

Vertex-Frequency Analysis: A Way to Localize Graph Spectral Components

Ljubiša Stanković, Miloš Daković, Ervin Sejdić

I. SCOPE

CURRENTLY, brain and social networks are examples of new data types that are massively acquired and disseminated [1]. These networks typically consist of vertices (nodes) and edges (connections between nodes). Typically, information is conveyed through the strength of connection among nodes, but in recent years it has been discovered that valuable information may also be conveyed in signals that occur on each vertex. However, traditional signal processing often does not offer reliable tools and algorithms to analyze such new data types. This is especially true for cases where networks (e.g., the strength of connections), or signals on vertices have properties that change over the network.

These lecture notes present a new method to analyze changes in signals on graphs. This method, called the vertex-frequency analysis, relies on Laplacian matrices to establish connections between vertex changes and spectral content [2], [3], [4], [5]. Specifically, these lecture notes aim to connect concepts from frequency and time-frequency analyses (e.g., [6], [7]) to the spectral analysis of graph signals. Graph signal processing is a major research area, however, we still lack understanding of how to relate graph signal processing concepts to concepts from traditional signal processing.

II. RELEVANCE

The vertex-frequency analysis presented here is a valuable tool that can be used to analyze vertex-varying changes in networks (graphs) such as brain networks (e.g., brain changes during consecutive swallows [8]), changes in social interactions in a large group of people, or to understand traffic patterns during rush-hour in major metropolitan areas. Theoretically, it connects principles of the Fourier analysis and eigenvalue decomposition from undergraduate courses, to more advanced topics such as time-frequency representations typically taught at a graduate level.

III. PREREQUISITES

The prerequisites for understanding these lecture notes are linear algebra and an understanding of basic signal processing concepts.

IV. PROBLEM STATEMENT AND SOLUTION

A. Problem Statement

A graph consists of vertices and edges. If we denote the weights of graph edges connecting vertices n and m as w_{nm} , then the graph Laplacian operator is defined by

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

where the matrix \mathbf{W} elements are weighting coefficients w_{nm} and \mathbf{D} is a diagonal matrix with elements $d_n = \sum_{m=1}^N w_{nm}$. An example of such a graph is shown in Fig.1(a).

Consider a signal \mathbf{x} whose samples are $x(n)$, as shown in Fig.1(c), and these samples are assigned to (sensed at) the graph vertices as shown in Fig.1(b). The Laplacian operator applied to a signals on the graph is equal to $\mathbf{L}\mathbf{x}$, with elements $\mathcal{L}_x(n) = \sum_m w_{nm}(x(n) - x(m))$.

The spectral representation of a discrete-time signal $x(n)$ on the graph is defined as its expansion onto the set of eigenvectors (discrete-time basis eigenfunctions) of the Laplacian. In order to accomplish this expansion, the Laplacian \mathbf{L} is decomposed as

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T,$$

where \mathbf{U} is a matrix of the Laplacian eigenvectors \mathbf{u}_k and $\mathbf{\Lambda}$ is a diagonal matrix of its eigenvalues λ_k .

The spectrum $X(\lambda_k)$ of signal \mathbf{x} on a graph is calculated as its projection onto the corresponding eigenvectors \mathbf{u}_k of the Laplacian:

$$X(\lambda_k) = \mathbf{u}_k^T \mathbf{x},$$

or in vector notation $\mathbf{X} = \mathbf{U}^T \mathbf{x}$.

Since the eigenvectors are orthogonal, the signal reconstruction is defined as $\mathbf{x} = \sum_{k=1}^N X(\lambda_k) \mathbf{u}_k = \mathbf{U}\mathbf{X}$.

Spectral decomposition of a graph signal is illustrated in Fig. 1(d). This spectrum contains three components corresponding to the constant component at $\lambda_1 = 0$, a low-frequency component at $\lambda_2 = 0.6934$, and a high-frequency component at $\lambda_6 = 2.4644$. We can split the signal into, for example, its low-frequency part by summing over $k = 1, 2$ and the high-frequency part by using $k = 6$ as $X(\lambda_6) \mathbf{u}_6$.

