processing* (2007—2010), and of the *IEEE Signal Processing Letters* (2004–2008). He was Technical Program Chair of Asilomar 2016, Technical Program Co-Chair of EUSIPCO 2004, Technical Area Chair for "MIMO Communications and Signal Processing" at Asilomar 2012, and Technical Area Chair for "Array Processing" at Asilomar 2015. He has been a member of the Technical Program Committee of numerous international conferences. In 2006 he received the Kardinal Innitzer Most Promising Young Investigator Award. He is an IEEE Senior Member and a member of the ÖVE.