Discrete Signal Processing on Graphs: Frequency Analysis

Aliaksei Sandryhaila, *Member, IEEE* and José M. F. Moura, *Fellow, IEEE*

Abstract—Signals and datasets that arise in physical and engineering applications, as well as social, genetics, biomolecular, and many other domains, are becoming increasingly larger and more complex. In contrast to traditional time and image signals, data in these domains are supported by arbitrary graphs. Signal processing on graphs extends concepts and techniques from traditional signal processing to data indexed by generic graphs. This paper studies the concepts of low and high frequencies on graphs, and low-, high- and band-pass graph signals and graph filters. In traditional signal processing, these concepts are easily defined because of a natural frequency ordering that has a physical interpretation. For signals residing on graphs, in general, there is no obvious frequency ordering. We propose a definition of total variation for graph signals that naturally leads to a frequency ordering on graphs and defines low-, high-, and band-pass graph signals and filters. We study the design of graph filters with specified frequency response, and illustrate our approach with applications to sensor malfunction detection and data classification.

Keywords: Signal processing on graphs, total variation, low pass, high pass, band pass, filter design, regularization.

I. Introduction

Signals indexed by graphs arise in many applications, including the analysis of preferences and opinions in social and economic networks [1], [2], [3]; research in collaborative activities, such as paper co-authorship and citations [4]; topics and relevance of documents in the World Wide Web [5], [6]; customer preferences for service providers; measurements from sensor networks; interactions in molecular and gene regulatory networks; and many others.

Signal processing on graphs extends the classical discrete signal processing (DSP) theory for time signals and images [7] to signals indexed by vertices of a graph. There are two basic approaches to signal processing on graphs. The first one uses the graph Laplacian matrix as its basic building block (see a recent review [8] and references therein). The second approach adopts the adjacency matrix of the underlying graph as its fundamental building block [9], [10], [11]. Both frameworks define fundamental signal processing concepts on graphs, but the difference in their foundation leads to different definitions and techniques for signal analysis and processing.

Methods for Laplacian-based graph signal analysis emerged from research on the spectral graph theory [12] and manifold discovery and embedding [13], [14]. Implicitly or explicitly, in these works graphs discretize continuous high-dimensional manifolds from $\mathbb{R}^M$: graph vertices sample a manifold and connect to nearest neighbors as determined by their geodesic distances over the underlying manifold. In this setting, the graph Laplacian operator is the discrete counterpart to the continuous Laplace-Beltrami operator on a manifold [12], [15].

This connection is propagated conceptually to Laplacian-based methods for signal processing on graphs. For example, the graph Fourier transform defined and considered in [8], as well as [16], [17], [18], [19], [20], [21], expands graph signals in the eigenbasis of the graph Laplacian. This parallels the classical Fourier transform that expands signals into the basis of complex exponentials that are eigenfunctions of the one-dimensional Laplace operator – the negative second order derivative operator [8]. The frequencies are the eigenvalues of the Laplace operator. Since the operator is symmetric and positive semi-definite, graph frequencies are real-valued and hence totally ordered. So, just like for time signals, the notions of low and high frequencies are easily defined in this model. However, due to the symmetry and positive semi-definiteness of the operator, the Laplacian-based methods are only applicable to undirected graphs with real, non-negative weights.

In [9], [10], [11] we take a different route. Our approach is motivated by the algebraic signal processing (ASP) theory introduced in [22], [23], [24], [25], [26]; see also [27], [28], [29], [30], [31] for additional developments. In ASP, the shift is the elementary non-trivial filter that generates, under an appropriate notion of shift invariance, all linear shift-invariant filters for a given class of signals. The key insight in [9] to build the theory of signal processing on graphs is to identify the shift operator. We adopted the weighted adjacency matrix of the graph as the shift operator and then developed appropriate concepts of z-transform, impulse and frequency response, filtering, convolution, and Fourier transform. In particular, the graph Fourier transform in this framework expands a graph signal into a basis of eigenvectors of the adjacency matrix, and the corresponding spectrum is given by the eigenvalues of the adjacency matrix. This contrasts with the Laplacian-based approach, where Fourier transform and spectrum are defined by the eigenvectors and eigenvalues of the graph Laplacian.

The association of the graph shift with the adjacency matrix is natural and has multiple intuitive interpretations. The graph shift is an elementary filter, and its output is a graph signal

with the value at vertex n given approximately by a weighted linear combination of the input signal values at neighbors of n [9]. With appropriate edge weights, the graph shift can be interpreted as a (minimum mean square) first-order linear predictor [23], [9]. Another interpretation of the graph filter comes from Markov chain theory [32], where the adjacency matrix represents the one-step transition probability matrix of the chain governing its dynamics. Finally, the graph shift can also be seen as a stencil approximation of the first-order derivative on the graph[1].

The last interpretation of the graph shift contrasts with the corresponding interpretation of the graph Laplacian: the adjacency matrix is associated with a first-order differential operator, while the Laplacian, if viewed as a shift, is associated with a second-order differential operator. In the one-dimensional case, the eigenfunctions for both, the first order and second order differential operators, are complex exponentials, since

$$\frac{1}{2\pi j}\frac{d}{dt}e^{2\pi jft} = fe^{2\pi jft}. \tag{1}$$

Interpreting the Laplacian as a shift introduces an even symmetry assumption into the corresponding signal model, and for one-dimensional signals [25], this model assumes that the signals are defined on lines of an image (undirected line graphs) rather than on a time line (directed line graphs). The use of the adjacency matrix as the graph shift does not impose such assumptions, and the corresponding framework can be used for arbitrary signals indexed by general graphs, regardless whether these graphs have undirected or directed edges with real or complex, non-negative or negative weights.

This paper is concerned with defining low and high frequencies and low-, high-, and band-pass graph signals and filters on generic graphs. In traditional discrete signal processing (DSP), these concepts have an intuitive interpretation, since the frequency contents of time series and digital images are described by complex or real sinusoids that oscillate at different rates [33]. The oscillation rates provide a physical notion of "low" and "high" frequencies: low-frequency components oscillate less and high-frequency ones oscillate more. However, these concepts do not have a similar interpretation on graphs, and it is not obvious how to order graph frequencies to describe the low- and high-frequency contents of a graph signal.

We present an ordering of the graph frequencies that is based on how "oscillatory" the spectral components are with respect to the indexing graph, i.e., how much they change from a node to neighboring nodes. To quantify this amount, we introduce the *graph total variation* function that measures how much signal samples (values of a graph signal at a node) vary in comparison to neighboring samples. This approach is analogous to the classical DSP theory, where the oscillations in time and image signals are also quantified by appropriately defined total variations [33]. In Laplacian-based graph signal processing [8], the authors choose to order frequencies based on a quadratic form rather than on the total variation of the

graph signal. Once we have an ordering of the frequencies based on the graph total variation function, we define the notions of low and high frequencies, as well as low-, high-, and band-pass graph signals and graph filters. We demonstrate that these concepts can be used effectively in sensor network analysis and semi-supervised learning. In our experiments, we show that naturally occurring graph signals, such as measurements of physical quantities collected by sensor networks or labels of objects in a dataset, tend to be low-frequency graph signals, while anomalies in sensor measurements or missing data labels can amplify high-frequency parts of the signals. We demonstrate how these anomalies can be detected using appropriately designed high-pass graph filters, and how unknown parts of graph signals can be recovered with appropriately designed regularization techniques. In particular, our experiments show that classifiers designed using the graph shift matrix lead to higher classification accuracy than classifiers based on the graph Laplacian matrices, combinatorial or normalized.

Summary of the paper. In Section II, we present the notation and review from [9] the basics of discrete signal processing on graphs (DSP$_G$). In Section III, we define the local and total variation for graph signals. In Section IV, we use the proposed total variation to impose an ordering on frequency components from lowest to highest. In Section V, we discuss low-, high-, and band-pass graph filters and their design. In Section VI, we illustrate these concepts with applications to corrupted measurement detection in sensor networks and data classification, and provide experimental results for real-world datasets. Finally, Section VII concludes the paper.

II. DISCRETE SIGNAL PROCESSING ON GRAPHS

In this section, we briefly review notation and concepts of the DSP$_G$ framework that are relevant to this paper. A complete introduction to the theory can be found in [9], [10], [11].

A. Graph Signals

Signal processing on graphs is concerned with the analysis and processing of datasets in which data elements can be connected to each other according to some relational property. This relation is expressed though a graph $G = (\mathcal{V}, \mathbf{A})$, where $\mathcal{V} = \{v_0, \ldots, v_{N-1}\}$ is a set of nodes and $\mathbf{A}$ is a weighted adjacency matrix of the graph. Each data element corresponds to node v_n (we also say the data element is *indexed* by v_n), and each weight $\mathbf{A}_{n,m} \in \mathbb{C}$ of a directed edge from v_m to v_n reflects the degree of relation of the mth data element to the nth one. Node v_m is an in-neighbor of v_n and v_n is an out-neighbor of v_m if $\mathbf{A}_{n,m} \neq 0$. All in-neighbors of v_n form its *in-neighborhood*, and we denote a set of their indices as $\mathcal{N}_n = \{m \mid \mathbf{A}_{n,m} \neq 0\}$. If the graph is undirected, the relation goes both ways, $\mathbf{A}_{n,m} = \mathbf{A}_{m,n}$, and the nodes are neighbors.

Using this graph, we refer to the dataset as a *graph signal*, which is defined as a map

$$\begin{aligned} \mathbf{s} \quad &: \quad \mathcal{V} \to \mathbb{C}, \\ &v_n \mapsto s_n. \end{aligned} \tag{2}$$

We assume that each dataset element s_n is a complex number. Since each signal is isomorphic to a complex-valued vector

---

[1]This analogy is more intuitive to understand if the graph is regular.

with N elements, we write graph signals as vectors

$$\mathbf{s} = \begin{bmatrix} s_0 & s_1 & \ldots & s_{N-1} \end{bmatrix}^T \in \mathbb{C}^N.$$

However, we emphasize that each element s_n is indexed by node v_n of a given representation graph $G = (\mathcal{V}, \mathbf{A})$, as defined by (2). The space $\mathcal{S}$ of graph signals (2) is isomorphic to $\mathbb{C}^N$, and its dimension is $\dim \mathcal{S} = N$.