The meaning of weighting coefficients w_{nm} in a graph is highly dependent on the application, especially as the graph Laplacian is defined by these coefficients, and the Laplacian operator then defines the set of basis functions for signal expansion. For example, classical Fourier analysis can be obtained by considering the second-order derivative estimation (Laplace operator), Box 1. The Laplacian operator is also known as the Kirchhoff matrix in electrical circuit theory, Box 2. In image processing, the coefficients w_{nm} may be proportional to the similarities of adjacent image pixels. Similarly, graphs are widely used in neuroscience, and edge coefficients are used to describe the strength of interactions among brain regions. In the case of a graph signal corresponding to a Euclidian network, the coefficient values are related to the vertex distances. A common way to define the coefficients in such networks is $w_{nm} = \exp(-r_{nm}/\tau)$ for $w_{nm} > \kappa$ and $w_{nm} = 0$ elsewhere, where r_{nm} is the

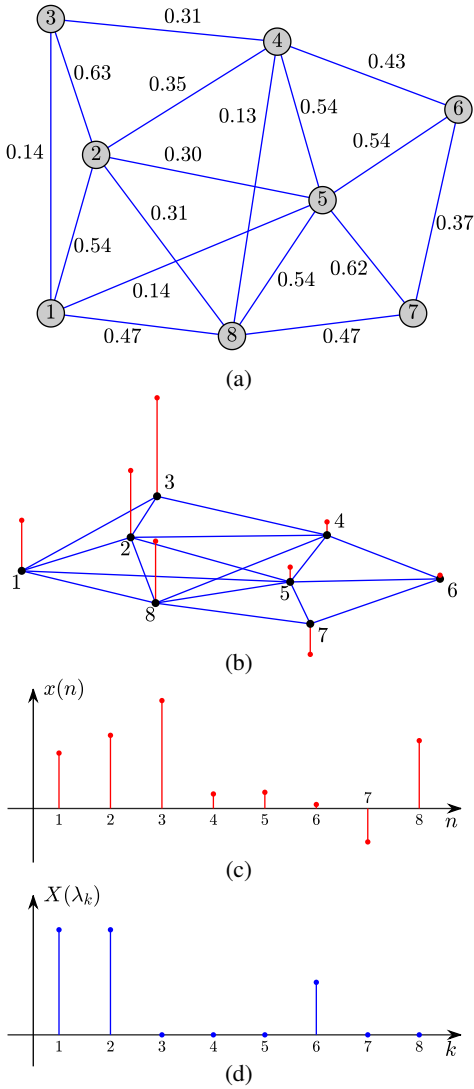


Fig. 1: Sample graphs: (a) a graph with weighted edges; (b) a sample graph with signal values on each vertex; (c) a signal, $x(n)$; (d) the spectrum of $x(n)$.

Euclidian distance between vertices n and m , and τ and κ are constants. This approach is used in Fig. 1(a).

The presented spectral analysis of signals on graphs provides a way to process signals in the graph spectral domain, that is, to implement signal processing techniques such as filtering, denoising, or to reconstruct missing signal values at some vertices if the graph signal spectrum is sparse.

Similar to Fourier domain analysis in traditional signal processing, the considered spectral analysis of signals on graphs has its limitations. For example, let us consider the graph shown in Fig. 2(a) and two signals on this graph, presented in Figs. 2(b) and 2(c). While these two signals $x_1(n)$ and $x_2(n)$ are obviously different, their spectral representations on this graph $|X_1(\lambda_k)|$ and $|X_2(\lambda_k)|$ are almost the same as depicted in Figs. 2(d) and 2(e). Hence, it would be very difficult to implement any machine learning schema that would be able to differentiate these two cases in the spectral domain. Therefore, an analysis is needed, similar to the time-frequency analysis in

traditional signal processing, that is able to provide localized vertex information about the spectrum.

B. Solution

Localizing a spectral content around each vertex n can be achieved via vertex-frequency analysis. This analysis is an extension of the traditional time-frequency analysis to graph signals. As in the classical time-frequency analysis, a spectral transformation of a signal localized around the considered vertex n yields the basic formulation of the vertex-frequency analysis. This spectral transformation is typically achieved using a localization window. While different approaches exist, we will present two of them, one based on shifting a window in the vertex-frequency domain, and the other based on a vertex neighborhood analysis.

