Ljubiša Stanković, Danilo P. Mandic, Miloš Daković,
Ilya Kisil, Ervin Sejdić, and Anthony G. Constantinides

Understanding the Basis of Graph Signal Processing via an Intuitive Example-Driven Approach

Graphs are irregular structures that naturally account for data integrity; however, traditional approaches have been established outside signal processing and largely focus on analyzing the underlying graphs rather than signals on graphs. Given the rapidly increasing availability of multisensor and multinode measurements, likely recorded on irregular or ad hoc grids, it would be extremely advantageous to analyze such structured data as "signals on graphs" and thus benefit from the ability of graphs to incorporate spatial sensing awareness, physical intuition, and sensor importance, together with the inherent "local versus global" sensor association. The aim of this lecture note is, therefore, to establish a common language between graph signals that are observed in irregular signal domains and some of the most fundamental paradigms in digital signal processing (DSP), such as spectral analysis, system transfer function, digital filter design, parameter estimation, and optimal denoising.

Scope

The move to gather more data from our environment, for our applications, health, and general well-being, is an established fact. The increase in the modalities of data acquisition and signal sensors, together with the corresponding increase in their structure and the complexity of information, has highlighted the need for a shift in thinking and new analytical frameworks. This need becomes even clearer when we take into consideration the intermodality and interlocality attributes and their interactions, which effectively call for radically new data analytics approaches. Such a paradigm shift is provided by graph signal theory, a framework that goes beyond the standard "vertices, links, and their structural properties" components of a graph. It is the aim of this lecture note to introduce a unifying perspective based on a real-world multisensor problem.

This is achieved through a physically meaningful and intuitive real-world example of geographically distributed estimation of multisensor temperature measurements. A similar spatial multisensor arrangement has already been widely used in signal processing curricula to introduce minimum variance estimators and Kalman filters, and by adopting this framework we facilitate a seamless integration of graph theory into the curriculum of existing DSP courses. By bridging the gap between standard approaches and graph signal processing, we also show that standard methods can be thought of as special cases of their graph counterparts, evaluated online graphs. This article was, therefore, primarily written in response to the urgent need to bring graph signal processing into the curricula of existing statistical signal processing and machine learning courses, and this material corresponds to a 2-h lecture that could come at the very end of the standard lecture course syllabi. We also hope that our approach will not only help to demystify graph-theoretic approaches for education purposes but will also empower practitioners and researchers to explore a whole host of otherwise prohibitive modern applications. The supporting material, lecture slides, data, and MATLAB code can be found at http://www.tfsa.ac.me/ln-gsp and http://www.commsp.ee.ic.ac.uk/~mandic/DSP_ML_Education.htm. **<AU: Kindly provide a valid URL; the last URL appears to be broken.>**

Relevance

In classical signal processing, the signal domain is determined by equidistant time instants or by a set of spatial sensing points on a uniform grid. Increasingly, however, the actual data sensing domain may not even be related to the physical dimensions of time and/or space, and it typically exhibits various forms of irregularity, as, for example, in social or web-related networks, where the sensing points and their connectivity pertain to specific objects or nodes and the ad hoc topology of their links. It should be noted that even for the data acquired in well-defined time and space domains, the introduction of new relations between the signal samples, through graphs, may yield new insights into the analysis and provide enhanced data processing (for example, based on local similarity, through neighborhoods). We therefore set out to show that the advantage of graphs over classical data domains is that graphs account naturally and

Digital Object Identifier 10.1109/MSP.2019.2929832
Date of publication: xxxxxxx

comprehensively for irregular data relations in the problem definition, together with the corresponding data connectivity in the analysis.

Through a real-world temperature estimation example, we show that graph signal and information processing is particularly well suited to making sense from data acquired over irregular data domains, which can be achieved, for example, by leveraging intuitions developed on Euclidean domains, by employing analogies with other irregular domains such as polygon meshes and manifolds, or by learning the mutual connectivity patterns from the available sets of data. In many emerging applications, for example, big data, this also introduces a number of new challenges:

- Basic concepts must be revisited to accommodate structured but often incomplete information.
- New physically meaningful frameworks, specifically tailored for heterogeneous data sources, are required.
- Tradeoffs between performance and numerical requirements are a prerequisite when operating in real time, especially given often combinatorial ways to analyze graphs.

The common language and enhanced intuition between the standard approaches and their graph counterparts, illuminated in this article through the relationships between the vertex and time domains, and between the corresponding eigenspectrum and frequency domains, may be naturally generalized to address the aforementioned challenges and spur further developments in the curricula on statistical signal processing, graph signal processing, and big data.

Prerequisites

This lecture note assumes a basic knowledge of linear algebra and DSP.

Problem statement and solutions

History of graph-theoretic application

Graph theory, as a branch of mathematics, has existed for almost three centuries. The beginning of graph theory applications in electrical engineering dates back to the mid-19th century and the definition of Kirchhoff's laws. Owing to their inherent "spatial awareness," graph models have since become a de facto standard for data analytics across the science and engineering areas, including chemistry, operational research, social networks, and computer sciences.

A systematic account of graph theory as an optimization tool can be attributed to the seminal book by Nicos Christofides of Imperial College London, published in 1975 [1]. Soon after graph theory gained prominence in general optimization, it was very natural to explore its application in signal processing and related data analytics areas [2]. **<AU: Please check whether the preceding edited sentence conveys the intended meaning.>** Indeed, perhaps the first lecture course to teach graph theory to then emerging communication networks and channel coding student cohort was introduced by the author Anthony Constantinides in the 1970s. This helped to establish and formalize the connections between general optimization and the topology of a communication network and has spurred further applications in image processing [3].

After a relative lull in the field over more than two decades, current developments in graph theory owe their prominence to the emergence of modern data sources, such as large-scale sensor and social networks, which inherently provide rich underlying physical, social, and geographic structures. These require new ways to establish statistical inference, such as through data analytics on graphs and within a new, fast-maturing field of graph signal processing [4]–[10].

An illustrative example

Graph signal processing represents quite a general mathematical formalism that, while different from classic concepts, does admit the development of graph-domain counterparts of well-established DSP paradigms. It would, therefore, be quite valuable to introduce such a general concept in an inductive and intuitive way, through a simple yet general enough and well-understood example of a commonly considered topic in classical DSP.

To this end, consider a multisensor setup, shown in Figure 1, for measuring temperature field in a known geographical region; such a setup is typically used in the context of minimum-variance estimation and Kalman filters. The temperature sensing locations are chosen according to the significance of a particular geographic area to local users, with $N = 64$ sensing points in total, as shown in Figure 1(a). The temperature field is denoted by $\{x(n)\}$, and a snapshot of its values is given in Figure 1(b). Each such measured temperature signal can then be mathematically expressed as

$$x(n) = s(n) + \varepsilon(n), \tag{1}$$

where $s(n)$ is the true temperature that would have be obtained in ideal measuring conditions, while the term $\varepsilon(n)$ comprises the adverse effects of the local environment on sensor readings or faulty sensor activity and is referred to as *noise* in the sequel. For illustrative purposes, the noise $\varepsilon(n)$ was modeled in our study as a realization of white, zero-mean, Gaussian process, with standard deviation $\sigma_\varepsilon = 4$, that is $\varepsilon \sim \mathcal{N}(0, 16)$, which was added to the signal, $s(n)$, to yield the signal-to-noise ratio in $\{x(n)\}$ of $\text{SNR}_0 = 14.2$ dB.

Remark 1

Classical signal processing requires an arrangement of the quintessentially spatial temperature samples in Figure 1(a) into a line structure shown in Figure 1(c). Obviously, such "lexicographic" ordering is not amenable to exploiting the spatial information related to the actual sensor arrangement, dictated by the terrain. For example, this renders classical analyses of this temperature field inadequate (or at best suboptimal), as in this case the performance critically depends on the chosen sensor ordering scheme. This exemplifies that even a most routine temperature measurement setup requires a more complex estimation structure than the simple linear one that corresponds to the classical signal processing framework, as in Figure 1(c).