B. Graph Filters

In general, a *graph filter* is a system $\mathbf{H}(\cdot)$ that takes a graph signal $\mathbf{s}$ as an input, processes it, and produces another graph signal $\tilde{\mathbf{s}} = \mathbf{H}(\mathbf{s})$ as an output. A basic non-trivial filter defined on a graph $G = (\mathcal{V}, \mathbf{A})$, called the *graph shift*, is a local operation that replaces a signal value s_n at node v_n with the linear combination of values at the neighbors of node v_n

$$\tilde{s}_n = \sum_{m \in \mathcal{N}_n} \mathbf{A}_{n,m} s_m. \tag{3}$$

Hence, the output of the graph shift is given by the product of the input signal with the adjacency matrix of the graph:

$$\tilde{\mathbf{s}} = \begin{bmatrix} \tilde{s}_0 & \ldots & \tilde{s}_{N-1} \end{bmatrix}^T = \mathbf{A}\mathbf{s}. \tag{4}$$

The graph shift is the basic building block in DSP_G.

All linear, shift-invariant[2] graph filters in DSP_G are polynomials in the adjacency matrix $\mathbf{A}$ of the form [9]

$$h(\mathbf{A}) = h_0 \mathbf{I} + h_1 \mathbf{A} + \ldots + h_L \mathbf{A}^L. \tag{5}$$

The output of the filter (5) is the signal

$$\tilde{\mathbf{s}} = \mathbf{H}(\mathbf{s}) = h(\mathbf{A})\mathbf{s}.$$

Linear, shift-invariant graph filters possess a number of useful properties. They have at most $L \leq N_\mathbf{A}$ taps h_ℓ, where $N_\mathbf{A} = \deg m_\mathbf{A}(x)$ is the degree of the minimal polynomial[3] $m_\mathbf{A}(x)$ of $\mathbf{A}$. If a graph filter (5) is invertible, i.e., matrix $h(\mathbf{A})$ is non-singular, then its inverse is also a graph filter $g(\mathbf{A}) = h(\mathbf{A})^{-1}$ on the same graph $G = (\mathcal{V}, \mathbf{A})$. Finally, the space of graph filters is an *algebra*, i.e., a vector space that is simultaneously a ring.

These properties guarantee that multiplying $\mathbf{A}$ by any nonzero constant does not change the set of corresponding linear, shift-invariant graph filters. In particular, we can define the normalized graph shift matrix

$$\mathbf{A}^{\mathrm{norm}} = \frac{1}{|\lambda_{\max}|}\mathbf{A}, \tag{6}$$

where $\lambda_{\max}$ denotes the eigenvalue of $\mathbf{A}$ with the largest magnitude, i.e.,

$$|\lambda_{\max}| \geq |\lambda_m| \tag{7}$$

for all $0 \leq m \leq M-1$. The normalized matrix (6) ensures the numerical stability of computing with graph filters $h(\mathbf{A}^{\mathrm{norm}})$ as it prevents excessive scaling of the shifted signal, since $\|\mathbf{A}^{\mathrm{norm}}\mathbf{s}\| / \|\mathbf{s}\| \leq 1$. In this paper, we use the graph shift $\mathbf{A}^{\mathrm{norm}}$ instead of $\mathbf{A}$ where appropriate.

---

[2] Filters are *linear* if for a linear combination of inputs they produce the same linear combination of outputs. Filters are *shift-invariant* if the result of consecutive processing of a signal by multiple graph filters does not depend on the order of processing; i.e., shift-invariant filters commute with each other.

[3] The minimal polynomial of $\mathbf{A}$ is the unique monic polynomial of the smallest degree that annihilates $\mathbf{A}$, i.e., $m_\mathbf{A}(\mathbf{A}) = 0$ [34], [35].

C. Graph Fourier Transform

In general, a Fourier transform performs the expansion of a signal into a *Fourier basis* of signals that are invariant to filtering. In the DSP_G framework, a graph Fourier basis corresponds to the Jordan basis of the graph adjacency matrix $\mathbf{A}$ (the Jordan decomposition is reviewed in Appendix A). Following the DSP notation, distinct eigenvalues $\lambda_0, \lambda_1, \ldots, \lambda_{M-1}$ of the adjacency matrix $\mathbf{A}$ are called the *graph frequencies* and form the *spectrum* of the graph, and the Jordan eigenvectors that correspond to a frequency λ_m are called the *frequency components* corresponding to the mth frequency. Since multiple eigenvectors can correspond to the same eigenvalue, in general, a graph frequency can have multiple graph frequency components associated with it.

As reviewed in Appendix A, Jordan eigenvectors form the columns of the matrix $\mathbf{V}$ in the Jordan decomposition (47)

$$\mathbf{A} = \mathbf{V}\mathbf{J}\mathbf{V}^{-1}.$$

Hence, the *graph Fourier transform* of a graph signal $\mathbf{s}$ is

$$\widehat{\mathbf{s}} = \mathbf{F}\mathbf{s}, \tag{8}$$

where $\mathbf{F} = \mathbf{V}^{-1}$ is the graph Fourier transform matrix. The values $\widehat{s}_n$ of the signal's graph Fourier transform (8) characterize the *frequency content* of the signal $\mathbf{s}$.

The *inverse graph Fourier transform* is given by

$$\mathbf{s} = \mathbf{F}^{-1}\widehat{\mathbf{s}} = \mathbf{V}\widehat{\mathbf{s}}. \tag{9}$$

It reconstructs the original signal from its frequency contents by constructing a linear combination of frequency components weighted by the signal's Fourier transform coefficients.

D. Frequency Response

The graph Fourier transform (8) also allows us to characterize the effect of a filter on the frequency content of an input signal. As follows from (5) and (8), as well as (47) in Appendix A,

$$\tilde{\mathbf{s}} = h(\mathbf{A})\mathbf{s} = \mathbf{F}^{-1} h(\mathbf{J})\mathbf{F}\mathbf{s} \quad \Leftrightarrow \quad \mathbf{F}\tilde{\mathbf{s}} = h(\mathbf{J})\widehat{\mathbf{s}}. \tag{10}$$

Hence, the frequency content of the output signal is obtained by multiplying the frequency content of the input signal by the block diagonal matrix

$$h(\mathbf{J}) = \begin{bmatrix} h(\mathbf{J}_{r_{0,0}}(\lambda_0)) & & \\ & \ddots & \\ & & h(\mathbf{J}_{r_{M-1,D_{M-1}}}(\lambda_{M-1})) \end{bmatrix}.$$

We call this matrix the *graph frequency response* of the filter $h(\mathbf{A})$, and denote it as

$$\widehat{h(\mathbf{A})} = h(\mathbf{J}). \tag{11}$$

Notice that (10) extends the *convolution theorem* from classical signal processing [7] to graphs, since filtering a signal on a graph is equivalent in the frequency domain to multiplying the signal's spectrum by the frequency response of the filter.

Fig. 1. Traditional graph representation for a finite discrete periodic time series of length N.

E. Consistency with the Classical DSP

The DSP_G framework is consistent with the classical DSP theory. Finite (or periodic) time series can be represented by the directed cycle graph shown in Fig. 1, see [24], [9]. The direction of the edges represents the flow of time from past to future, and the edge from the last vertex v_{N-1} to v_0 captures the periodic signal extension $s_N = s_0$ for time series. The adjacency matrix of the graph in Fig. 1 is the $N \times N$ cyclic permutation matrix

$$\mathbf{A} = \mathbf{C} = \begin{bmatrix} & & & 1 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix}. \tag{12}$$

Substituting (12) into the graph shift (4) yields the standard time delay

$$\widetilde{s}_n = s_{n-1 \mod N}. \tag{13}$$

The matrix (12) is diagonalizable. Its eigendecomposition, which coincides with the Jordan decomposition (47), is

$$\mathbf{C} = \frac{1}{N} \mathbf{DFT}_N^{-1} \begin{bmatrix} e^{-j\frac{2\pi \cdot 0}{N}} & & \\ & \ddots & \\ & & e^{-j\frac{2\pi \cdot (N-1)}{N}} \end{bmatrix} \mathbf{DFT}_N,$$

where $\mathbf{DFT}_N$ is the discrete Fourier transform matrix. Thus, as expected, the graph Fourier transform for signals indexed by the graph in Fig. 1 is $\mathbf{F} = \mathbf{DFT}_N$, and the corresponding frequencies are[4], for $0 \le n < N$,

$$e^{-j\frac{2\pi}{N}n}. \tag{14}$$

III. Total Variation on Graphs

In this section, we define he total variation on graph signals that is based on the concept of graph shift.

In classical DSP, the *total variation* (TV) of a discrete signal is defined as the sum of magnitudes of differences between two consecutive signal samples [33]:

$$\text{TV}(\mathbf{s}) = \sum_n \left| s_n - s_{n-1} \right|. \tag{15}$$

For a finite time series, the periodicity condition $s_n = s_{n \mod N}$ yields a modified definition

$$\text{TV}(\mathbf{s}) = \sum_{n=0}^{N-1} \left| s_n - s_{n-1 \mod N} \right|. \tag{16}$$

The total variation (15) and (16) for time series or space signals, such as images, has an intuitive interpretation: it

[4]In DSP, the ratio $\frac{2\pi}{N}n$ in the exponent (14) sometimes is also called frequency. In this case, the frequencies are N real numbers between 0 and 2π. However, to remain consistent with the discussion in this paper, we refer to the exponentials (14) as frequencies, and view them as complex numbers of magnitude 1 residing on the unit circle in the complex plane.

compares how the signal varies with time or space. These concepts lie at the heart of many applications of DSP, including signal regularization and denoising [33], [36], image compression [37] and others.