1) *Convolution-based Definition:* In order to define a localized spectrum, let us consider two signals $x(n)$ and $h(n)$ on a graph with the corresponding Laplacian \mathbf{L} , whose eigenvalues and eigenvectors are λ_k and $u_k(n)$, respectively, while signal spectra are given by $X(\lambda_k)$ and $H(\lambda_k)$. Here, the signal $h(n)$ is used to localize the spectral characteristics of $x(n)$. For these two graph signals Parseval's theorem is given by $\sum_{n=1}^N x(n)h(n) = \sum_{k=1}^N X(\lambda_k)H(\lambda_k)$.

The shift of a signal on a graph cannot be extended in a direct way from the traditional signal processing theory. Hence, a generalized convolution operator on graphs is defined under the assumption that the spectrum of convolution $y(n) = x(n) * h(n)$ on a graph is equal to the product of signal spectra $Y(\lambda_k) = X(\lambda_k)H(\lambda_k)$. The convolution is then equal to the inverse transform of $Y(\lambda_k)$,

$$y(n) = x(n) * h(n) = \sum_{k=1}^N X(\lambda_k)H(\lambda_k)u_k(n).$$

This is the definition of the generalized convolution operator of two signals on a graph [3].

Convolution can be used to define the shift on a graph as $h(m-n) = h(m) * \delta_n(m) = \sum_{k=1}^N H(\lambda_k)u_k(n)u_k(m)$ where $\delta_n(m)$ is the delta function at the n th vertex. Its spectrum $\Delta_n(\lambda_k)$ is equal to the n th sample of the k th eigenfunction, since $\Delta_n(\lambda_k) = \sum_{m=1}^N \delta_n(m)u_k(m) = u_k(n)$.

The localized vertex spectrum (LVS) on a graph can be calculated as the spectrum of a graph function $x(n)$ multiplied by a shifted window $h(n-m)$

$$LVS_x(n, \lambda_k) = \sum_{m=1}^N x(m)h(m-n)u_k(m) = \sum_{m=1}^N x(m)h_{n,k}(m),$$

where the localized version of the window on vertex and frequency axes is denoted by

$$h_{n,k}(m) = \sum_{l=1}^N [H(\lambda_l)u_l(n)u_l(m)]u_k(m)$$

where we can use, for example, $H(\lambda_k) = C \exp(-\lambda_k \tau)$.

The inverse formula is then a sum of $LVS_x(n, \lambda_k)$, multiplied by the shifted and modulated windows $h_{n,k}(m)$,

$$x(n) = \frac{1}{\sum_{k=1}^N |H(\lambda_k)|^2 |u_k(n)|^2} \sum_{i=1}^N \sum_{k=1}^N LVS_x(i, \lambda_k) h_{i,k}(n).$$

Fourier analysis and Laplacian: Fourier analysis uses the idea that a signal, $x(t)$, can be expanded in terms of orthogonal basis functions $\cos(2\pi ft)$ and $\sin(2\pi ft)$. In other words, the resulting Fourier representation is a projection (scalar product) of the signal onto sinusoidal basis functions.

Interestingly enough, the Fourier expansion can be also considered from the Laplacian (Laplace differential operator), $\mathcal{L}\{x(t)\} = -d^2x(t)/dt^2$, point of view. The Laplacian eigenfunctions $u(t)$ are the solutions of $\mathcal{L}\{u(t)\} = \lambda u(t)$. We can easily conclude that $\cos(2\pi ft)$ and $\sin(2\pi ft)$ are the eigenfunctions of the Laplacian with the eigenvalues $\lambda = (2\pi f)^2$. Therefore, the Fourier analysis can be defined as an expansion of the signal $x(t)$ onto the set of eigenfunctions of the Laplacian operator.

In the discrete-time domain the Laplacian can be defined using a symmetric second-order difference operator

$$\begin{aligned}\mathcal{L}\{x(n)\} &= -x(n-1) + 2x(n) - x(n+1) \\ &= (x(n) - x(n-1)) + (x(n) - x(n+1)) \\ &= \sum_{m \in \{n-1, n+1\}} w_{nm}(x(n) - x(m)).\end{aligned}$$

The Laplacian is a matrix \mathbf{L} that can be used to transform a discrete-time signal $x(n)$ into its second-order difference. Eigenvectors of this Laplacian are the discrete-time sine and cosine functions.