To introduce a "situation-aware" noise reduction scheme for the temperature field in Figure 1, we proceed to explore

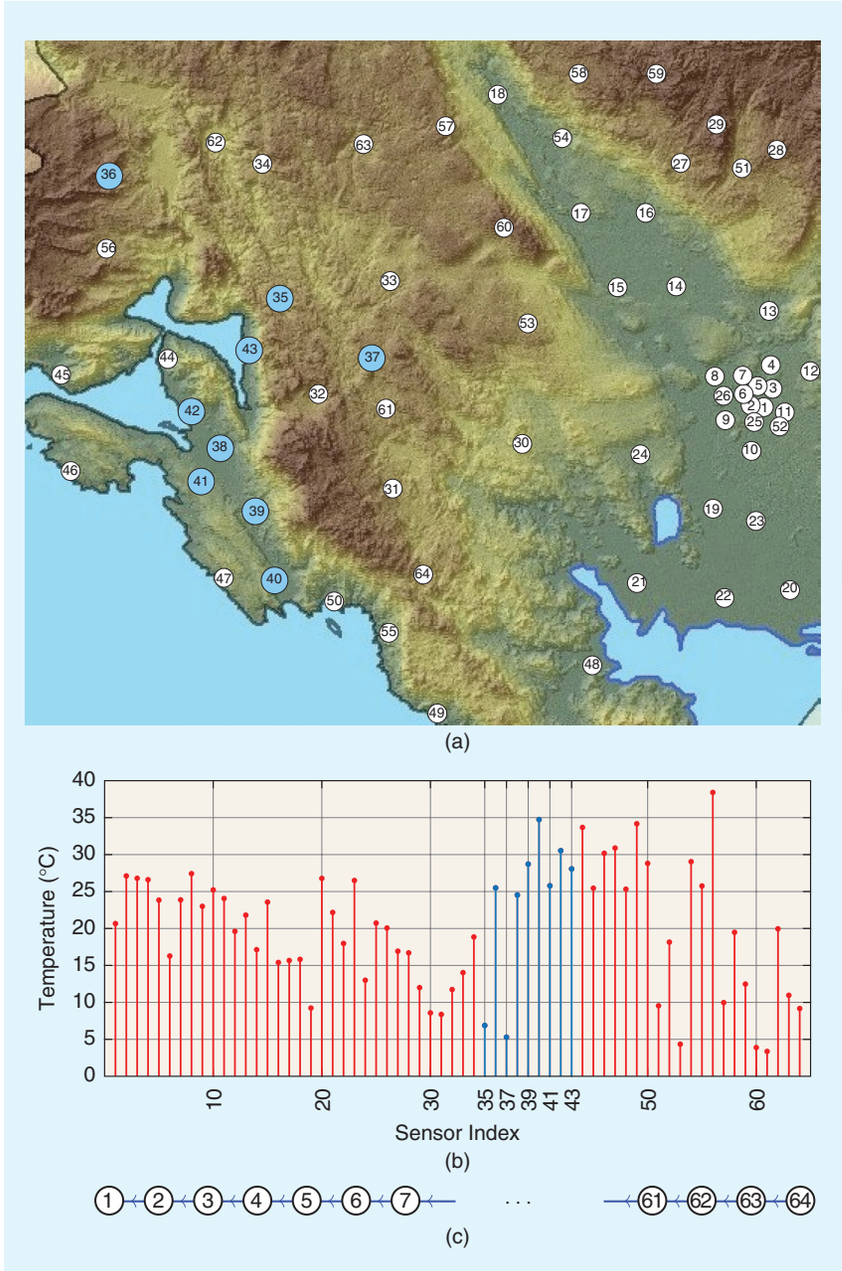


FIGURE 1. An illustration of temperature sensing as a classical signal processing problem. (a) A map of sensing locations in a geographic region along the Adriatic Sea (Montenegro). (b) A graph of the temperatures measured at $N = 64$ sensing locations. In standard signal processing, the spatial sensor index is used for the horizontal axis and serves as the signal domain. (c) This domain can also be interpreted as a directed line graph, but lacks physical intuition, as for example, sensor 37 (mountains) is followed by sensor 38 (coast), with a drastic difference in temperature. ***<AU: Please confirm whether the updated figure caption is acceptable.>***

a graph-theoretic framework for this class of problems, starting from a local signal average operator. In classical signal processing, this can be achieved through a moving average operator, for example, through averaging across the neighboring data samples, or equivalently neighboring nodes, as in the line graph in Figure 1(c), and for each sensing point. Physically, such a local neighborhood should indeed include geographically close neighboring sensing points, but these should also exhibit similar meteorological properties defined by the distance, altitude difference, and other terrain properties. In other words, since the sensor network in Figure 1 measures a set of related temperatures from irregularly spaced sensors, an effective estimation strategy should include domain knowledge, which it is not possible to achieve with standard DSP (line graph).

Problem setup

Consider the local neighborhoods for the sensing points $n = 20$, 29, 37, and 41, shown in Figure 2(a). The cumulative temperature for each sensing point is then given by

$$y(n) = \sum_{m \text{ at and around } n} x(m),$$

so that the local average temperature for a sensing point n may be easily obtained by dividing the cumulative temperature, $y(n)$, by the number of sensing points involved. For example, for the sensing points $n = 20$ and $n = 37$, presented in Figure 2(a), the "domain knowledge aware" local estimation takes the form

$$y(20) = x(20) + x(19) + x(22) + x(23) \tag{2}$$

$$y(37) = x(37) + x(32) + x(33) + x(35) + x(61). \tag{3}$$

For convenience, the full set of relations among the sensing points can now be rearranged into a matrix form, to give

$$\mathbf{y} = \mathbf{x} + \mathbf{A}\mathbf{x}, \tag{4}$$

where the matrix $\mathbf{A}$ indicates the connectivity structure of the neighboring sensing locations that should be involved in the calculation for each estimate, $y(n)$. The matrix $\mathbf{A}$ is therefore referred to as the *connectivity* or *adjacency matrix* of a graph. Its elements are either 1 (if the corresponding vertices are related) or 0 (if they are not related). Figure 2(b) shows the sensing locations with the corresponding connectivity patterns for the temperature estimation scenario in Figure 2(a). From (2) we can observe, for example, that the 20th row of the adjacency matrix $\mathbf{A}$ will have all zero elements, except for $A_{20,19} = 1$, $A_{20,22} = 1$, and $A_{20,23} = 1$ (see the supplementary materials in IEEE *Xplore* for more information).

This simple real-world example can be interpreted within the graph signal processing framework as follows:

- The sensing points where the signal is measured are designated as the *graph vertices* (see Figure 1).
- The vertex-to-vertex lines that indicate connectivity among the sensing points are called the *graph edges*.
- The vertices and edges form *a graph*, as in Figure 2(b), a new and very structurally rich signal domain.
- The graph, rather than a standard vector of sensing points, is then used for analyzing and processing data, as it is equipped with spatial and physical awareness.
- The measured temperatures are now interpreted as signal samples on a graph, as shown in Figure 3.
- Similar to traditional signal processing, this new graph signal may have many realizations on the same graph and may include noise, thus paving the way for statistical approaches.
- Through relation (4), we have therefore introduced a simple system for graph signals that performs physically and spatially aware signal averaging (a linear first-order system for a graph signal).

To emphasize our trust in a particular sensor and to model mutual sensor relevance, a weighting scheme may be imposed on the edges (connectivity) between the sensing points, in the form

$$y(n) = x(n) + \sum_{m \neq n} W_{nm} x(m). \qquad (5)$$

The weight W_{nm} indicates the strength of the coupling between signal values at the sensing points n and m; it has the value $W_{nm} = 0$ if the points n and m are not related or if $n = m$. We have now arrived at a weighted graph, whereby each edge has an associated weight, W_{nm}, which adds "mutual sensor relevance" information to the already established spatial awareness modeled by the edges. This equips graph signal models with additional flexibility. In our example, a matrix form of a weighted cumulative graph signal now becomes