The variation (16) compares two consecutive signal samples and calculates a cumulative magnitude of the signal change over time. In terms of the time shift (13), we can say that the total variation compares a signal $\mathbf{s}$ to its shifted version: the smaller the difference between the original signal and the shifted one, the lower the signal's variation. Using the cyclic permutation matrix (12), we can write (16) as

$$\text{TV}(\mathbf{s}) = ||\mathbf{s} - \mathbf{C}\,\mathbf{s}||_1. \tag{17}$$

The total variation (17) measures the difference between the signal samples at each vertex and at its neighbor on the graph that represents finite time series in Fig. 1.

The DSP_G generalizes the DSP theory from lines and regular lattices to arbitrary graphs. Hence, we extend (17) to an arbitrary graph $G = (\mathcal{V}, \mathbf{A})$ by defining the total variation on a graph as a measure of similarity between a graph signal and its shifted version (4):

Definition 1 (Total Variation on Graphs): The *total variation on a graph* (TV_G) of a graph signal $\mathbf{s}$ is defined as

$$\text{TV}_\text{G}(\mathbf{s}) = ||\mathbf{s} - \mathbf{A}^{\text{norm}}\mathbf{s}||_1. \tag{18}$$

The definition uses the normalized adjacency matrix $\mathbf{A}^{\text{norm}}$ to guarantee that the shifted signal is properly scaled for comparison with the original signal, as discussed in Section II-B.

The intuition behind Definition 1 is supported by the underlying mathematical model. Similarly to the calculus on discrete signals that defines the discretized derivative as $\nabla_n(\mathbf{s}) = s_n - s_{n-1}$ [33], in DSP_G the derivative (and the gradient) of a graph signal at the nth vertex is defined by the graph shift $\mathbf{A}^{\text{norm}}$ as

$$\frac{d\mathbf{s}}{dv_n} = \nabla_n(\mathbf{s}) = s_n - \sum_{m \in \mathcal{N}_n} \mathbf{A}^{\text{norm}}_{n,m} s_m. \tag{19}$$

The *local variation* of the signal at vertex v_n is the magnitude $|\nabla_n(\mathbf{s})|$ of the corresponding gradient, and the *total variation* is the sum of local variations for all vertices [33], [8]. In particular, if we define the discrete p-Dirichlet form

$$\text{S}_p(\mathbf{s}) = \frac{1}{p} \sum_{n=0}^{N-1} |\nabla_n(\mathbf{s})|^p, \tag{20}$$

then for $p = 1$ the form

$$
\begin{aligned}
\text{S}_1(\mathbf{s}) &= \sum_{n=0}^{N-1} |\nabla_n(\mathbf{s})| \\
&= \sum_{n=0}^{N-1} \left| s_n - \sum_{m \in \mathcal{N}_n} \mathbf{A}^{\text{norm}}_{n,m} s_m \right| \\
&= ||\mathbf{s} - \mathbf{A}^{\text{norm}}\mathbf{s}||_1
\end{aligned} \tag{21}
$$

defines the total variation of the graph signal $\mathbf{s}$. It coincides with Definition 1.

The total variation defined through the 1-Dirichlet form (21) depends on the definition of the signal gradient at a graph

vertex. For finite time DSP, the gradient is defined by the discretized derivative $\nabla_n(\mathbf{s}) = s_n - s_{n-1}$ [33] and yields the total variation (17). The DSP_G extends the notion of the shift to (3), which leads to the gradient (19) and the total variation (18).

Remark. In [8], the frequencies are ordered using a 2-Dirichlet form, i.e., a quadratic function.

IV. LOW AND HIGH FREQUENCIES ON GRAPHS

In this section, we use the total variation (18) to introduce an ordering on frequencies that leads to the definition of low and high frequencies on graphs. We demonstrate that this ordering is unique for graphs with real spectra and not unique for graphs with complex spectra.

A. Variation of the Graph Fourier Basis

As discussed in Section II, the graph Fourier basis for an arbitrary graph is given by the Jordan basis of the adjacency matrix $\mathbf{A}$. Consider an eigenvalue λ of $\mathbf{A}$, and let $\mathbf{v} = \mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{R-1}$ be a Jordan chain of generalized eigenvectors that corresponds to this eigenvalue. Let the indicator function

$$\mathbf{i}_r = \begin{cases} 0, & r = 0 \\ 1, & 1 \leq r < R \end{cases}$$

specify whether $\mathbf{v}_r$ is a proper eigenvector of $\mathbf{A}$ or a generalized one. Then we can write the condition (43) on generalized eigenvectors (see Appendix A) as

$$\mathbf{A}\mathbf{v}_r = \lambda \mathbf{v}_r + \mathbf{i}_r \mathbf{v}_{r-1}. \tag{22}$$

Using (22), we write the total variation (18) of the generalized eigenvector $\mathbf{v}_r$ as

$$\begin{aligned} \text{TV}_\text{G}(\mathbf{v}_r) &= \left\| \mathbf{v}_r - \mathbf{A}^{\text{norm}}\mathbf{v}_r \right\|_1 \tag{23} \\ &= \left\| \mathbf{v}_r - \frac{1}{|\lambda_{\max}|}\mathbf{A}\mathbf{v}_r \right\|_1 \\ &= \left\| \mathbf{v}_r - \frac{\lambda}{|\lambda_{\max}|}\mathbf{v}_r - \frac{\mathbf{i}_r}{|\lambda_{\max}|}\mathbf{v}_{r-1} \right\|_1. \end{aligned}$$

In particular, when $\mathbf{v}_r$ is a proper eigenvector of $\mathbf{A}$, i.e. $r = 0$ and $\mathbf{v}_0 = \mathbf{v}$, we have $\mathbf{i}_0 = 0$. In this case, it follows from (23) that the total variation of the eigenvector $\mathbf{v}$ is

$$\text{TV}_\text{G}(\mathbf{v}) = \left| 1 - \frac{\lambda}{|\lambda_{\max}|} \right| \|\mathbf{v}\|_1. \tag{24}$$

When a frequency component is a proper eigenvector of the adjacency matrix $\mathbf{A}$, its total variation (24) is determined by the corresponding eigenvalue, since we can scale all eigenvectors to have the same ℓ_1-norm. Moreover, all proper eigenvectors corresponding to the same eigenvalue have the same total variation. However, when a frequency component is not a proper eigenvector, we must use (23) to compare its variation with other frequency components. Finally, it follows from (7) for $\|\mathbf{v}\|_1 = 1$ that

$$\text{TV}_\text{G}(\mathbf{v}) = \left| 1 - \frac{\lambda}{|\lambda_{\max}|} \right| \leq 1 + \left| \frac{\lambda}{|\lambda_{\max}|} \right| \leq 2. \tag{25}$$

Hence, the total variation of a normalized proper eigenvector is a real number between 0 and 2.

B. Frequency Ordering

The total variation of the Fourier basis, given by (23) and (24), allows us to order the graph frequency components in the order of increasing variation. Following DSP convention, we call frequency components with smaller variations *low* frequencies and components with higher variations *high* frequencies.

Here, we determine the frequency ordering induced by the total variation (24) for graphs that have diagonalizable adjacency matrices, i.e., only have proper eigenvectors. This ordering can be similarly extended to graphs with non-diagonalizable adjacency matrices using the variation (23) of generalized eigenvectors.[5]

The following theorem establishes the relative ordering of two distinct real frequencies.

Theorem 1: Consider two distinct real eigenvalues $\lambda_m, \lambda_n \in \mathbb{R}$ of the adjacency matrix $\mathbf{A}$ with corresponding eigenvectors $\mathbf{v}_m$ and $\mathbf{v}_n$. If the eigenvalues are ordered as

$$\lambda_m < \lambda_n, \tag{26}$$

then the total variations of their eigenvectors satisfy

$$\text{TV}_\text{G}(\mathbf{v}_m) > \text{TV}_\text{G}(\mathbf{v}_n). \tag{27}$$

Proof: Since the eigenvalues are real, it follows from (26) that the difference between the total variations of the two eigenvectors satisfies

$$\begin{aligned} \text{TV}_\text{G}(\mathbf{v}_m) - \text{TV}_\text{G}(\mathbf{v}_n) &= \left| 1 - \frac{\lambda_m}{|\lambda_{\max}|} \right| - \left| 1 - \frac{\lambda_n}{|\lambda_{\max}|} \right| \\ &\overset{(a)}{=} \left(1 - \frac{\lambda_m}{|\lambda_{\max}|} \right) - \left(1 - \frac{\lambda_n}{|\lambda_{\max}|} \right) \\ &= \frac{\lambda_n - \lambda_m}{|\lambda_{\max}|} > 0, \end{aligned}$$

which yields (27). Here, equality (a) follows from (7). ∎

As follows from Theorem 1, if a graph has a real spectrum and its frequencies are ordered as

$$\lambda_0 > \lambda_1 > \ldots > \lambda_{M-1}, \tag{28}$$

then λ_0 represents the lowest frequency and λ_{M-1} is the highest frequency. Moreover, the ordering (28) is a unique ordering of all frequencies from lowest to highest. This frequency ordering for matrices with real spectra is visualized in Fig. 2(a).

The next theorem extends Theorem 1 and establishes the relative ordering of two distinct frequencies corresponding to complex eigenvalues.

Theorem 2: Consider two distinct complex eigenvalues $\lambda_m, \lambda_n \in \mathbb{C}$ of the adjacency matrix $\mathbf{A}$. Let $\mathbf{v}_m$ and $\mathbf{v}_n$ be the corresponding eigenvectors. The total variations of these eigenvectors satisfy

$$\text{TV}_\text{G}(\mathbf{v}_m) < \text{TV}_\text{G}(\mathbf{v}_n) \tag{29}$$

---

[5]Some real-world datasets are described by graphs with non-diagonalizable adjacency matrices and thus require a proper ordering of frequency components that correspond to generalized eigenvectors. An example of such a dataset is the directed graph of hyperlink references between political blogs used in Section VI-B.
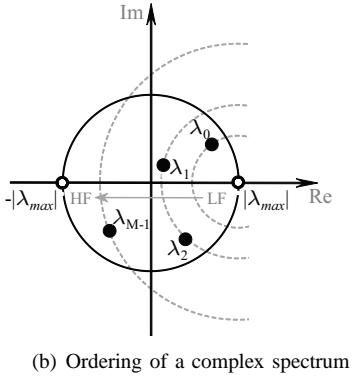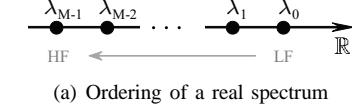
(a) Ordering of a real spectrum

(b) Ordering of a complex spectrum

Fig. 2. Frequency ordering from low frequencies (LF) to high frequencies (HF) for graphs with real and complex spectra.