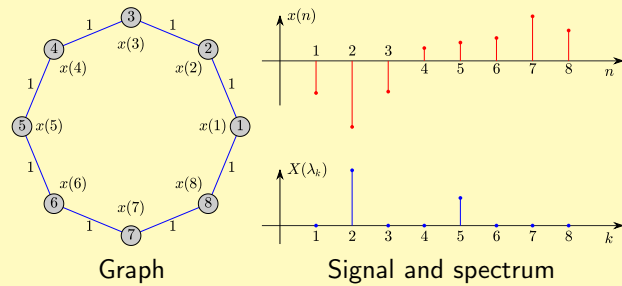
A graph corresponding to circular form of this Laplacian, sample signal $x(n)$ and corresponding spectrum $X(\lambda_k)$, for $N = 8$, are shown in the figure. Signal values are

assigned to graph vertices, and the resulting spectrum $X(\lambda_k)$ is obtained by decomposing $x(n)$ onto the Laplacian eigenvectors. The Laplacian matrix (in a circular form) is

$$\mathbf{L} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}.$$

The Laplacian operator maintains the relationship among vertices (signal samples) regardless of their ordering. Even if vertices (signal samples) are arbitrarily reordered, the eigenvectors of the Laplacian produce the same spectrum.

If the signal vector \mathbf{x} is a stacked-column representation of a two-dimensional $N \times N$ image, then with the summation for $m \in \{n-N, n-1, n+1, n+N\}$, when $w_{nm} = 1$ in $\mathcal{L}\{x(n)\}$, the two-dimensional Fourier analysis can be defined.



Box 1: Fourier analysis and Laplacian

Kirchhoff and “Ohm’s” laws on a graph: The Laplacian can be considered from the basic electric circuit theory point of view (Kirchhoff matrix). Let us assume that a graph represents an electric circuit. Then, the signal values $x(n)$ represent node voltages at the corresponding circuit vertices $x(n) = v(n)$. The weight coefficients $w_{nm} = 1/R_{nm}$ represent conductance (reciprocal resistance R_{nm}) values in the edges connecting vertices n and m , for vertices that are not connected by an edge $w_{nm} = 0$. The value of the current in the edge from vertex n to m is given by $i_{nm} = (v(n) - v(m))/R_{nm} = w_{nm}(x(n) - x(m))$. The sum of all currents going into a vertex n must be 0. In general, the external current source connected to vertex n is equal to the sum of all currents going from this vertex, $i_G(n) = \sum_j w_{nj}(x(n) - x(m)) = d_n x(n) - \sum_m w_{nm} x(m)$, where $d_n = \sum_m w_{nm}$.

The matrix form of the voltage to current relation is $\mathbf{Lx} = \mathbf{i}_G$, where \mathbf{L} is the Laplacian of the graph (circuit).

The node (vertex) voltage vector $\mathbf{v} = \mathbf{x}$ is determined (up to the constant referent voltage) from the vector of external currents \mathbf{i}_G via a system of linear equations.

As in the Fourier analysis, the solution of this system can be simplified using the spectral decomposition of the current and the node voltage vectors onto the set of eigenvectors of the Laplacian. Starting with $\mathbf{Lv} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T\mathbf{v} = \mathbf{i}_G$ and understanding that $\mathbf{U}^T\mathbf{U}$ is a unitary matrix, we obtain $\mathbf{\Lambda}\mathbf{U}^T\mathbf{v} = \mathbf{U}^T\mathbf{i}_G$. This represents an Ohm’s law analogue on a graph, $\lambda_k V(\lambda_k) = I(\lambda_k)$, where $V(\lambda_k) = \mathbf{u}_k^T \mathbf{v}$ and $I(\lambda_k) = \mathbf{u}_k^T \mathbf{i}_G$ are the spectral coefficients of the vertex voltage $x(n) = v(n)$, and the external vertex current $i_G(n)$ on the graph.