$$\mathbf{y} = \mathbf{x} + \mathbf{W}\mathbf{x}. \qquad (6)$$

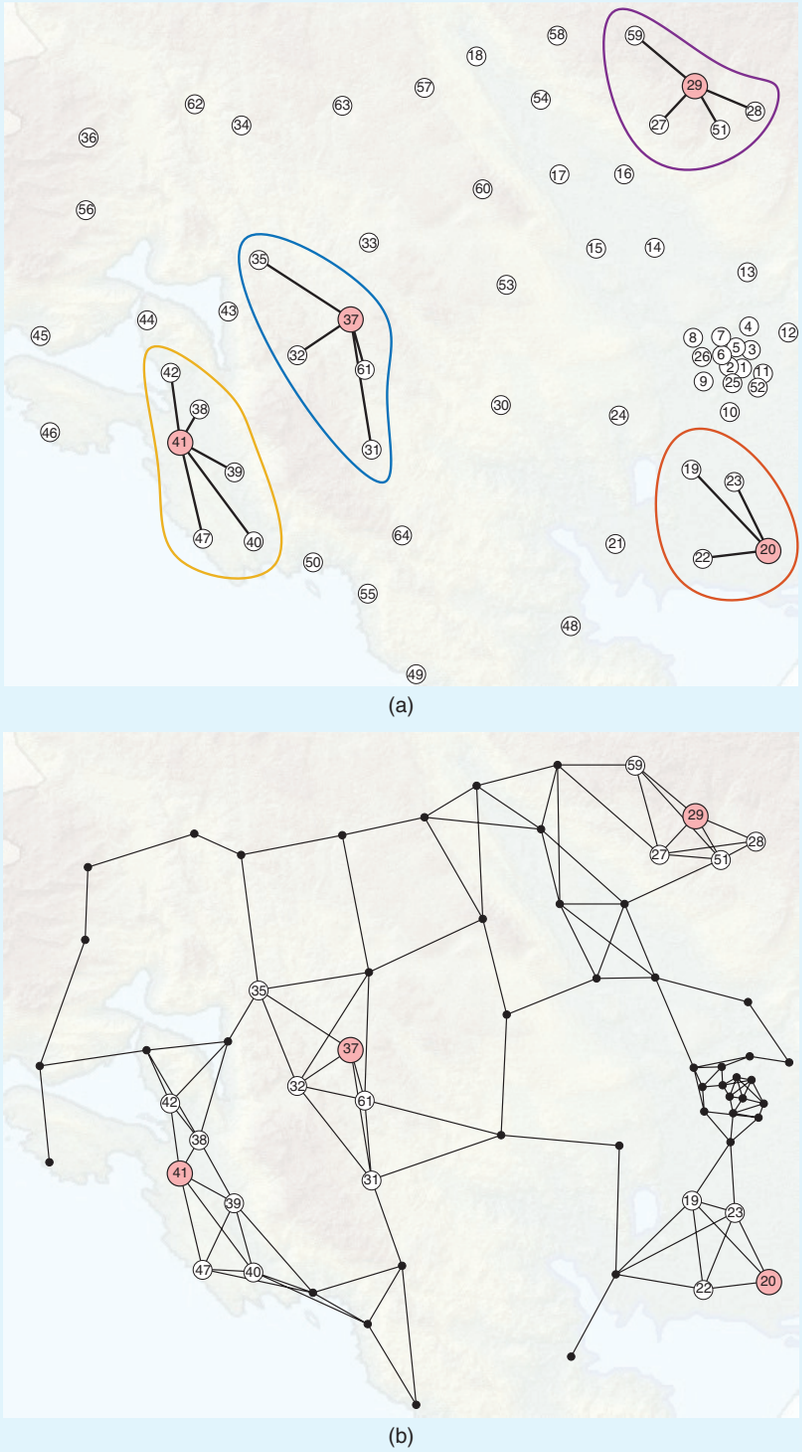


FIGURE 2. A temperature estimation setup as a domain-aware graph signal processing problem. (a) The local neighborhoods for the sensing points $n = 20$, 29, 37, and 41. These neighborhoods are chosen using "domain knowledge," dictated by local terrain and by taking into account the distance and altitude of sensors. This allows for the neighboring sensors for each of these sensing locations (vertices) to be chosen in a physically meaningful way, with their relation indicated by the connectivity lines, called *edges*. (b) The local neighborhoods for all sensing vertices from Figure 1, presented in a graph form.

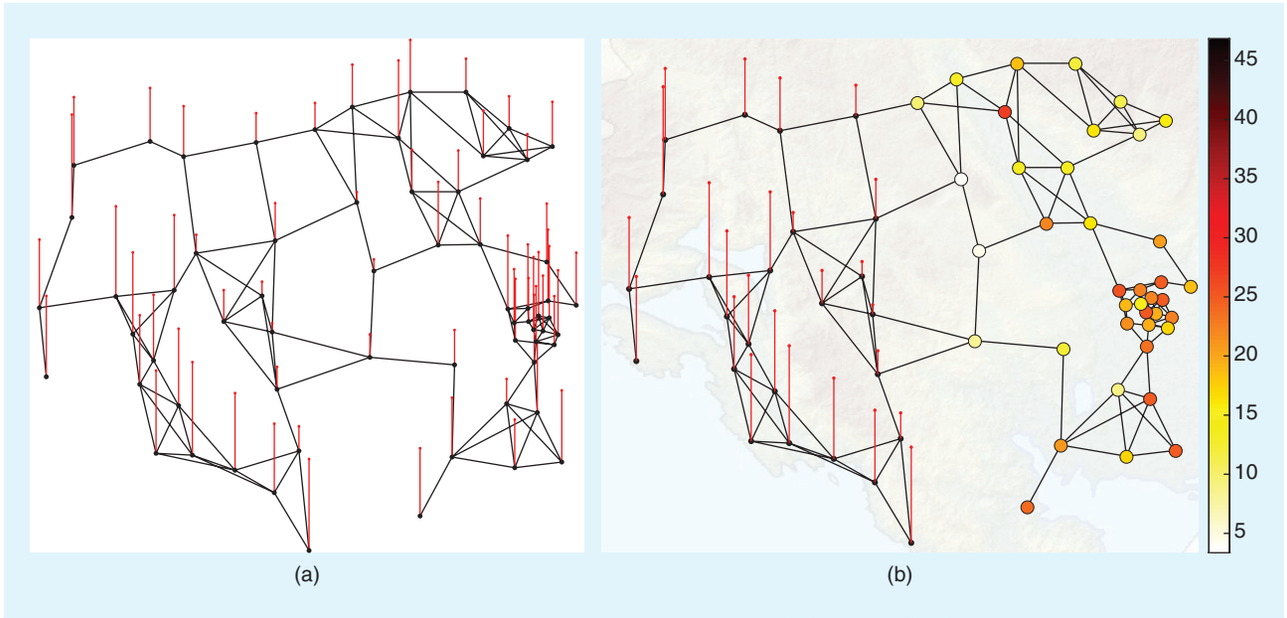


FIGURE 3. The move from a multisensor measurement to a graph signal. (a) The temperature field is represented on a graph that combines spatially unaware measurements from Figure 1(b) and the physically relevant graph topology from Figure 2(b). (b) The graph signal intensity may also be designated by the vertex color, as in the right half of the panel.

To produce unbiased estimates, instead of the cumulative sums in (4) and (5), the weighting coefficients within the estimate for each $y(n)$ should sum up to unity. This may be achieved through a normalized form of (6), given by

$$\mathbf{y} = \frac{1}{2}(\mathbf{x} + \mathbf{D}^{-1}\mathbf{W}\mathbf{x}), \tag{7}$$

where the elements of the diagonal normalization matrix, $\mathbf{D}$, called *the degree matrix*, are $D_{nn} = \sum_m W_{nm}$ while the term $\mathbf{D}^{-1}\mathbf{W}$ is referred to as a *random walk or diffusion weight matrix*. When this simple normalized first-order system is employed to filter the original noisy signal from Figure 3, the SNR = 20.2 dB was achieved—an improvement of 6 dB over the original signal-to-noise ratio, $\text{SNR}_0 = 14.2$ dB.

Another important operator for graph signal processing is the graph Laplacian, $\mathbf{L}$, which is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{W}.$$

Remark 2

A graph is fully specified by the set of its vertices and their connectivity scheme (designated by edges). The edges may be defined by the adjacency matrix, $\mathbf{A}$, with $A_{mn} \in \{0, 1\}$, for unweighted graphs or by the "connectivity strength" matrix, referred to as the *weight matrix*, $\mathbf{W}$, with $W_{mn} \in \mathbb{R}^+$, for weighted graphs. The degree matrix, $\mathbf{D}$, and the Laplacian matrix, $\mathbf{L}$, with $L_{mn} \in \mathbb{R}$, are defined using the adjacency/weight matrix. When the relations between all pairs of vertices are mutually symmetric, then all the matrices involved are also symmetric, and such graphs are called *undirected*. If that is not the case, then the adjacency/weight matrix is not symmetric, and such graphs are called *directed graphs*.

The previously introduced graph framework is quite general and admits application to many different scenarios. For example, when performing an opinion poll within a social network, the members of that social network are treated as vertices (data acquisition points), while their friendship relations are represented by the edges that model graph connectivity, with the member attributes playing the role of graph signal values.

Remark 3

The definition of an appropriate graph structure is a prerequisite for physically meaningful and computationally efficient graph signal processing applications. Three important classes of problems, regarding the definition of the graph topology, are described in "Graph Topology (Edges and Weights)." In the following, we demonstrate how this simple and intuitive concept provides a natural and straightforward platform to introduce the graph counterparts of several fundamental signal processing algorithms.

System for graph signals

In classical signal processing, a system is an operator that transforms an input signal into another (output) signal. The purpose of this section is to provide a definition of *a system that operates over graph signals*. Our focus will be on a subclass of systems called *graph shift invariant systems*, also referred to by some authors as simply *graph filters*. As illustrated in the following pages, this class of systems accounts for the topology of the graph where the signals reside, while maintaining analytical and computational tractability.

Signal shift on a graph

The signal shift operator (unit time delay) is the linchpin in discrete-time signal processing, but its definition on graphs is not so obvious due to the rich underlying connectivity structure.

Graph Topology (Edges and Weights)

While in classical graph theory, the graphs are typically given (e.g., in various computer, social, road, transportation, and power networks), in graph signal processing oftentimes the first step is to employ background knowledge of signal-generating mechanisms to define the graph as a signal domain. This poses a number of challenges; e.g., while the data sensing points (graph vertices) are usually well defined in advance, their connectivity (graph edges) is often not available. In other words, the data domain definition within the graph signal paradigm represents a part of the problem itself and has to be determined based on the properties of the sensing positions or features of the acquired set of data. All in all, the definition of an appropriate graph structure is a prerequisite for physically meaningful and computationally efficient graph signal processing applications.