Fig. 3. Frequency ordering for finite discrete periodic time series. Frequencies λ_m and λ_{N-m} have the same total variations since they lie on the same circle centered around 1.

if the eigenvalue λ_m is located closer to the value $|\lambda_{\max}|$ on the complex plane than the eigenvalue λ_n.

Proof: This result follows immediately from the interpretation of the total variation (24) as a distance function on the complex plane. Since $\lambda_{\max} \neq 0$ (otherwise $\mathbf{A}$ would have been a zero matrix), multiplying both sides of (29) by $|\lambda_{\max}|$ yields the equivalent inequality

$$\left| |\lambda_{\max}| - \lambda_m \right| < \left| |\lambda_{\max}| - \lambda_n \right|. \tag{30}$$

The expressions on both sides of (30) are the distances from λ_m and λ_n to $|\lambda_{\max}|$ on the complex plane. ∎

As follows from Theorem 2, frequencies of a graph with a complex spectrum are ordered by their distance from $|\lambda_{\max}|$. As a result, in contrast to the graphs with real spectra, the induced ordering of complex frequencies from lowest to highest is not unique, since distinct complex frequencies can yield the same total variation for their corresponding frequency components. In particular, all eigenvalues lying on a circle of radius ρ centered at point $|\lambda_{\max}|$ on the complex plane have the same total variation $\rho/|\lambda_{\max}|$. It also follows from (7) that all graph frequencies λ_m can lie only inside and on the boundary of the circle with radius $|\lambda_{\max}|$. The frequency ordering for adjacency matrices with complex spectra is visualized in Fig. 2(b).

Consistency with DSP theory. The frequency ordering induced by the total variation (18) is consistent with classical DSP. Recall from (14) that the cycle graph in Fig. 1, which represents finite time series, has a complex spectrum

$$\lambda_n = e^{-j\frac{2\pi}{N}n}$$

for $0 \leq n < N$. Hence, the total variation of the nth frequency component is

$$
\begin{aligned}
\mathrm{TV_G}(\mathbf{v}_n) &= \left| 1 - e^{-j\frac{2\pi n}{N}} \right| \\
&= \left| 1 - \cos\frac{2\pi n}{N} \right| + \left| \sin\frac{2\pi n}{N} \right|.
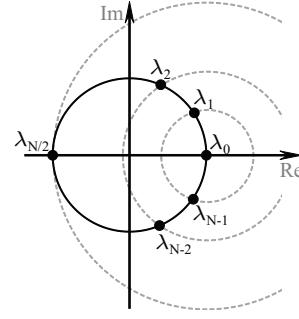\end{aligned}
$$

Hence, the frequencies λ_m and λ_{N-m} have the same variation, and the induced order from lowest to highest frequencies is $\lambda_0, \lambda_1, \lambda_{N-1}, \lambda_2, \lambda_{N-2}, \ldots$, with the lowest frequency corresponding to $\lambda_0 = 1$ and the highest frequency corresponding to $\lambda_{N/2} = -1$ for even N or $\lambda_{(N\pm1)/2}$ for odd N. This ordering is visualized in Fig. 3, and it is the conventional frequency ordering in DSP [7].

C. Frequency Ordering Based on Quadratic Form

Here we compare our ordering of the frequencies based on the total variation with an ordering based on using the 2-Dirichlet form, $p = 2$, like in [8]. Taking $p = 2$ in (20), we get

$$
\begin{aligned}
\mathrm{S}_2(\mathbf{s}) &= \frac{1}{2} \sum_{n=0}^{N-1} |\nabla_n(\mathbf{s})|^2 \\
&= \frac{1}{2} ||\mathbf{s} - \mathbf{A}^{\mathrm{norm}}\mathbf{s}||_2^2 \tag{31} \\
&= \frac{1}{2}\mathbf{s}^H \left(\mathbf{I} - \mathbf{A}^{\mathrm{norm}}\right)^H \left(\mathbf{I} - \mathbf{A}^{\mathrm{norm}}\right) \mathbf{s}.
\end{aligned}
$$

This quadratic form defines the seminorm

$$||\mathbf{s}||_G = \sqrt{\mathrm{S}_2(\mathbf{s})}, \tag{32}$$

since $\left(\mathbf{I} - \mathbf{A}^{\mathrm{norm}}\right)^H \left(\mathbf{I} - \mathbf{A}^{\mathrm{norm}}\right)$ is a positive-semidefinite matrix. The rationale in [8] is that the quadratic form is small when signal values are close to the corresponding linear combinations of their neighbors' values, and large otherwise.

We introduce an ordering of the graph Fourier basis from lowest to highest frequencies based on the graph shift quadratic form. As we demonstrate next, this ordering coincides with the ordering induced by the total variation.

The quadratic form (31) of an eigenvector $\mathbf{v}$ that corresponds to the eigenvalue λ is

$$
\begin{aligned}
\mathrm{S}_2(\mathbf{v}) &= \frac{1}{2} ||\mathbf{v} - \mathbf{A}^{\mathrm{norm}}\mathbf{v}||_2^2 \\
&= \left| 1 - \frac{\lambda}{|\lambda_{\max}|} \right|^2 ||\mathbf{v}||_2. \tag{33}
\end{aligned}
$$

Consider two real eigenvalues λ_m and λ_n with corresponding eigenvectors $\mathbf{v}_m$ and $\mathbf{v}_n$. If these eigenvalues satisfy

$\lambda_m < \lambda_n$, then it follows from (33) that

$$
\begin{aligned}
S_2(\mathbf{v}_m) - S_2(\mathbf{v}_n) &= \left|1 - \frac{\lambda_m}{|\lambda_{\max}|}\right|^2 - \left|1 - \frac{\lambda_n}{|\lambda_{\max}|}\right|^2 \\
&= \left(1 - \frac{\lambda_m}{|\lambda_{\max}|}\right)^2 - \left(1 - \frac{\lambda_n}{|\lambda_{\max}|}\right)^2 \\
&= \frac{\lambda_n - \lambda_m}{|\lambda_{\max}|^2} \left(\lambda_m + \lambda_n - 2|\lambda_{\max}|\right) > 0.
\end{aligned}
$$

Hence, $S_2(\mathbf{v}_m) > S_2(\mathbf{v}_n)$, and we obtain a reformulation of Theorem 1 for the graph shift quadratic form. As a consequence, arranging frequency components in the increasing order of their graph shift quadratic form leads to the same ordering (27) from lowest to highest as the total variation.

A similar reformulation of Theorem 2 for the graph shift quadratic form is demonstrated analogously, which leads to the same ordering of complex frequencies as the ordering induced by the total variation.

V. FILTER DESIGN

When a graph signal is processed by a graph filter, its frequency content changes according to the frequency response (11) of the filter. Similarly to classical DSP, we can characterize graph filters as low-, high-, and band-pass filters based on their frequency response.

A. Low-, High-, and Band-pass Graph Filters

Following the DSP convention, we call filters *low-pass* if they do not significantly affect the frequency content of low-frequency signals but attenuate, i.e., reduce, the magnitude of high-frequency signals. Analogously, *high-pass* filters pass high-frequency signals while attenuating low-frequency ones, and *band-pass* filters pass signals with frequency content within a specified frequency band while attenuating all others.

The action of a graph filter $h(\mathbf{A})$ of the form (5) on the frequency content of a graph signal $\mathbf{s}$ is completely specified by its frequency response (11). For simplicity of presentation, we discuss here graphs with diagonalizable adjacency matrices, for which (11) is a diagonal matrix with $h(\lambda_m)$ on the main diagonal, $0 \leq m < M$. In this case, the Fourier transform coefficients of the filtered signal

$$\widetilde{\mathbf{s}} = h(\mathbf{A})\mathbf{s}$$

are the Fourier transform coefficients of the input signal multiplied element-wise by the frequency response of the filter:

$$
\mathbf{F}\widetilde{\mathbf{s}} = \begin{bmatrix} h(\lambda_0) & & \\ & \ddots & \\ & & h(\lambda_{M-1}) \end{bmatrix} \widehat{\mathbf{s}} = \begin{bmatrix} h(\lambda_0)\widehat{s}_0 \\ \vdots \\ h(\lambda_{M-1})\widehat{s}_{N-1} \end{bmatrix}.
$$

Hence, to attenuate the frequency content of a signal inside a specific part of the spectrum, we should design a filter $h(\mathbf{A})$ that for corresponding frequencies λ_m satisfies $h(\lambda_m) \approx 0$.

Consider an example of ideal low-pass and high-pass filters $h(\mathbf{A})$ and $g(\mathbf{A})$. Let the cut-off frequency λ_{cut} equal to the median of the bandwidth, i.e., be such that exactly half of

frequencies λ_m are lower frequencies than λ_{cut}. The frequency responses of these filters are defined as

$$
h(\lambda_m) = 1 - g(\lambda_m) = \begin{cases} 1, & \lambda_m > \lambda_{\text{cut}}, \\ 0, & \lambda_m \leq \lambda_{\text{cut}}. \end{cases} \tag{34}
$$

As we demonstrate next, the design of such filters, as well as any low-, high-, and band-pass graph filters, is a linear problem.