A similar analysis can be performed for a heat transfer flux, with edge weights representing heat transfer coefficients in an appropriate thermodynamics problem definition.

Box 2: Laplacian, Kirchhoff and “Ohm’s” laws on an electric circuit graph

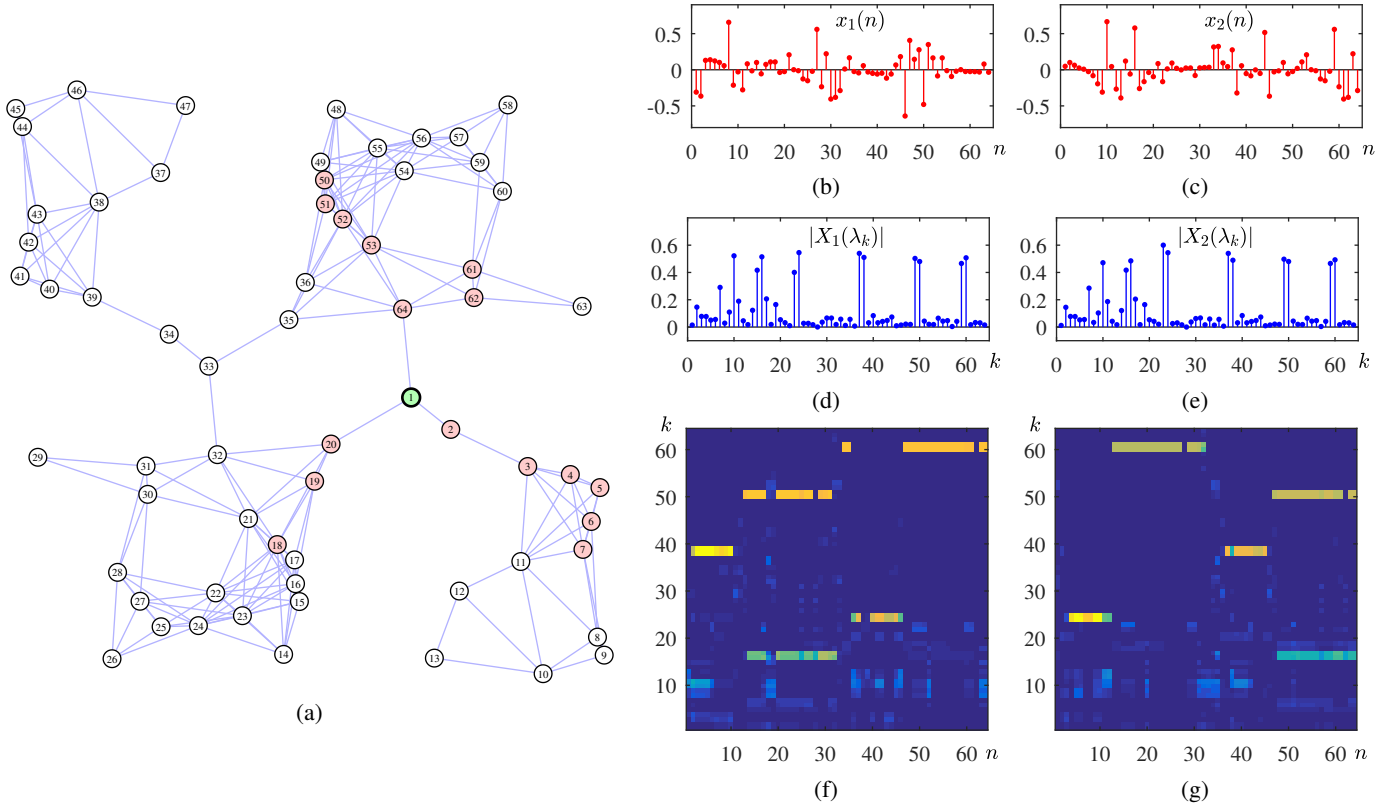


Fig. 2: Graph signal processing in action: (a) a graph with shaded nodes belonging to a localization window centered at the first vertex (the green shaded vertex); (b) a sample signal, $x_1(n)$; (c) a sample signal, $x_2(n)$; (d) the graph spectrum, $|X_1(\lambda_k)|$, of $x_1(n)$; (e) the graph spectrum, $|X_2(\lambda_k)|$, of $x_2(n)$; (f) a vertex-frequency representation of $x_1(n)$; and (g) a vertex-frequency representation of $x_2(n)$.