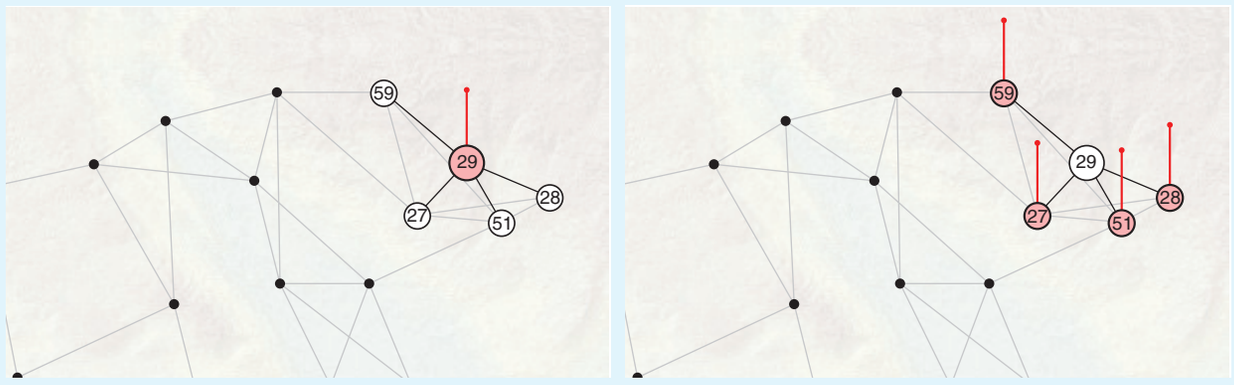
Three important classes of problems regarding the definition of graph edges are as follows:

- *Geometry of the vertex positions*: The distances between vertex positions play a crucial role in establishing relations among the sensed data. In many physical processes, both the existence of edges and their associated connecting weights are defined based on vertex distances. To this end, an exponential function of the Euclidean distance between vertices, r_{mn}, may be used, where for a given distance threshold, τ,

$$W_{mn} = e^{-r_{mn}^2/\alpha} \quad \text{or} \quad W_{mn} = e^{-r_{mn}/\alpha} \quad \text{if } r_{mn} < \tau,$$

and $W_{mn} = 0$ for $r_{mn} \geq \tau$. This form has been used in the graph in Figure 2, whereby the altitude difference, h_{mn}, was accounted for as $W_{mn} = e^{-r_{mn}/\alpha} e^{-h_{mn}/\beta}$.
- *Physically well-defined relations among the sensing positions*: Examples include electric circuits, linear heat transfer systems, spring-mass systems, and various forms of networks such as social, computer, or power networks. In these cases, the edge weights are given as a part of problem definition.
- *Data similarity dictates graph topology*: This scenario is the most common in image and biomedical signal processing (see "Graph Topology Based on Signal Similarity: An Image Processing Example"). Various approaches and metrics can be used to define data similarity, including the correlation matrix between the signals at various vertices or the corresponding inverse covariance (precision) matrix, combined with signal smoothness and edge sparsity conditions.

Learning a graph (its edges) based on the set of the available data is an interesting and currently extensively studied research area.



(a) (b)

FIGURE 4. A shift operator on a graph. (a) A single-pulse graph signal, $\mathbf{x}$, located at the vertex $n = 29$, and given by $x(n) = \delta(n - 29)$. (b) The graph shifted version of $\mathbf{x}$, given by $\mathbf{x}_{\text{shifted}} = \mathbf{A}\mathbf{x}$. The graph shift operator is demonstrated on the northeast part of the graph from Figure 3, around the vertex $n = 29$.

Topologically, the signal shift on a graph can be viewed as the movement of a signal sample from the considered vertex along all edges connected to this vertex. The (backward) shift operator on a graph can then be compactly defined using the graph adjacency matrix as $\mathbf{x}_{\text{shifted}} = \mathbf{A}\mathbf{x}$.

To draw distinction between the standard shift and the graph shift operator, consider the line graph in Figure 1(c) and the spatially aware graph in Figure 2, and assume that the input signal is a pulse that occurs only at the sensor $n = 29$, that is, $x(n) = \delta(n - 29)$. The shifted signal in classical signal processing [line graph in Figure 1(c)] will be $x_{\text{shifted}}(n) = \delta(n - 28)$ and can be considered as a movement of the delta pulse along the line graph from vertex n to vertex $(n - 1)$. The same principle can be applied to the

graph domain in Figure 2(a), whereby the delta pulse from vertex $n = 29$ is moved to all its connected vertices, to obtain the shifted graph signal, $x_{\text{shifted}}(n) = \delta(n - 27) + \delta(n - 28) + \delta(n - 51) + \delta(n - 59)$, as shown in Figure 4.

If the values of the shifted signal are also scaled by the weighting coefficients of the corresponding edges, then the shifted signal is given by $\mathbf{x}_{\text{shifted}} = \mathbf{W}\mathbf{x}$. Observe that the Laplacian operator applied on a signal, $\mathbf{L}\mathbf{x}$, can also be considered as a graph shift operator, since it is a combination of the scaled original signal, $\mathbf{D}\mathbf{x}$, and its weighted shifted version, $\mathbf{W}\mathbf{x}$, that is, $\mathbf{L}\mathbf{x} = \mathbf{D}\mathbf{x} - \mathbf{W}\mathbf{x}$. In the sequel, we will adopt the symbol $\mathbf{S}$ to denote a general shift operator on a graph, which yields a graph shifted signal $\mathbf{x}_{\text{shifted}} = \mathbf{S}\mathbf{x}$.

Remark 4

The standard shift operator, $x(n) = x(n-1)$, is a "one-to-one" mapping, while the graph shift operator, $\mathbf{x}_{\text{shifted}} = \mathbf{S}\mathbf{x}$, is a "one-to-many" mapping, which accounts for the underlying physics of the sensing process (as in our example), not possible to achieve with standard DSP. Moreover, by its very nature, the graph shift also allows us to incorporate a contextual relation between the vertices within an irregular grid through the weight matrix $\mathbf{W}$. Notice that the graph shift operator does not satisfy the isometry property, since the energy of the shifted signal is not the same as the energy of the original signal.

System for graph signals

In analogy with the pivotal role of time shift in standard system theory, a system for graph signals can be implemented as a linear combination of a graph signal and its graph shifted versions. Here, the notion of a system is used in its classical sense, as a set of physical rules (an algorithm) that transforms an input graph signal into another (output) graph signal. The output graph signal can then be written as

$$\mathbf{y} = h_0\mathbf{S}^0\mathbf{x} + h_1\mathbf{S}^1\mathbf{x} + \cdots + h_{M-1}\mathbf{S}^{M-1}\mathbf{x} = \sum_{m=0}^{M-1} h_m\mathbf{S}^m\mathbf{x}, \tag{8}$$

where, by definition $\mathbf{S}^0 = \mathbf{I}$, while h_0, h_1, …, h_{M-1} are the system coefficients (see the section "Spectral Domain Graph Filter Design").

Remark 5

Common choices for the graph shift operator are 1) adjacency matrix, $\mathbf{S} = \mathbf{A}$, and 2) graph Laplacian, $\mathbf{S} = \mathbf{L}$. Normalized versions of the adjacency matrix, graph Laplacian, $\mathbf{S} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$, and random walk (diffusion) matrix, $\mathbf{S} = \mathbf{D}^{-1}\mathbf{W}$, can also be used.

Notice that for the directed and unweighted line graph in Figure 1(c), the system for graph signals in (8) reduces to the well-known standard finite impulse response filter, given by

$$y(n) = h_0x(n) + h_1x(n-1) + \cdots + h_{M-1}x(n-M+1). \tag{9}$$

Remark 6

The previously established link between the classical transfer function of a physical system and its graph-theoretic counterpart may serve to promote new algorithmic approaches that stem from signal processing into many application scenarios that may be related to graphs.

Remark 7

A system defined based on the graph Laplacian, $\mathbf{L}$, is obtained from (8) by replacing $\mathbf{S} = \mathbf{L}$ and allows us to produce an unbiased estimate of a constant, c, whereby if $\mathbf{x} = \mathbf{c}$ then $\mathbf{y} = \mathbf{c}$, since by the definition of the graph Laplacian, $\mathbf{L}\mathbf{c} = \mathbf{0}$. Hence, a simple first-order system based on the graph Laplacian can be written as

$$\mathbf{y} = \mathbf{x} + h_1\mathbf{L}\mathbf{x} \tag{10}$$

and is amenable to being used for efficient low-pass graph filtering (see "Smoothness and Filtering on a Graph").