B. Frequency Response Design for Graph Filters

A graph filter can be defined through its frequency response $h(\lambda_m)$ at its distinct frequencies λ_m, $m = 0, \ldots, M-1$. Since a graph filter (5) is a polynomial of degree L, the construction of a filter with frequency response $h(\lambda_m) = \alpha_m$ corresponds to inverse polynomial interpolation, i.e., solving a system of M linear equations with $L+1$ unknowns $h_0, \ldots, h_L$:

$$
\begin{aligned}
h_0 + h_1\lambda_0 + \ldots + h_L\lambda_0^L &= \alpha_0, \\
h_0 + h_1\lambda_1 + \ldots + h_L\lambda_1^L &= \alpha_1, \\
&\vdots \\
h_0 + h_1\lambda_{M-1} + \ldots + h_L\lambda_{M-1}^L &= \alpha_{M-1}.
\end{aligned} \tag{35}
$$

This system can be written as

$$
\begin{bmatrix} 1 & \lambda_0 & \ldots & \lambda_0^L \\ 1 & \lambda_1 & \ldots & \lambda_1^L \\ \vdots & \vdots & & \vdots \\ 1 & \lambda_{M-1} & \ldots & \lambda_{M-1}^L \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_L \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{M-1} \end{bmatrix}. \tag{36}
$$

The system matrix in (36) is a full-rank $M \times (L+1)$ Vandermonde matrix [34], [35]. Hence, the system has infinitely many exact solutions if $M \leq L$ and one unique exact solution if $M = L + 1$.

When $M \geq L + 1$, the system is overdetermined and does not have an exact solution. This is a frequent case in practice, since the number of coefficients in the graph filter may be restricted by computational efficiency or numerical stability requirements. In this case, we can find an approximate solution, for example, in the least-squares sense.

As an example of filter construction, consider a network of 150 weather stations that measure daily temperature near major cities across the United States [38]. We represent these stations with a directed 6-nearest neighbor graph, in which every sensor corresponds to a vertex and is connected to six closest sensors by directed edges. The edge weight between connected vertices v_n and v_m is

$$
\mathbf{A}_{n,m} = \frac{e^{-d_{nm}^2}}{\sqrt{\sum_{k \in \mathcal{N}_n} e^{-d_{nk}^2} \sum_{\ell \in \mathcal{N}_m} e^{-d_{m\ell}^2}}}, \tag{37}
$$

where $d_{n,m}$ denotes the geodesical distance between the nth and mth sensors. A daily snapshot of all 150 measurements forms a signal indexed by this graph, such as the example signal shown in Fig. 4.

Fig. 5 shows the frequency responses of the low- and high-pass filters for this graph that have degree $L = 10$. These filters are least-squares approximations of the ideal low- and
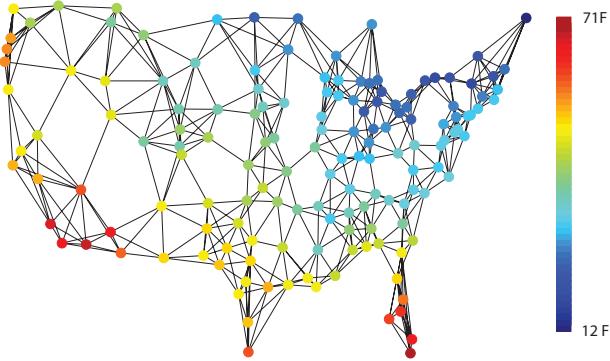
Fig. 4. Temperature measured by 150 weather stations across the United States on February 1, 2003
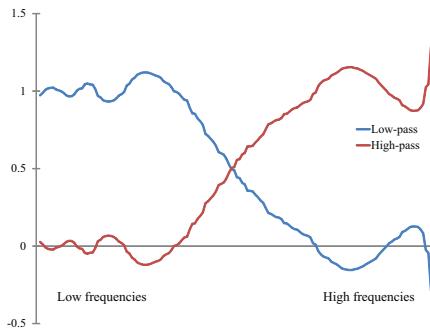

Fig. 5. Frequency responses of low-pass and high-pass filters for the sensor graph in Fig. 4. The length of the filters is restricted to 10 coefficients.

high-pass filters (34). The frequency response in (35) for the low-pass filter is $\alpha_m = 1$ for frequencies lower than λ_{cut} and 0 otherwise; and vice versa for the high-pass filter. By design, the constructed filters satisfy the relation [39]

$$h(\mathbf{A}) = \mathbf{I}_N - g(\mathbf{A}). \tag{38}$$

If we require that $h(\mathbf{A})$ and $g(\mathbf{A})$ do not have the same number of coefficients or if we use an approximation metric other than least squares, the constructed polynomials will not satisfy (38).

VI. Applications

In this section, we apply the theory discussed in this paper to graphs and datasets that arise in different contexts. We demonstrate how the DSP$_G$ framework extends standard signal processing techniques of band-pass filtering and signal regularization to solve interesting problems in sensor networking and data classification.

A. Malfunction Detection in Sensor Networks

Today, sensors are ubiquitous. They are usually cheap to manufacture and deploy, so networks of sensors are used to measure and monitor a wide range of physical quantities from structural integrity of buildings to air pollution. However, the sheer quantity of sensors and the area of their deployment may make it challenging to check that every sensor is operating correctly. As an alternative, it is desirable to detect
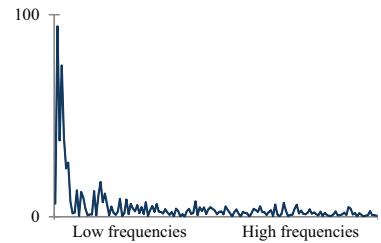

Fig. 6. The frequency content of the graph signal in Fig. 4. Frequencies are ordered from the lowest to highest.

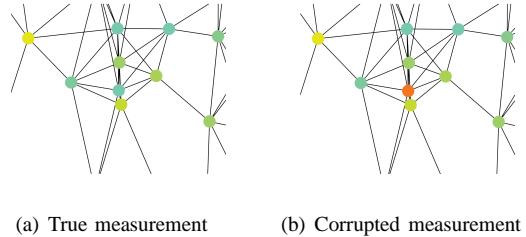(a) True measurement   (b) Corrupted measurement

Fig. 7. A subgraph of the sensor graph in Fig. 4 showing the (a) true and (b) corrupted measurement by the sensor located in Colorado Springs, CO.

a malfunctioning sensor solely from the data it generates. We illustrate here how the DSP$_G$ framework can be used to devise a simple solution to this problem.

Many physical quantities represent graph signals with small variation with respect to the graph of sensors. As an illustration, consider the temperature across the United States measured by 150 weather stations located near major cities [38]. An example temperature measurement is shown in Fig. 4, and the construction of the corresponding weather station graph is discussed in Section V-B. The graph Fourier transform of this temperature snapshot is shown in Fig. 6, with frequencies ordered from lowest to highest. Most of the signal's energy is concentrated in the low frequencies. This suggests that the signal varies slowly across the graph, i.e., that cities located close to each other have similar temperatures.

A sensor malfunction may cause an unusual difference between its measurements and the measurements of nearby stations. Fig. 7 shows an example of an (artificially) corrupted measurement, where the station located near Colorado Springs, CO, reports a temperature that contains an error of 20 degrees (temperature at each sensor is color-coded using the same color scheme as in Fig. 4). The true measurement in Fig. 7(a) is very similar to measurements at neighboring cities, while the corrupted measurement in Fig. 7(b) differs significantly from its neighbors.

Such difference in temperature at closely located cities results in the increased presence of higher frequencies in the corrupted signal. By high-pass filtering the signal and then thresholding the filtered output, we can detect this anomaly.

Experiment. We consider the problem of detecting a corrupted measurement from a single temperature station. We simulate a signal corruption by changing the measurement of one sensor by 20 degrees; such an error is reasonably small and is hard to detect by direct inspection of measurements of each station separately. To detect the malfunction, we extract
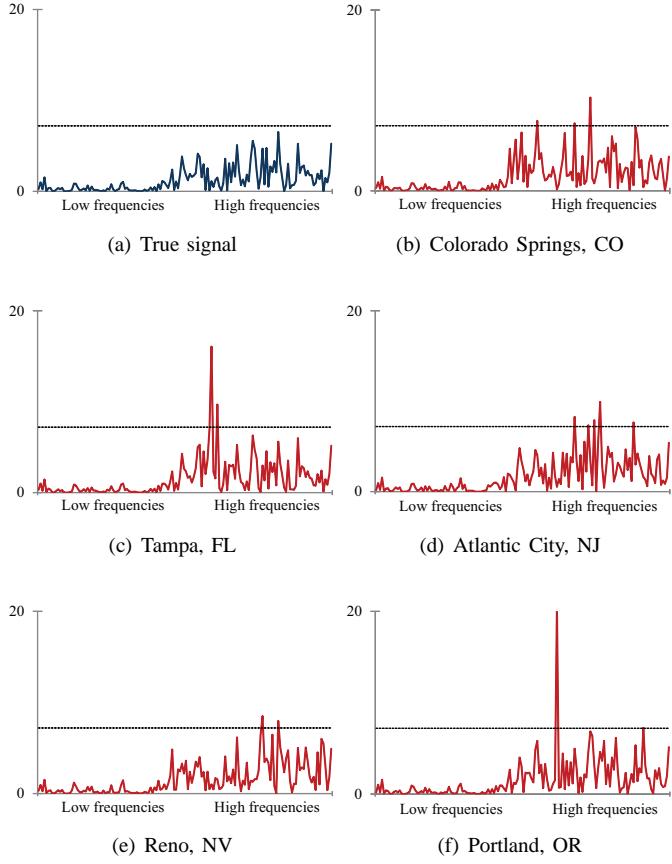
Fig. 8. The magnitudes of spectral coefficients of the original and corrupted temperature measurements after high-pass filtering: (a) the true signal from Fig. 4; (b)-(f) signals obtained from the true signal by corrupting the measurement of a single station located at the indicated city.

the high-frequency component of the resulting graph signal using the high-pass filter in Fig. 5 and then threshold it. If one or more Fourier transform coefficients exceed the threshold value, we conclude that a sensor is malfunctioning. The cut-off threshold is selected automatically as the maximum absolute value of graph Fourier transform coefficients of the high-pass filtered measurements from the previous three days.