It should be mentioned that the outlined approach [3] can be computationally expensive, and a fast implementation algorithm is proposed in [9].

2) *Definition Based on Vertex Neighborhood:* To obtain a localized spectrum of a graph signal, we can utilize localization functions (windows) corresponding to window functions in classical signal processing. As in classical signal processing, a window function should be narrow enough to provide good localization of the spectral components, but wide enough to produce high resolution of such components. In other words, the window should contain the considered signal sample and some neighboring vertex samples. That is, the window is defined by a set of vertices that contain the current vertex n and all vertices that are close to the n th vertex.

There are several ways to define the local neighborhood for a vertex. For example, we can consider that two vertices are close if there is an edge between them, or if there is a path with its length (number of edges) smaller than an assumed threshold. Edge weights could also be taken into consideration to decide whether two nodes are close enough or not.

Commonly, the edge weights are given by $w_{nm} = \exp(-r_{nm}/\tau)$ for $r_{nm} < \kappa$, and $w_{nm} = 0$ otherwise. Here, r_{nm} denotes a distance between vertices, while τ and κ are constants. If we consider two arbitrary vertices n and m on a graph, then the path weight between these two vertices can be

defined as the product of all edge weights that are included in the considered path: $p_{nm} = w_{nk_1}w_{k_1k_2} \cdots w_{k_b m}$. If there is more than one path between n th and m th vertices, the shortest path (with the highest p_{nm} value) is considered. It can also be stated that the vertex m belongs to the local neighborhood of the n th vertex if $p_{nm} \geq h_T$, where h_T is a threshold defining the window size.

The simplest window has a value of $h_n(m) = 1$ for all vertices m that belong to the window centered at the n th vertex, and $h_n(m) = 0$ otherwise. It is analogous to a rectangular window in traditional signal processing. When $h_n(m) = 1$ for each m , the standard spectrum is obtained on a graph. We can define window function values $h_n(m)$ based on the distances p_{nm} that will attenuate farther vertex samples. Now we can define the signal localized around the n th vertex as $x_h(m) = x(m)h_n(m)$. The corresponding local spectrum is then defined as:

$$LV S_x(n, \lambda_k) = \sum_{m=1}^N x(m)h_n(m)u_k(m),$$

and the inverse definition follows from the inverse spectrum relation with additional summation over all vertices:

$$x(n) = \frac{1}{\sum_{i=1}^N h_i(n)} \sum_{i=1}^N \sum_{k=1}^N LV S_x(i, \lambda_k)u_k(n).$$