Remark 8

A system for graph signals is conveniently defined by the graph transfer function, $H(\mathbf{S})$, as

$$\mathbf{y} = H(\mathbf{S})\mathbf{x}. \tag{11}$$

Properties of a system for graph signals

Following the aforementioned discussion, it is now possible to define the properties of systems for graph signals. From (8)–(11), the system for graph signals is said to be as follows:

- Linear, if

$$H(\mathbf{S})(a_1\mathbf{x}_1 + a_2\mathbf{x}_2) = a_1\mathbf{y}_1 + a_2\mathbf{y}_2.$$

- Shift invariant, if

$$H(\mathbf{S})(\mathbf{S}\mathbf{x}) = \mathbf{S}(H(\mathbf{S})\mathbf{x}).$$

Remark 9

A system for graph signals, defined as in (8), in the form

$$H(\mathbf{S}) = h_0\mathbf{S}^0 + h_1\mathbf{S}^1 + \cdots + h_{s-1}\mathbf{S}^{M-1} \tag{12}$$

is linear and shift invariant, since the matrix multiplication of the square shift matrices is associative $(\mathbf{S}(\mathbf{S}\mathbf{S}) = (\mathbf{S}\mathbf{S})\mathbf{S})$, that is, $\mathbf{S}\mathbf{S}^m = \mathbf{S}^m\mathbf{S}$.

Graph Fourier transform

While classic spectral analysis is performed in the Fourier domain, spectral representations of graph signals employ the eigenspectrum (or simply "spectrum" hereafter) of the graph shift operator, $\mathbf{S}$, given by

$$\mathbf{S} = \mathbf{U}\Lambda\mathbf{U}^{-1},$$

where $\mathbf{U}$ is an orthonormal matrix of the eigenvectors, $\mathbf{u}_k$, in its columns, and Λ is a diagonal matrix of the corresponding eigenvalues, λ_k.

As in the majority of works in the literature dealing with the analysis of the frequency content of graph signals, here we will use $\mathbf{S} = \mathbf{L}$ in numerical examples, while $\mathbf{S} = \mathbf{A}$ is also used in illustrative examples, especially when relating graph signal processing to classical signal processing. The eigenvectors of graph Laplacian, $\mathbf{L}$, may then be used for the spectral-based clustering of graph vertices, as shown in "Vertex Clustering."

The graph Fourier transform (GFT), $\mathbf{X}$, of a graph signal, $\mathbf{x}$, is then defined as

$$\mathbf{X} = \mathbf{U}^{-1}\mathbf{x}. \tag{13}$$

The quadratic form of a graph signal is given by

$$E_x = \mathbf{x}^T\mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} W_{nm}(x(n)-x(m))^2$$

and can be used to define signal smoothness, since small values of the squared local deviation, $(x(n)-x(m))^2$, correspond to a smooth, slow-varying signal. For a constant signal, $\mathbf{x}=\mathbf{c}$, we therefore have $E_x=0$.

Physically, the minimum of $\mathbf{x}^T\mathbf{L}\mathbf{x}$ implies the smoothest possible signal, and to arrive at this solution we may employ steepest descent. Then, the signal value at an iteration p is adjusted in the opposite direction of the gradient, toward the minimum of $\mathbf{x}^T\mathbf{L}\mathbf{x}$. The gradient of this quadratic form is $\partial E_x/\partial\mathbf{x}^T = 2\mathbf{L}\mathbf{x}$, which yields the iterative procedure

$$\mathbf{x}_{p+1} = \mathbf{x}_p - \alpha\mathbf{L}\mathbf{x}_p = (\mathbf{I}-\alpha\mathbf{L})\mathbf{x}_p.$$

Notice that the signal $\mathbf{x}_{p+1}$ can be considered as an output of the first-order system in (11), with $h_1=-\alpha$, and this relation can be used for simple and efficient filtering of graph signals.

Since the minimum of the quadratic form $\mathbf{x}^T\mathbf{L}\mathbf{x}$ corresponds to a constant signal, to avoid obtaining only a constant steady state (that is, to also account for the slow-varying part of the graph signal), the aforementioned iteration process can be used in alternation with $\mathbf{x}_{p+2}=(\mathbf{I}+\beta\mathbf{L})\mathbf{x}_{p+1}$. A compact form of these two iterative processes is known as Taubin's $\alpha-\beta$ algorithm and is given by

$$\mathbf{x}_{p+2} = (\mathbf{I}+\beta\mathbf{L})(\mathbf{I}-\alpha\mathbf{L})\mathbf{x}_p. \tag{S1}$$

For appropriate values of α and β, this system can give a good and very simple approximation of a low-pass graph filter with transfer function $H(\lambda_k)=(1+(\beta-\alpha)\lambda_k-\alpha\beta\lambda_k^2)^P$, and in P iterations, where k denotes the spectral index (see the section "Spectral Domain Graph Filter Design").

In our experiment, the original noisy signal from Figure 3 was filtered using Taubin's algorithm, with $\alpha=0.2$ and $\beta=0.1$. After 50 iterations, the signal-to-noise ratio improved from the original $\mathrm{SNR}_0=14.2$ dB to 26.8 dB (see Figure S1). **<AU: Kindly check whether the citation of Figure S1 is appropriate.>** With these parameters, the transfer function, $H(\lambda_k)$, retained seven out of 64 spectral components in the signal (with an attenuation lower than 3 dB).

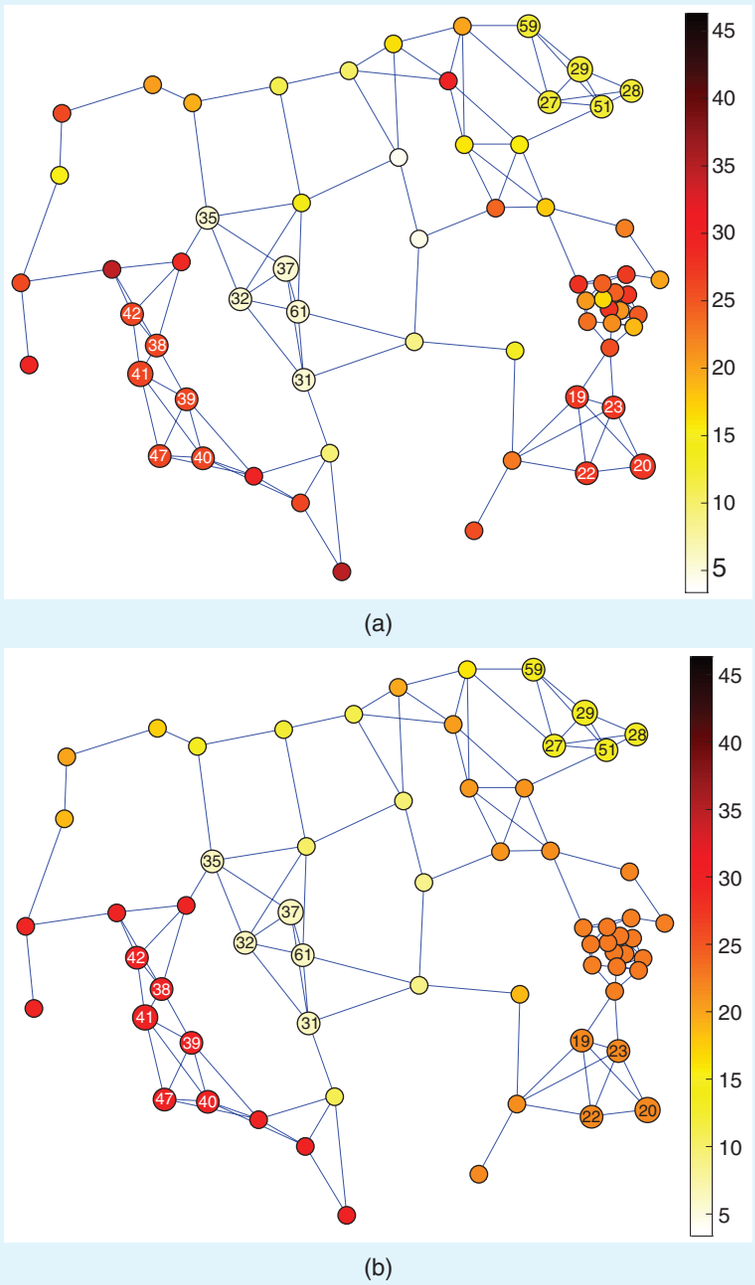


FIGURE S1. An illustration of low-pass filtering on a graph. (a) The original noisy signal. (b) The filtered signal. The graph signal intensity is designated by the vertex color.

Vertex Clustering

The term *vertex clustering* here refers to the task of identifying and arranging the vertices of a graph into nonoverlapping vertex subsets, with data in each subset expected to exhibit relative similarity in some sense. One efficient approach to vertex clustering is based on spectral graph analysis. For a graph with N vertices, the orthogonal eigenvectors of its Laplacian build an N-dimensional space, called the *spectral space*. The elements $u_k(n)$ of the eigenvector $\mathbf{u}_k$, $k = 1, 2, \ldots, N$, can then be assigned to vertices n, $n = 1, 2, \ldots, N$ to form an N-dimensional spectral vector $\mathbf{q}_n = [u_1(n), u_2(n), \ldots, u_N(n)]$. The elements of the first eigenvector, $\mathbf{u}_1$, of the graph Laplacian are constant and are omitted, since they do not convey any spectral difference to the graph vertices.