Results. We considered 365 measurements collected during the year 2003 by all 150 stations, and conducted $150 \times 365 = 54,750$ tests. The resulting average detection accuracy was 89%, so the proposed approach, despite its relative simplicity, correctly detected a corrupted measurement almost 9 times out of 10.

Fig. 8 illustrates the conducted experiment. It shows frequency contents of high-pass filtered signals that contain a corrupted measurement from a sensor at five different locations. A comparison with the high-pass component of the uncorrupted signal in Fig. 8(a) shows coefficients above thresholds that lead to the detection of a corrupted signal.

B. Data Classification

Data classification is an important problem in machine learning and data mining [40], [41]. Its objective is to classify each element of the dataset based on specified characteristics

of the data. For example, image and video databases may be classified based on their contents; documents are grouped with respect to their topics; and customers may be distinguished based on their shopping preferences. In all cases, each dataset element is assigned a label from a pre-defined group of labels.

Large datasets often cannot be classified manually. In this case, a common approach is to classify only a subset of elements and then use a known structure of the dataset to predict the labels for the remaining elements. A key assumption in this approach is that similar elements tend to be in the same class. In this case, the labels tend to form a signal with small variation over the graph of similarities. Hence, information about similarity between dataset elements provides means for inferring unknown labels from known ones.

Consider a graph $G = (\mathcal{V}, \mathbf{A})$ with N vertices that represent N data elements. We assume that two elements are similar to each other if the corresponding vertices are connected; if their connection is directed, the similarity is assumed only in the direction of the edge. We define a signal $\mathbf{s}^{(\text{known})}$ on this graph that captures known labels. For a two-class problem, this signal is defined as

$$s_n^{(\text{known})} = \begin{cases} +1, & n\text{th element belongs to class 1}, \\ -1, & n\text{th element belongs to class 2}, \\ 0, & \text{class is unknown}. \end{cases}$$

The predicted labels for all data elements are found as the signal that varies the least on the graph $G = (\mathcal{V}, \mathbf{A})$. That is, we find the predicted labels as the solution to the optimization problem

$$\mathbf{s}^{(\text{predicted})} = \underset{\mathbf{s} \in \mathbb{R}^N}{\operatorname{argmin}} \ \mathrm{S}_2(\mathbf{s}) \qquad (39)$$

subject to

$$|| \mathbf{C}\,\mathbf{s}^{(\text{known})} - \mathbf{C}\,\mathbf{s} ||_2^2 < \epsilon, \qquad (40)$$

where $\mathbf{C}$ is a $N \times N$ diagonal matrix such that

$$\mathbf{C}_{n,n} = \begin{cases} 1, & \text{if } s_n^{(\text{known})} \neq 0, \\ 0, & \text{otherwise}. \end{cases}$$

The parameter ϵ in (40) controls how well the known labels are preserved. Alternatively, the problem (39) with condition (40) can be formulated and solved as

$$\mathbf{s}^{(\text{predicted})} = \underset{\mathbf{s} \in \mathbb{R}^N}{\operatorname{argmin}} \ \left(\mathrm{S}_2(\mathbf{s}) + \alpha || \mathbf{C}\,\mathbf{s}^{(\text{known})} - \mathbf{C}\,\mathbf{s} ||_2^2 \right). \quad (41)$$

Here, the parameter α controls the relative importance of conditions (39) and (40). Once the predicted signal $\mathbf{s}^{(\text{predicted})}$ is calculated, the unlabeled data elements are assigned to class 1 if $s_n^{(\text{predicted})} > 0$ and another class otherwise.

In classical DSP, minimization-based approaches to signal recovery and reconstruction are called *signal regularization*. They have been used for signal denoising, deblurring and recovery [42], [43], [44], [45]. In signal processing on graphs, minimization problems similar to (39) and (41) formulated with the Laplacian quadratic form (see (52) in Appendix B) are used for data classification [46], [47], [41], [48], characterization of graph signal smoothness [18] and recovery [8]. The problems (39) and (41) minimize the graph shift quadratic

form (31) and represent an alternative approach to graph signal regularization.

Experiments. We illustrate the application of graph signal regularization to data classification by solving the minimization problem (41) for two datasets. The first dataset is a collection of images of handwritten digits 4 and 9 from the NIST database [49]. Since these digits look quite similar, their automatic recognition is a challenging task. For each digit, we use 1000 grayscale images of size 28×28. The graph is constructed by viewing each image as a point in a $28^2 = 784$-dimensional vector space, computing Euclidean distances between all images, and connecting each image with six nearest neighbors by *directed* edges, so the resulting graph is a directed six-nearest neighbor graph. We consider unweighted graphs,[6] for which all non-zero edge weights are set to 1.

The second dataset is a collection of 1224 online political blogs [5]. The blogs can be either "conservative" or "liberal." The dataset is represented by a directed graph with vertices corresponding to blogs and directed edges corresponding to hyperlink references between blogs. For this dataset we also use only the unweighted graph (since we cannot assign a similarity value to a hyperlink).

For both datasets, we consider trade-offs between the two parts of the objective function in (41) ranging from 1 to 100. In particular, for each ratio of known labels 0.5%, 1%, 2%, 3%, 5%, 7%, 10% and 15%, we run experiments for 199 different values of $\alpha \in \{1/100, 1/99, \ldots, 1/2, 1, 2, \ldots, 100\}$, a total of $8 \times 199 = 1592$ experiments. In each experiment, we calculate the average classification accuracy over 100 runs. After completing all experiments, we select the highest average accuracy for each ratio of known labels.

For comparison, we consider the Laplacian quadratic form (52), and solve the minimization problems

$$\mathbf{s}^{(\text{predicted})} = \underset{\mathbf{s} \in \mathbb{R}^N}{\operatorname{argmin}} \left(\mathbf{s}^T \, \mathbf{L} \, \mathbf{s} + \alpha || \, \mathbf{C}(\mathbf{s}^{(\text{known})} - \mathbf{s}) ||_2^2 \right) \quad (42)$$

for two definitions of the Laplacian: standard (49) and normalized (50). As before, the values of parameter α vary between 0.01 and 100. Since the Laplacian matrix can only be used with undirected graphs, we convert the original directed graphs for digits and blogs to undirected graphs by making all edges undirected.

For a fair comparison with the Laplacian-based minimization (42), we also test our approach (41) on the same undirected graphs. These experiments provide an equal testing ground for the two methods. In addition, by comparing results for our approach on directed and undirected graphs, we determine whether the direction of graph edges provides additional valuable information that can improve the classification accuracy.

Results. Average classification accuracies for the handwritten digits images dataset and the blog dataset are shown,

---

[6]We have also considered weighted graphs with edge weights set to $exp(-d_{n,m}^2)$, where $d_{n,m}$ is the Euclidean distance between the nth and mth images. This is a common way of assigning edge weights for graphs that reflect similarity between objects [50], [51], [16]. Results obtained for these weighted graphs were practically indistinguishable from the results in Fig. 9 and Fig. 10 obtained for unweighted graphs.
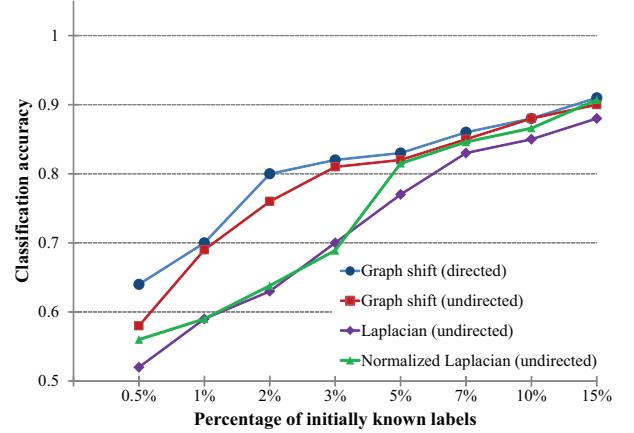

Fig. 9.   Classification accuracy of images of handwritten digits 4 and 9 using the graph shift-based regularization and Laplacian-based regularization on weighted and unweighted similarity graphs.
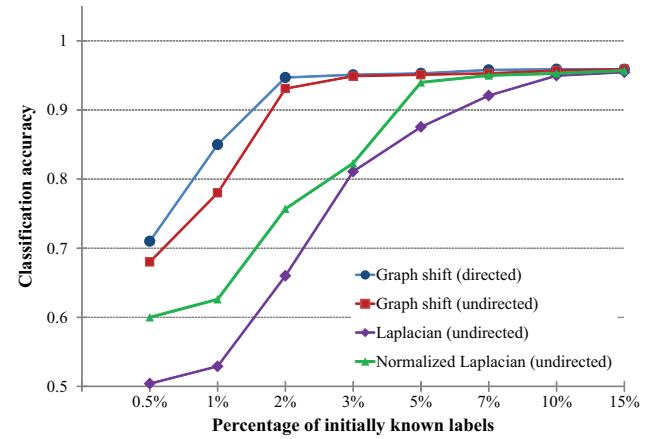

Fig. 10.   Classification accuracy of political blogs using the graph shift-based regularization and Laplacian-based regularization on an unweighted graph of hyperlink references.

respectively, in Fig. 9 and Fig. 10. For both datasets, the total variation minimization approach (41) applied to directed graphs has produced the highest accuracies. This observation demonstrates that using the information about the direction of graph edges improves the accuracy of regularization-based classification.

Furthermore, the approach (41) that uses the graph shift-based regularization significantly outperforms the Laplacian-based approach (42) on undirected graphs for the standard and normalized Laplacian matrices. In particular, for small ratios of known labels (under 5%), the differences in average accuracies can exceed 10% for image recognition and 20% for blog classification.