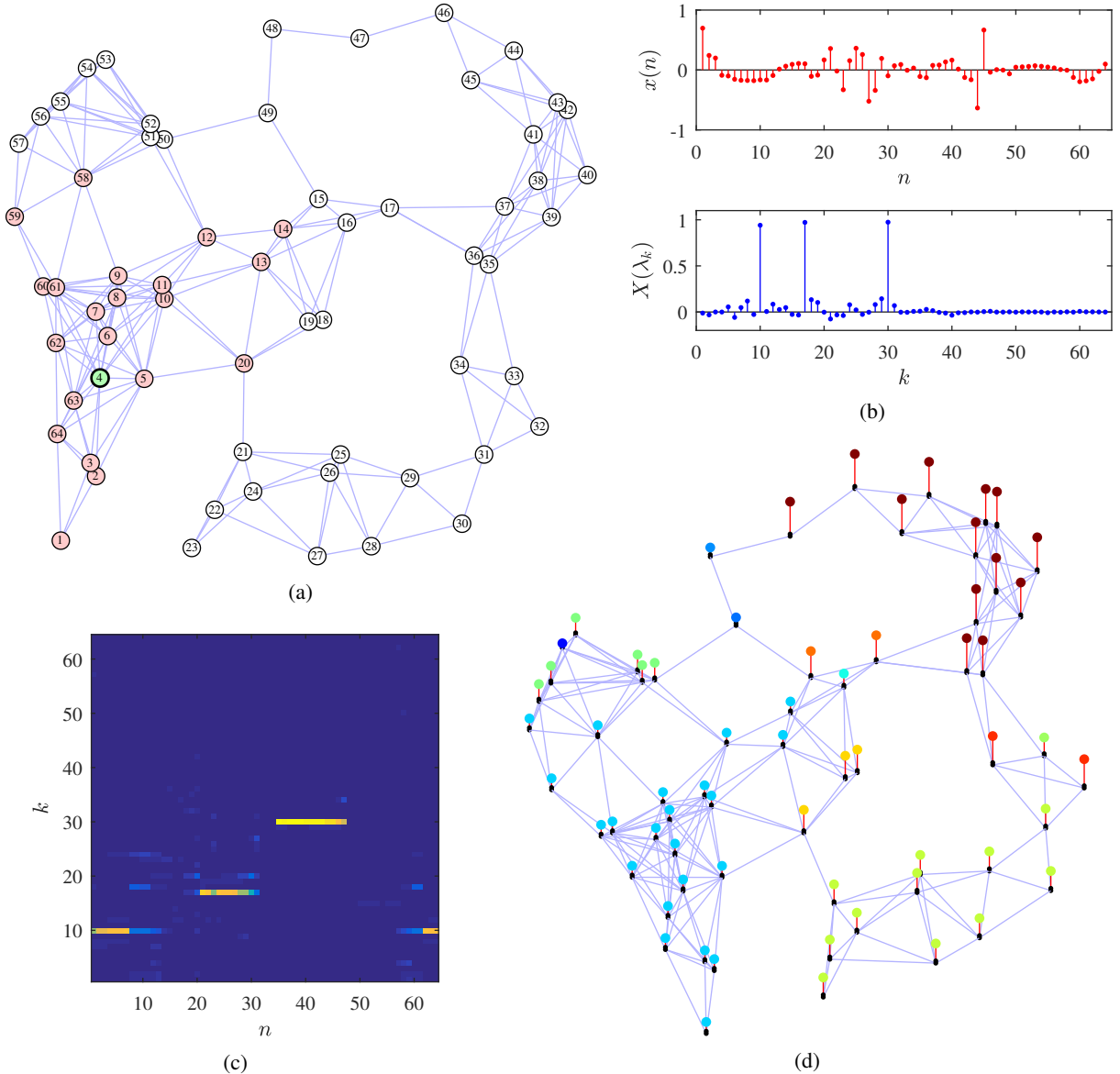


Fig. 3: Vertex-frequency analysis of a graph signal: (a) a graph and a window centered at the fourth vertex (the vertices included in the window are shaded); (b) a graph signal and its spectrum; (c) a vertex-frequency representation of the graph signal; and (d) an instantaneous (vertex-based) frequency representation.

Note that for the windowed signal $x(m)h_n(m)$ only $M \leq N$ samples are nonzero, meaning, it can be considered as a zero padded signal. To reconstruct this signal, we only need M spectral coefficients $LVS_x(n, \lambda_k)$ for M different values of λ_k . The remaining coefficients can be calculated from the system of equations obtained by using the fact that $x(m)h_n(m) = 0$ outside the window support. It produces a system of $N - M$ linear equations $\sum_{l=1}^N LVS_x(l, \lambda_k)u_k(n) = 0$ for vertices outside the window support. This system provides conditions for the spectral coefficients “interpolation” using M calculated values $LVS_x(n, \lambda_k)$.

In order to visualize the local spectral content, we should order vertices, i.e. find the Hamiltonian path in the corresponding graph. This ordering is not unique, and a possible way for ordering is to keep in mind that neighboring vertices

have the highest possible edge weights.

V. NUMERICAL EXAMPLES

Example 1: Let us first consider a vertex-frequency analysis of two graph signals shown in Figs. 2(b) and 2(c). For each vertex n we can define a window $h_n(m)$ and calculate the local spectrum $LVS_x(n, \lambda_k)$ of the windowed signal. A localized support for a window, centered at the 1st vertex, is presented in Fig. 2(a) with red shaded vertices corresponding to the window support. In this way, we obtain a 2D representation $LVS_x(n, k)$ of the analyzed signal presented, as a function of the vertex and eigenvalue index, in Fig. 2(c). We can see that different signals, having almost the same spectrum on graph, have different vertex-frequency representations.