For the purpose of vertex clustering, the original N-dimensional spectral vector space may be reduced to a new L-dimensional spectral space ($L < N$), where the spectral vectors,

$$\mathbf{q}_n = [u_2(n), u_3(n), \ldots, u_{L+1}(n)],$$

are used to define the spectral similarity between any two vertices, n and m, as $\|\mathbf{q}_n - \mathbf{q}_m\|_2$. Vertex clustering is then performed by grouping spectrally similar vertices.

The simplest (and most widely used) case is when only one eigenvector, $\mathbf{u}_2$, is used for spectral clustering, whereby the order of vertices in the sorted $\mathbf{u}_2$ corresponds to its smoothest representation. This procedure can be used for ordering the vertices in graphs, even if we desire to perform any form of classical presentation or processing with vertices on a line graph, as in Figure 1(c).

The spectral vector, $\mathbf{q}_n$, either can be used to designate a position of a vertex in a new low L-dimensional space, or it can be used for coloring of the vertices at their original positions. For the graph from Figure 2, such coloring was performed using the spectral vector elements $\mathbf{q}_n = [u_2(n), u_3(n), u_4(n)]$ as color coordinates for the vertex n (see Figure S2). Similar colors indicate high spectral similarity. ***<AU: Kindly check whether the citation of Figure S2 is appropriate.>***

Note that vertex clustering is a signal-independent operation. It just roughly indicates the expected relation between sensor data values on the considered graph and suggests that data processing operations (including processing of the signal from Figure 3) will be predominantly localized within these clusters.

Formally, the so-achieved reduction in spectral vertex dimensionality, from the original N eigenvectors to L eigenvectors with lowest variations (with the smallest smoothness index $\mathbf{u}_k^T \mathbf{L} \mathbf{u}_k = \lambda_k$), corresponds to low-pass filtering in graph signal processing, whereby a signal with N spectral components is projected onto a reduced spectral space with L slowest-varying spectral components, within a given set of basis functions (cf. truncated Fourier representation).

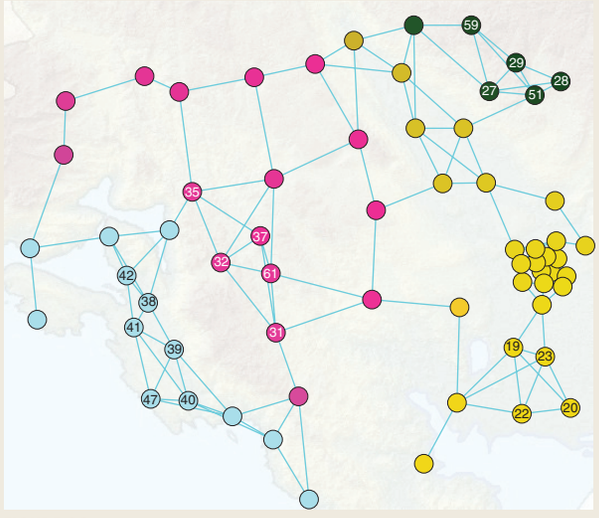


FIGURE S2. The vertices in the graph from Figure 2 have been colored using the spectral vectors $\mathbf{q}_n = [u_2(n), u_3(n), u_4(n)]$ as red, green, blue color coordinates.

The *GFT domain* is referred to as the *graph spectral domain*, since the domain for the GFT, $\mathbf{X}$ (with elements commonly denoted by $X(k)$ or $X(\lambda k)$), is the graph spectrum, $\lambda_k, k = 1, 2, \ldots, N$.

Physically, since $\mathbf{U}^{-1} = \mathbf{U}^T$, the element $X(k)$ of a GFT, $\mathbf{X}$, represents a projection of the graph signal, $\mathbf{x}$, onto the kth eigenvector, $\mathbf{u}_k \in \mathbf{U}$, that is

$$X(k) = \sum_{n=1}^{N} x(n) u_k(n). \tag{14}$$

The inverse GFT is then straightforwardly obtained as

$$\mathbf{x} = \mathbf{U}\mathbf{X} \tag{15}$$

or

$$x(n) = \sum_{k=1}^{N} X(k) u_k(n). \tag{16}$$

Remark 10

In analogy to the classical Fourier transform where the signal is projected onto a set of harmonic orthogonal bases, $\mathbf{X} = \mathbf{U}^{-1}\mathbf{x}$, where $\mathbf{U}$ is the matrix of harmonic bases, $\mathbf{u}_k = [1, e^{j2\pi k/N}, \ldots, e^{j2\pi(N-1)k/N}]^T/\sqrt{N}$, the GFT can be understood as a signal decomposition onto the set of eigenvectors of the graph Laplacian (or the adjacency matrix) that serve as orthonormal basis functions. In the case of a circular graph, the GFT reduces to the standard discrete Fourier transform (DFT). For this reason, the transform in (14) is referred to as the GFT.

Classic spectral analysis can thus be considered as a special case of graph sig-

nal spectral analysis, with the adjacency matrix defined on an unweighted circular directed graph (a line graph with the connected last and first vertex), where $\mathbf{u}_k = [1, e^{j2\pi k/N}, \ldots, e^{j2\pi(N-1)k/N}]^T/\sqrt{N}$. This becomes obvious upon recognizing that the eigenvalues of a directed unweighted circular graph, $\lambda_k = e^{j2\pi k/N}$, are easily obtained as a solution of the eigenvalue/eigenvector relation $\mathbf{A}\mathbf{u}_k = \lambda_k \mathbf{u}_k$. For a vertex n, this relation is of the form $u_k(n-1) = \lambda_k u_k(n)$. The solutions of this difference equation are the elements of the previously discussed eigenvector, $u_k(n) = e^{j2\pi nk/N}$, and the corresponding eigenvalues, $\lambda_k = e^{-j2\pi k/N}$. It can be shown that the eigenvectors of the graph Laplacian of a circular graph are real-valued harmonic functions, whose combinations can produce the standard complex-valued DFT basis functions, albeit in an indirect way. The standard signal representation in Figure 1(b), therefore, corresponds to a signal whose domain is a line graph.

As is common in signal processing, for our example in Figure 1 and (1) the temperature values were generated through a linear combination of several graph Laplacian eigenvectors (serving as basis functions) in the form $\mathbf{x} = 160\mathbf{u}_1 + 16\mathbf{u}_2 - 8\mathbf{u}_3 - 40\mathbf{u}_4 + 16\mathbf{u}_5 - 24\mathbf{u}_6 + \varepsilon(n)$, where the random Gaussian noise, $\varepsilon(n)$, had standard deviation $\sigma_\varepsilon = 4$, to yield the signal-to-noise ratio of $\text{SNR}_0 = 14.2$ dB.

Spectral domain of a system for graph signals

Consider a system for graph signals, as in (8), defined by a general shift operator on a graph $\mathbf{S}$, given by

$$\mathbf{y} = \sum_{m=0}^{M-1} h_m \mathbf{S}^m \mathbf{x}. \tag{17}$$

Upon employing the eigendecomposition of the shift operator, $\mathbf{S} = \mathbf{U}\Lambda\mathbf{U}^{-1}$, we arrive at

$$\mathbf{y} = \sum_{m=0}^{M-1} h_m \mathbf{U}\Lambda^m\mathbf{U}^{-1}\mathbf{x} = \mathbf{U}H(\Lambda)\mathbf{U}^{-1}\mathbf{x}, \tag{18}$$

where

$$H(\Lambda) = \sum_{m=0}^{M-1} h_m \Lambda^m \tag{19}$$

is the transfer function of the system for graph signals.

From (18), $\mathbf{U}^{-1}\mathbf{y} = H(\Lambda)\mathbf{U}^{-1}\mathbf{x}$, or in terms of the GFT of the input and output signal

$$\mathbf{Y} = H(\Lambda)\mathbf{X}. \tag{20}$$

The classic spectral transfer function for (9) is then obtained by using the adjacency matrix of an unweighted directed circular graph, whose eigenvalues are $\lambda_k = e^{-j2\pi k/N}$.

Spectral domain graph filter design

A system for graph signals that is designed to modify spectral content of graph signals in a desired way shall be referred to as *a graph filter*. Consider a graph filter with a desired transfer function, $G(\Lambda)$. As in classical signal processing, a filter with this transfer function can be implemented either in the spectral domain or in the vertex domain.

The spectral domain implementation is straightforward and can be performed in the following three steps:

- Calculate the GFT of the input graph signal, $\mathbf{x}$, in the form $\mathbf{X} = \mathbf{U}^{-1}\mathbf{x}$.
- Multiply the GFT of $\mathbf{x}$ with the transfer function, $G(\Lambda)$, to obtain $\mathbf{Y} = G(\Lambda)\mathbf{X}$.
- Calculate the output graph signal as the inverse GFT of $\mathbf{Y}$, to yield $\mathbf{y} = \mathbf{U}\mathbf{Y}$.