Discussion. The following example illustrates how classification based on signal regularization works. Fig. 11 shows a subgraph of 40 randomly selected blogs with their mutual hyperlinks. Fig. 11(a) contains true labels for these blogs obtained from [5], while the labels in Fig. 11(b) are obtained by randomly switching 7 out of 40 labels to opposite values. The frequency content of the true and synthetic labels as
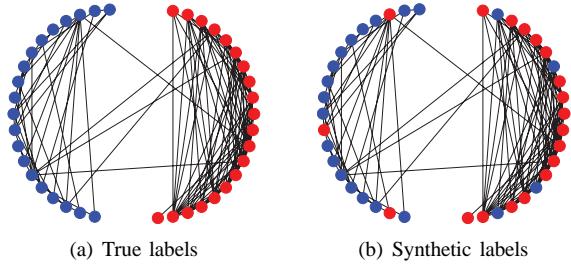
Fig. 11. A subgraph of 40 blogs labels: blue corresponds to "liberal" blogs and red corresponds to "conservative" ones. Labels in (a) form a smoother graph signal than labels in (b).
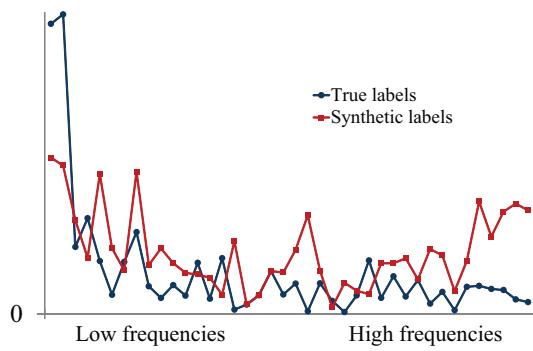

Fig. 12. Magnitudes of the spectral coefficients for graph signals formed by true and synthetic labels in Fig. 11.

signals on this subgraph are shown in Fig. 12. The true labels form a signal that has more energy concentrated in lower frequencies, i.e., has a smaller variation than the signal formed by the synthetic labels. This observation supports our assumption that the solution to the regularization problem (41) should correspond to correct label assignment.

Incidentally, this assumption also explains why the maximum classification accuracy achieved in our experiments is 96%, as seen in Fig. 10. We expect that every blog contains more hyperlinks to blogs of its own type than to blogs of the opposite type. However, after inspecting the entire dataset, we discovered that 50 blogs out of 1224, i.e., 4% of the total dataset, do not obey this assumption. As a result, 4% of all blogs are always misclassified, which results in maximum achievable accuracy of 96%.

VII. CONCLUSIONS

In this paper, we defined the concepts of low-, high-, and band-pass graph signals and graph filters. These definitions are not straightforward extensions of the concepts from the traditional discrete signal processing. Rather, we defined them using the concept of frequencies in digital signal processing on graphs introduced in [9]. We proposed a novel definition of total variation of graph signals that measures the difference between a signal and its shifted version. We then used the total variation to order the graph frequencies and to define low- and high-pass graph signals and graph filters. We demonstrated how to design filters with specified

frequency response by finding least squares approximations to solutions of systems of linear algebraic equations. We applied these concepts and methodologies to the analysis and learning of real-world datasets. We studied detection of corrupted data in the dataset of temperature measurements collected by a network of 150 weather stations across the U.S. during one year, and classification of handwritten digit images from the NIST database [49] and hyperlinked documents [5]. The experiments showed that the techniques presented in this paper are promising in problems like detecting sensor malfunctions, graph signal regularization, and classification of partially labeled data.

APPENDIX A: JORDAN DECOMPOSITION

An arbitrary matrix $\mathbf{A} \in \mathbb{C}^{N \times N}$ has $M \leq N$ distinct eigenvalues $\lambda_0, \ldots, \lambda_{M-1}$. Each eigenvalue λ_m has D_m corresponding eigenvectors $\mathbf{v}_{m,0}, \ldots, \mathbf{v}_{m,D_m-1}$ that satisfy

$$(\mathbf{A} - \lambda_m \mathbf{I}_N)\mathbf{v}_{m,d} = 0.$$

Moreover, each eigenvector $\mathbf{v}_{m,d}$ can generate a *Jordan chain* of $R_{m,d} \geq 1$ *generalized eigenvectors* $\mathbf{v}_{m,d,r}$, $0 \leq r < R_{m,d}$, where $\mathbf{v}_{m,d,0} = \mathbf{v}_{m,d}$, that satisfy

$$(\mathbf{A} - \lambda_m \mathbf{I})\mathbf{v}_{m,d,r} = \mathbf{v}_{m,d,r-1}. \tag{43}$$

All eigenvectors and corresponding generalized eigenvectors are linearly independent.

For each eigenvector $\mathbf{v}_{m,d}$ and its Jordan chain of size $R_{m,d}$, we define a *Jordan block* matrix of dimensions $R_{m,d} \times R_{m,d}$ as

$$J_{R_{m,d}}(\lambda_m) = \begin{bmatrix} \lambda_m & 1 & & \\ & \lambda_m & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_m \end{bmatrix} \in \mathbb{C}^{R_{m,d} \times R_{m,d}}. \tag{44}$$

By construction, each eigenvalue λ_m is associated with D_m Jordan blocks, each of dimension $R_{m,d} \times R_{m,d}$, where $0 \leq d < D_m$. Next, for each eigenvector $\mathbf{v}_{m,d}$, we collect its Jordan chain into a $N \times R_{m,d}$ matrix

$$\mathbf{V}_{m,d} = \begin{bmatrix} \mathbf{v}_{m,d,0} & \cdots & \mathbf{v}_{m,d,R_{m,d}-1} \end{bmatrix}. \tag{45}$$

We concatenate all blocks $\mathbf{V}_{m,d}$, $0 \leq d < D_m$ and $0 \leq m < M$, into one block matrix

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_{0,0} & \cdots & \mathbf{V}_{M-1,D_{M-1}} \end{bmatrix}, \tag{46}$$

so that the block $\mathbf{V}_{m,d}$ is at position $\sum_{k=0}^{m-1} D_k + d$ in this matrix. Then matrix $\mathbf{A}$ is written in its *Jordan decomposition* form as

$$\mathbf{A} = \mathbf{V} \mathbf{J} \mathbf{V}^{-1}, \tag{47}$$

where the block-diagonal matrix

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{R_{0,0}}(\lambda_0) & & \\ & \ddots & \\ & & \mathbf{J}_{R_{M-1,D_{M-1}}}(\lambda_{M-1}) \end{bmatrix} \tag{48}$$

is called the *Jordan normal form* of $\mathbf{A}$. The columns of $\mathbf{V}$, i.e., all eigenvectors and generalized eigenvectors of $\mathbf{A}$, are called the *Jordan basis* of $\mathbf{A}$.

APPENDIX B: CONNECTION WITH LAPLACIAN-BASED VARIATION

The Laplacian matrix for an undirected graph $G = (\mathcal{V}, \mathbf{A})$ with real, non-negative edge weights $\mathbf{A}_{n,m}$ is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \qquad (49)$$

where $\mathbf{D}$ is a diagonal matrix with diagonal elements

$$\mathbf{D}_{n,n} = \sum_{m=0}^{N-1} \mathbf{A}_{n,m}.$$

Alternatively, the (normalized) Laplacian matrix is defined as

$$\mathbf{L} = \mathbf{I}_N - \mathbf{D}^{-1/2} \mathbf{A} \, \mathbf{D}^{-1/2}. \qquad (50)$$

The Laplacian matrix (49) has real non-negative eigenvalues $0 = \beta_0 < \beta_1 \leq \beta_2 \leq \ldots \leq \beta_{N-1}$ and a complete set of corresponding orthonormal eigenvectors $\mathbf{u}_n$ for $0 \leq n < N$.

Similarly to the DSP_G definition of the graph Fourier transform (8), the Laplacian-based Fourier transform expands a graph signal $\mathbf{s}$ into the eigenbasis of $\mathbf{L}$ [8]. The total variation is defined as

$$\mathrm{TV_L}(\mathbf{s}) = \sum_{n=0}^{N-1} \left(\sum_{m \in \mathcal{N}_n} \mathbf{A}_{n,m} (s_n - s_m)^2 \right)^{1/2}, \qquad (51)$$

and the graph Laplacian quadratic form is

$$\mathrm{S}_2^{(L)}(\mathbf{s}) = \mathbf{s}^T \mathbf{L} \, \mathbf{s}. \qquad (52)$$

In particular, the Laplacian quadratic form (51) of a Fourier basis vector is

$$\mathrm{S}_2^{(L)}(\mathbf{u}_n) = \beta_n. \qquad (53)$$

It imposes the following order of the Laplacian Fourier basis from the lowest frequency to the highest:

$$\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{N-1}. \qquad (54)$$

For a general graph, the total variation (18) and the graph shift quadratic form (31) are different from the (51) and (52). However, as we demonstrate in the following theorem, the DSP_G and the Laplacian-based approach to signal processing on graphs lead to the same graph Fourier basis, notions of low and high frequencies, and frequency ordering on any regular[7] graph.

Theorem 3: The quadratic forms (31) and (52) induce the same ordering on the graph Fourier basis for regular graphs.

Proof: Consider a d-regular graph with adjacency matrix $\mathbf{A}$. Since the Laplacian matrix (49) can be defined only for undirected graphs with real non-negative edge weights, we also require that $\mathbf{A} = \mathbf{A}^T$ and has only real non-negative entries. Hence, $\mathbf{A}$ has real eigenvalues and a complete set of orthonormal eigenvectors [34], and its Jordan decomposition (47) becomes the eigendecomposition

$$\mathbf{A} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T,$$

where $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues. Since the graph is d-regular, its Laplacian matrix (49) satisfies

$$\mathbf{L} = d \, \mathbf{I} - \mathbf{A} = \mathbf{V}(d \, \mathbf{I} - \mathbf{\Lambda}) \mathbf{V}^T.$$

Hence, $\mathbf{L}$ and $\mathbf{A}$ have the same eigenvectors $\mathbf{u}_n = \mathbf{v}_n$, i.e., the same graph Fourier basis. The corresponding eigenvalues are

[7] All vertices of a d-regular graph have the same degree d, so that $\sum_{m=0}^{N-1} \mathbf{A}_{n,m} = d$.