Notice that this procedure may be computationally very demanding for large graphs, where it may be easier to implement the desired filter (or its close approximation) in the vertex domain, in analogy to the time domain in the classical approach. In other words, we have to find the coefficients, $h_0, h_1, \ldots, h_{M-1}$ in (8), such that their spectral representation, $H(\Lambda)$, is equal (or at least as close as possible) to the desired graph filter $G(\Lambda)$.

The previously mentioned condition that the transfer function of the vertex domain system for graph signals in (19), given by $H(\lambda_k) = h_0 + h_1\lambda_k^1 + \cdots + h_{M-1}\lambda_k^{M-1}$, should be equal to the desired transfer function, $G(\lambda_k)$, for each spectral index, k, leads to a system of linear equations

$$\begin{aligned} h_0 + h_1\lambda_1^1 + \cdots + h_{M-1}\lambda_1^{M-1} &= G(\lambda_1) \\ h_0 + h_1\lambda_N^1 + \cdots + h_{M-1}\lambda_N^{M-1} &= G(\lambda_N) \\ &\vdots \\ h_0 + h_1\lambda_2^1 + \cdots + h_{M-1}\lambda_2^{M-1} &= G(\lambda_2) \end{aligned} \tag{21}$$

of which the matrix form is given by

$$\mathbf{V}_\lambda \mathbf{h} = \mathbf{g}, \tag{22}$$

where $\mathbf{V}_\lambda$ is a Vandermonde matrix formed of the eigenvalues, λ_k, while $\mathbf{h} = [h_0, h_1, \ldots, h_{M-1}]^T$ is the vector of system coefficients that we wish to estimate, and

$$\begin{aligned} \mathbf{g} &= [G(\lambda_1), G(\lambda_2), \ldots, G(\lambda_N)]^T \\ &= \text{diag}(G(\Lambda)). \end{aligned}$$

Comments on the Graph Filter in (22)

Consider the following cases.

1) All the eigenvalues of $\mathbf{S}$ are distinct:
   a) For $M = N$, the solution is unique.
   b) For $M < N$ (overdetermined system), the least squares solution is obtained.
2) Some of the eigenvalues are of a degree higher than one, the system reduces to $N_m < N$ linear equations.
   a) For $N_m < M \le N$ (underdetermined system), $(M - N_m)$ filter coefficients are free variables, and an infinite number of *equivalent filters* is obtained.
   b) For $M = N_m$, the solution is unique.
   c) For $M < N_m$ (overdetermined system), the least squares solution is obtained.
3) Any filter of an order $M > N_m$ has a *unique equivalent filter* whose order is at most N_m. Such equivalence can be obtained by setting the free variables to zero, $h_i = 0$ for $i = N_m, N_m + 1, \ldots, N - 1$.

Graph Topology Based on Signal Similarity: An Image Processing Example

The graph weights in our temperature field example are defined based on the geometric distance of vertices (sensing points). However, in some applications signal values themselves may be used as an indicator of signal similarity, as is the case with image processing, where this is achieved in combination with the pixel/vertex distances. For the image intensity values at pixels indexed by n and m, denoted by $x(n)$ and $x(m)$, a simple difference of intensities

$$\text{intensity distance}(n, m) = s_{nm} = |x(n) - x(m)|,$$

may be used in *an exponential kernel* to define the corresponding edge weights as

$$W_{nm} = e^{-(x(n)-x(m))^2/\tau^2} \text{ for } r_{nm} \le \kappa,$$

and $W_{nm} = 0$ for $r_{nm} \ge \kappa$, where r_{nm} is a geometric distance of the considered pixels/vertices.

We next present an example of this kind of edge weighting applied to a simple graph image filtering problem.

Example

Consider the problem of denoising a 50-pixel-square, 8-bit grayscale, image (see Figure S3). **<AU: Kindly check whether the citation of Figure S3 is appropriate.>** The vertices of the graph are the pixels. The edge weights for the graph representation of this noisy image were calculated with $\kappa = \sqrt{2}$ and $\tau = 20$. This value of κ means that each vertex is connected with eight neighboring vertices (including diagonal ones) with the weights, W_{nm}, defined by the exponential kernel. Low-pass filtering was performed on the corresponding image graph using iterative filtering (Taubin's algorithm) over 200 iterations, with $\alpha = 0.1$ and $\beta = 0.15$.

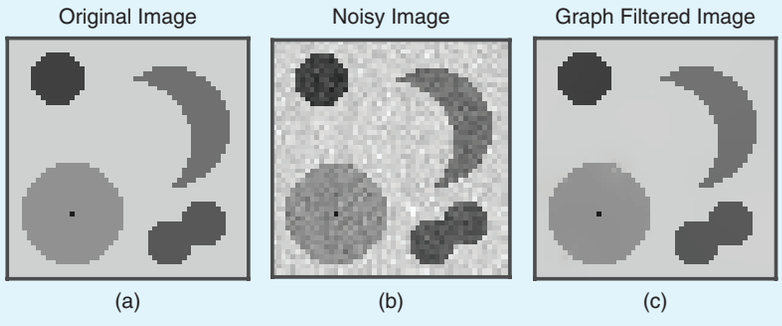


FIGURE S3. (a) The original image, (b) the noise-corrupted image, and (c) the image filtered using Taubin's algorithm (see "Smoothness and Filtering on a Graph").

In practice, the system order M, or equivalently the order of the graph filter, is typically significantly lower than the number of equations, N, in (21). For such an overdetermined case, the least-squares approximation of $\mathbf{h}$ is obtained by minimizing the squared error, $e^2 = \|\mathbf{V}_\lambda \mathbf{h} - \mathbf{g}\|_2^2$. As in standard least-squares, the solution is obtained by a direct minimization, $\partial e^2 / \partial \mathbf{h}^T = \mathbf{0}$, to yield the coefficient estimates, $\hat{\mathbf{h}}$, in the form

$$\hat{\mathbf{h}} = (\mathbf{V}_\lambda^T \mathbf{V}_\lambda)^{-1} \mathbf{V}_\lambda^T \mathbf{g} = \text{pinv}(\mathbf{V}_\lambda)\mathbf{g}. \quad (23)$$

The so obtained solution, $\hat{\mathbf{h}}$, therefore represents the mean-square error minimizer for $\mathbf{V}_\lambda \mathbf{h} = \mathbf{g}$. Notice that this solution may not satisfy $\mathbf{V}_\lambda \mathbf{h} = \mathbf{g}$, in which case the coefficients $\hat{\mathbf{g}}$ (its spectrum $\hat{G}(\Lambda)$) may be used, that is

$$\mathbf{V}_\lambda \hat{\mathbf{h}} = \hat{\mathbf{g}}.$$

Such a solution, in general, differs from the desired system coefficients $\mathbf{g}$ (its spectrum $G(\Lambda)$).

Example

Consider the graph signal from Figure 3 and the graph Laplacian employed as the shift operator. The task is to design a graph filter whose frequency response is $g(\lambda_k) = \exp(-\lambda_k)$, to filter the graph signal using this spectral domain graph filter. For $M = 4$, the corresponding system coefficients can be found to be $h_0 = 0.9606$, $h_1 = -0.7453$, $h_2 = 0.1936$, and $h_3 = -0.0162$. Upon filtering the graph signal using the so-defined graph transfer function, the output signal-to-noise ratio was $\text{SNR} = 21.74$ dB, that is, a 7.54 dB improvement over the original signal-to-noise ratio of $\text{SNR}_0 = 14.2$ dB. More detail on the solution of the system in (21) and (22) is provided in "Comments on the Graph Filter in (22)."

Optimal denoising

Consider a measurement, as in the temperature measurement scenario in Figure 1, that is composed of a slow-varying desired signal, $\mathbf{s}$, and a superimposed fast-changing disturbance, ε, to give

$$\mathbf{x} = \mathbf{s} + \varepsilon.$$

The aim is to design a graph filter for disturbance suppression (denoising), the output of which is denoted by $\mathbf{y}$ [11]. The optimal denoising task can then be defined through a minimization of the cost function

$$J(\mathbf{y}|\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 + \alpha \mathbf{y}^T \mathbf{L} \mathbf{y}, \quad (24)$$

where the minimization of the first term, $1/2\|\mathbf{y} - \mathbf{x}\|_2^2$, enforces the output signal, $\mathbf{y}$, to be as close as possible, in terms of the minimum residual disturbance power, to the available observations, $\mathbf{x}$. As shown in "Smoothness

and Filtering on a Graph," the second term, $\mathbf{y}^T\mathbf{L}\mathbf{y}$, represents a measure of smoothness of the graph filter output, $\mathbf{y}$, while the parameter α models a balance between the closeness of the output, $\mathbf{y}$, to the observed data, $\mathbf{x}$, and the smoothness-constrained output estimate $\mathbf{y}$. While the problem in (24) could also be expressed through a constrained Lagrangian optimization, we here focus more on the graph-theoretic issues and hence adopt a simpler option whereby the mixing parameter α is chosen empirically.

The solution to the minimization problem in (24) follows from

$$\frac{\partial J(\mathbf{y}|\mathbf{x})}{\partial \mathbf{y}^T} = \mathbf{y} - \mathbf{x} + 2\alpha\mathbf{L}\mathbf{y} = \mathbf{0}$$

and results in a smoothing optimal denoiser in the form

$$\mathbf{y} = (\mathbf{I} + 2\alpha\mathbf{L})^{-1}\mathbf{x}.$$

The Laplacian spectral domain form of this relation then becomes

$$\mathbf{Y} = (\mathbf{I} + 2\alpha\Lambda)^{-1}\mathbf{X},$$

with the corresponding graph filter transfer function given by

$$H(\lambda_k) = \frac{1}{1 + 2\alpha\lambda_k}.$$

Observe that for a small α, $H(\lambda_k) \approx 1$ and $\mathbf{y} \approx \mathbf{x}$, while for a large α, $H(\lambda_k) \approx \delta(k)$ and $\mathbf{y} \approx const.$, which enforces $\mathbf{y}$ to be maximally smooth (a constant, without any variation). Using $\alpha = 4$, the obtained output signal-to-noise ratio for the graph signal from Figure 3 was SNR = 26 dB, an 11.8 dB improvement over the original $\text{SNR}_0 = 14.2$ dB.

Remark 11

There are many cases when the graph topology is unknown, so that the graph structure, that is, the graph edges and their weights, is also unknown. To this end, we may employ a class of methods for graph topology learning, which are based on the minimization of the cost function in (24) with respect to both a chosen graph connectivity matrix and the output signal, $\mathbf{y}$, with additional (commonly sparsity) constraints imposed on the elements of the considered connectivity matrix.

What we have learned

Natural signals (speech, biomedical, video) often reside over irregular domains and are, unlike the standard signals in, for example, communications, not adequately processed using standard harmonic analyses. While data analytics on graphs as irregular domains are heavily dependent on advances in DSP, neither the electrical and electronics engineering **<AU: Kindly check that "EEE" is spelled out correctly.>** graduates worldwide nor practical data analysts are yet best prepared to employ graph algorithms in their future jobs. Our aim has been to fill this void by providing an example-driven platform to introduce graphs, signals on graphs, and their properties through a well-understood multisensor measurement scenario and the graph notions of transfer function, Fourier transform, and digital filtering.

We have illuminated that while both a graph with N vertices and a classical discrete time signal with N samples can be viewed as N-dimensional vectors, structured graphs represent much richer irregular domains that convey information about both the signal and its generation and propagation mechanisms. This allows us to employ intuition and physical know-how from Euclidean domains to revisit basic dimensionality-reduction operations, such as coarse graining of graphs (cf. standard downsampling). In addition, in the vertex domain a number of different distances (shortest-path, resistance, diffusion) have useful properties that can be employed to maintain data integrity throughout the processing, storage, communication, and analysis stages, as the vertex connectivities and edge weights are either dictated by the physics of the problem at hand or are inferred from the data. This particularly facilitates maintaining control and intuition over distributed operations throughout the processing chain.

It is our hope that this lecture note has helped to demystify graph signal processing for students and educators, together with empowering practitioners with enhanced intuition in graph-theoretic design and optimization. This material may also serve as a vehicle to seamlessly merge curricula in electrical engineering and computing. The generic and physically meaningful nature of this example-driven lecture note is also likely to promote intellectual curiosity and serve as a platform to explore the numerous opportunities in manifold applications in our ever-growing interconnected world, facilitated by the Internet of Things.

Acknowledgments

We give our sincere thanks to the anonymous reviewers for their constructive and insightful comments. We are privileged to have had the help and advice of one of the pioneers in graph theory, Prof. Nicos Christofides. We are grateful for his time, his incisive comments, and valuable advice. The help from Bruno Scalzo Dees and Shengxi Li from Imperial College London and Miloš Brajović from the University of Montenegro is also greatly appreciated. Last but not least, we would also like to express our sincere gratitude to the students in our respective postgraduate courses, for their feedback on the material taught based on this lecture note.

Authors

Ljubiša Stanković (ljubisa@ucg.ac.me) is a professor at the University of Montenegro. His research interests include time–frequency analysis, compressive sensing, and graph signal processing. He is a vice president of the National Academy of Sciences and Arts of Montenegro and is a recipient of the 2017 European Association for Signal Processing Best Journal Paper Award. He is a Fellow of the IEEE. **<AU: Please provide undergraduate and graduate information.>**

Danilo P. Mandic (d.mandic@imperial.ac.uk) is a professor of signal processing at Imperial College London, United Kingdom. He has a keen interest in signal processing education, is a mem-

ber of the IEEE Signal Processing Society Education Technical Committee, and his contributions were recognized with the President's Award for Excellence in Postgraduate Supervision at Imperial College in 2014. He is a Fellow of the IEEE.

Miloš Daković (milos@ucg.ac.me) is professor at the University of Montenegro. His research interests include graph signal processing, compressive sensing, and time–frequency analysis. He is a Member of the IEEE. **<AU: Please provide undergraduate and graduate information.>**

Ilya Kisil (i.kisil15@imperial.ac.uk) is a Ph.D. candidate at Imperial College London, United Kingdom. His research interests include tensor decompositions, big data, efficient software for large-scale problems, and graph signal processing. **<AU: Please provide undergraduate information.>**

Ervin Sejdić (esejdic@ieee.org) is an ~~assistant~~ professor at the University of Pittsburgh, Pennsylvania. His research interests include biomedical signal processing, rehabilitation engineering, and neuroscience. He received the U.S. Presidential Early Career Award for Scientists and Engineers in 2016. He is a Senior Member of the IEEE. **<AU: Please provide undergraduate and graduate information.>**

Anthony G. Constantinides (a.constantinides@imperial.ac.uk) is emeritus professor of signal processing in the Department of Electrical and Electronic Engineering at Imperial College London, United Kingdom. He is a Life Fellow of the IEEE. **<AU: Please provide undergraduate and graduate information.>**

References

[1] N. Christofides, *Graph theory: An algorithmic approach*. New York: Academic, 1975. **<AU: Please confirm publisher location.>**

[2] F. Afrati and A. G. Constantinides, "The use of graph theory in binary block code construction," in *Proc. Int. Conf. on Digital Signal Processing*, Florence, Italy, 1978, pp. 228–233.

[3] O. J. Morris, M. de J. Lee, and A. G. Constantinides, "Graph theory for image analysis: An approach based on the shortest spanning tree," *IEE Proc. F Commun., Radar Signal Processing*, vol. 133, no. 2, pp. 146–152, 1986. doi: 10.1049/ip-f-1.1986.0025.

[4] S. Chen, R. 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. doi: 10.1109/TSP.2015.2469645.

[5] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013. doi: 10.1109/TSP.2013.2238935.

[6] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, June 2014. doi: 10.1109/TSP.2014.2321121.

[7] G. Cheung, E. Magli, Y. Tanaka, and M. K. Ng, "Graph spectral image processing," *Proc. IEEE*, vol. 106, no. 5, pp. 907–930, May 2018. doi: 10.1109/JPROC.2018.2799702.

[8] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018. doi: 10.1109/JPROC.2018.2820126.

[9] S. Saito, H. Suzuki, and D. P. Mandic, "Hypergraph p-Laplacian: A differential geometry view," in *Proc. 32nd AAAI Conf. on Artificial Intelligence (AAAI-18)*, 2018, pp. 3984–3991.

[10] L. Stanković and E. Sejdić, *Vertex-Frequency Analysis of Graph Signals*. Cham, Switzerland: Springer Nature, 2019.

[11] S. Segarra, A. G. Marques, and A. Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 4117–4131, 2017. doi: 10.1109/TSP.2017.2703660.

SP

Graphs are irregular structures that naturally account for data integrity.

This article was primarily written in response to the urgent need to bring graph signal processing into the curricula of existing statistical signal processing and machine learning courses.

The advantage of graphs over classical data domains is that graphs account naturally and comprehensively for irregular data relations in the problem definition.

We show that graph signal and information processing is particularly well suited to making sense from data acquired over irregular data domains.

The beginning of graph theory applications in electrical engineering dates back to the mid-19th century and the definition of Kirchhoff's laws.

Owing to their inherent "spatial awareness," graph models have since become a de facto standard for data analytics across the science and engineering areas.

When the relations between all pairs of vertices are mutually symmetric, then all the matrices involved are also symmetric, and such graphs are called *undirected*.

In analogy with the pivotal role of time shift in standard system theory, a system for graph signals can be implemented as a linear combination of a graph signal and its graph shifted versions.

Structured graphs represent much richer irregular domains that convey information about both the signal and its generation and propagation mechanisms.