$\beta_m = d - \lambda_m$. Since the smallest eigenvalue of $\mathbf{L}$ is $\beta_0 = 0$, we also obtain $\lambda_{\max} = d$.

The graph shift quadratic form (31) of the eigenvector $\mathbf{v}_m$ satisfies

$$
\begin{aligned}
\mathrm{S}_2(\mathbf{u}_m) &= \frac{1}{2} \left\| \left(\mathbf{I} - \frac{1}{d} \mathbf{A} \right) \mathbf{u}_m \right\|_2^2 \\
&= \frac{1}{2} \left(1 - \frac{\lambda_m}{d} \right)^2 \\
&= \frac{1}{2d^2} (d - \lambda_m)^2 \\
&= \frac{1}{2d^2} \beta_m^2 \\
&= \frac{1}{2d^2} \left(\mathrm{S}_2^{(L)}(\mathbf{s}) \right)^2. \qquad (55)
\end{aligned}
$$

Since $\beta^2/(2d^2)$ is a monotonically increasing function for $\beta \geq 0$, it follows from (55) that ordering the graph Fourier basis $\mathbf{u}_n$, $0 \leq n < N$, by increasing values of the quadratic form (31) leads to the same order as (54). Hence, the notions of low and high frequencies, and frequency orderings from lowest to highest coincide on regular graphs for the DSP_G and the Laplacian-based approach. ∎

REFERENCES

[1] M. Jackson, *Social and Economic Networks*, Princeton Univ., 2008.
[2] M. Newman, *Networks: An Introduction*, Oxford Univ. Press, 2010.
[3] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets*, Cambridge Univ. Press, 2010.
[4] M. E. J. Newman, "Coauthorship networks and patterns of scientific collaboration," *Proc. Nat. Acad. Sci.*, vol. 101, pp. 5200–5205, 2004.
[5] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 U.S. election: Divided they blog," in *Proc. Int. Workshop on Link Discovery*, 2005, pp. 36–43.
[6] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Comp. Networks and ISDN Syst.*, vol. 30, pp. 107–117, 1998.
[7] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, Prentice Hall, 2nd edition, 1999.
[8] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs," *IEEE Signal Proc. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.
[9] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Proc.*, vol. 61, no. 7, pp. 1644–1656, 2013.
[10] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Graph Fourier transform," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Proc.*, 2013, pp. 6167–6170.
[11] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Graph filters," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Proc.*, 2013, pp. 6163–6166.
[12] F. R. K. Chung, *Spectral Graph Theory*, AMS, 1996.
[13] J. F. Tenenbaum, V. Silva, and J. C. Langford, "A global geometric framework for nonlinear dmensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.
[14] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.
[15] R. R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. J. Warner, and S. W. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," *Proc. Nat. Acad. Sci.*, vol. 102, no. 21, pp. 7426–7431, 2005.
[16] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *J. Appl. Comp. Harm. Anal.*, vol. 30, no. 2, pp. 129–150, 2011.
[17] S. K. Narang and A. Ortega, "Perfect reconstruction two-channel wavelet filter banks for graph structured data," *IEEE Trans. Signal Proc.*, vol. 60, no. 6, pp. 2786–2799, 2012.
[18] A. Agaskar and Y. Lu, "A spectral graph uncertainty principle," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4338–4356, 2013.

[19] V. Ekambaram, G. Fanti, B. Ayazifar, and K. Ramchandran, "Multiresolution graph signal processing via circulant structures," in *Proc. IEEE DSP/SPE Workshop*, 2013.

[20] V. Ekambaram, G. Fanti, B. Ayazifar, and K. Ramchandran, "Circulant structures and graph signal processing," in *Proc. IEEE Int. Conf. Image Proc.*, 2013.

[21] X. Zhu and M. Rabbat, "Approximating signals supported on graphs," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Proc.*, 2012, pp. 3921–3924.

[22] M. Püschel and J. M. F. Moura, "The algebraic approach to the discrete cosine and sine transforms and their fast algorithms," *SIAM J. Comp.*, vol. 32, no. 5, pp. 1280–1316, 2003.

[23] M. Püschel and J. M. F. Moura, "Algebraic signal processing theory," http://arxiv.org/abs/cs.IT/0612077.

[24] M. Püschel and J. M. F. Moura, "Algebraic signal processing theory: Foundation and 1-D time," *IEEE Trans. Signal Proc.*, vol. 56, no. 8, pp. 3572–3585, 2008.

[25] M. Püschel and J. M. F. Moura, "Algebraic signal processing theory: 1-D space," *IEEE Trans. Signal Proc.*, vol. 56, no. 8, pp. 3586–3599, 2008.

[26] M. Püschel and J. M. F. Moura, "Algebraic signal processing theory: Cooley-Tukey type algorithms for DCTs and DSTs," *IEEE Trans. Signal Proc.*, vol. 56, no. 4, pp. 1502–1521, 2008.

[27] M. Püschel and M. Rötteler, "Fourier transform for the directed quincunx lattice," in *Proc. ICASSP*, 2005, vol. 4, pp. 401–404.

[28] M. Püschel and M. Rötteler, "Fourier transform for the spatial quincunx lattice," in *Proc. ICIP*, 2005, vol. 2, pp. 494–497.

[29] M. Püschel and M. Rötteler, "Algebraic signal processing theory: 2-D hexagonal spatial lattice," *IEEE Trans. on Image Proc.*, vol. 16, no. 6, pp. 1506–1521, 2007.

[30] A. Sandryhaila, J. Kovacevic, and M. Püschel, "Algebraic signal processing theory: 1-D Nearest-neighbor models," *IEEE Trans. on Signal Proc.*, vol. 60, no. 5, pp. 2247–2259, 2012.

[31] A. Sandryhaila, J. Kovacevic, and M. Püschel, "Algebraic signal processing theory: Cooley-Tukey type algorithms for polynomial transforms based on induction," *SIAM J. Matrix Analysis and Appl.*, vol. 32, no. 2, pp. 364–384, 2011.

[32] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, 4th edition, 2002.

[33] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 3rd edition, 2008.

[34] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, Academic Press, 2nd edition, 1985.

[35] F. R. Gantmacher, *Matrix Theory*, vol. I, Chelsea, 1959.

[36] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*, Signal Processing, Prentice Hall, Englewood Cliffs, NJ, 1995.

[37] A. Bovik, *Handbook of Image and Video Processing*, Academic Press, 2nd edition, 2005.

[38] "National Climatic Data Center," 2011, ftp://ftp.ncdc.noaa.gov/pub/data/gsod.

[39] T. J. Rivlin, *An Introduction to the Approximation of Functions*, Dover Publications, 1969.

[40] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley, 2nd edition, 2000.

[41] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*, MIT Press, 2006.

[42] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, no. 1–4, pp. 259–268, 1992.

[43] T. F. Chan, S. Osher, and J. Shen, "The digital TV filter and nonlinear denoising," *IEEE Trans. Image Proc.*, vol. 10, no. 2, pp. 231–241, 2001.

[44] J. P. Oliveira, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "Adaptive total variation image deblurring: A majorization-minimization approach," *Signal Proc.*, vol. 89, no. 9, pp. 1683–1693, 2009.

[45] I. W. Selesnick and P. Chen, "Total variation denoising with overlapping group sparsity," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Proc.*, 2013.

[46] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. Neural Inf. Proc. Syst.*, 2004, pp. 321–328.

[47] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and semi-supervised learning on large graphs," in *Proc. Conf. Learn. Th.*, 2004, pp. 624–638.

[48] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 985–992.

[49] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[50] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comp.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[51] R. R. Coifman and M. Maggioni, "Diffusion wavelets," *Appl. Comp. Harm. Anal.*, vol. 21, no. 1, pp. 53–94, 2006.

Aliaksei Sandryhaila (S'06–M'10) received a B.S. degree in Computer Science from Drexel University, Philadelphia, PA, in 2005, and a Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University (CMU), Pittsburgh, PA, in 2010. He is currently a Research Scientist in the Department of Electrical and Computer Engineering at CMU. Previously, he was a Postdoctoral Researcher in the Department of Electrical and Computer Engineering at CMU and a Senior Research Scientist at SpiralGen in Pittsburgh, PA. His research interests include signal processing, large-scale data analysis, machine learning machine learning, biomedical signal and image analysis, high-performance computing, and design and optimization of fast algorithms.

José M. F. Moura (S'71–M'75–SM'90–F'94) is the Philip and Marsha Dowd University Professor at Carnegie Mellon University in the Departments of Electrical and Computer Engineering and Biomedical Engineering (by courtesy). In the academic year 2013-14, he is a Visiting Professor at New York University (NYU) with the Center for Urban Science and Progress (CUSP). He holds an EE degree from Instituto Superior Técnico (IST, Portugal) and MSc, EE, and DSc degrees from the Massachusetts Institute of Technology. He has held visiting professor appointments with MIT and NYU and was on the faculty of IST (Portugal). He is currently ECE Associate Department Head for Research and Strategic Initiatives. He founded and directs the CMU Information and Communication Technologies Institute (ICTI) that manages the CMU/Portugal Program, and co-founded the Center for Sensed Critical Infrastructure Research (CenSIR). With CMU colleagues, he co-founded SpiralGen that commercializes, under a CMU license, the Spiral technology (www.spiral.net). His interests are on statistical signal and image processing and data science. He holds eleven issued patents and has published over 450 papers. Moura has been an IEEE Board Director, IEEE Division IX Director, member of several IEEE Boards, President of the IEEE Signal Processing Society (SPS), and Editor-in-Chief for the Transactions on Signal Processing. He has been on the Editorial Boards of several IEEE and ACM Journals. He received several awards including the IEEE Signal Processing Society Technical Achievement Award and the IEEE Signal Processing Society Award for outstanding technical contributions and leadership in signal processing. Moura is a Fellow of the IEEE, a Fellow of AAAS, a corresponding member of the Academy of Sciences of Portugal, and a member of the US National Academy of Engineering.