Example 2: Consider a signal, $x(n)$, defined on a graph with $N = 64$ vertices as presented in Fig. 3(a). Let's assume that the signal values $x(n)$ are defined with Laplacian eigenvectors $u_k(n)$ as: $x(n) = u_{17}(n)$ for $17 \leq n \leq 32$, $x(n) = u_{30}(n)$ for $33 \leq n \leq 48$, and $x(n) = u_{10}(n)$ otherwise.

Signal samples and its spectrum are given in Fig. 3(b). The signal spectrum clearly depicts peaks at $k = 10$, $k = 17$ and $k = 30$. Small spectrum values (side lobes) around these eigenvalues exist since the components are not complete over all vertices. The vertex-frequency analysis of this signal is performed using localization window $h_n(m)$. A localization area for the $h_4(m)$ window, centered at the 4th vertex, is shown in Fig. 3(a) (red shaded vertices). The local spectrum $LV S_x(n, \lambda_k)$ of the windowed signal is calculated and presented in Fig. 3(c). From this representation, we can see localized signal components at “frequencies” $k = 10$, $k = 17$ and $k = 30$.

Finally, an instantaneous frequency representation is provided in Fig. 3(d). It should be noted that the “instantaneous frequency” definition for graphs is different from such a definition in traditional signal processing, where instantaneous frequency is defined as a signal's phase derivative with respect to time. Here, we determine the “frequency” (eigenvalue) index at each vertex, and this “frequency” index represents the k th index for which the spectrum reaches maximum at that particular vertex. Next, we can plot vertical lines with their lengths and colors proportional to the position (frequency) of the spectral maximum for each vertex as depicted in Fig. 3(d). This essentially yields a vertex-based instantaneous frequency representation depicting the localization of signal components on graph vertices.

VI. WHAT WE HAVE LEARNED

Graph signal processing is a new field that compliments traditional signal processing. While traditional signal processing techniques for the analysis of time-varying signals are well established, the corresponding graph signal processing equivalent approaches are in their infancy. In these lecture notes, we presented novel algorithms for the analysis of vertex-varying graph signals. We expect that the considered technique will find its many uses in neuroscience, social sciences, and genome processing, as graphs (networks) in those applications tend to be “non-stationary” and current analytical tools widely ignore this fact. Hence, the vertex-frequency analysis is of paramount importance for such applications.

AUTHORS

Ljubiša Stanković (ljubisa@ac.me) is professor at the University of Montenegro. His research interests include digital signal processing and time-frequency analysis. He is a Fellow of the IEEE.

Miloš Daković (milos@ac.me) is professor at the University of Montenegro. His research interests include signal processing and time-frequency analysis. He is a member of the IEEE.

Ervin Sejdić (esejdic@ieee.org) is an assistant professor at the University of Pittsburgh. His research interests include

biomedical signal processing, rehabilitation engineering, neuroscience, swallowing and gait. He received the Presidential Early Career Award for Scientists and Engineers in 2016, and the National Science Foundation CAREER Award in 2017. He is a senior member of the IEEE.

REFERENCES

- [1] A. Sandryhaila and J. M. F. Moura, “Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure,” *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80–90, Sept 2014.
- [2] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, May 2013.
- [3] D. I. Shuman, B. Ricaud, and P. Vandergheynst, “Vertex-frequency analysis on graphs,” *Applied and Computational Harmonic Analysis*, vol. 40, no. 2, pp. 260 – 291, 2016.
- [4] 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.
- [5] 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, Jun. 2014.
- [6] L. Stanković, M. Daković, T. Thayaparan, *Time-Frequency Signal Analysis with Applications*, Artech House, Boston, March 2013
- [7] B. Boashash, ed, *Time-Frequency Signal Analysis and Processing, A Comprehensive Reference*, Academic Press, 2015.
- [8] I. Jestrović, J. L. Coyle, and E. Sejdić, “Differences in brain networks during consecutive swallows detected using an optimized vertex–frequency algorithm,” *Neuroscience*, vol. 344, pp. 113–123, 2017.
- [9] —, “A fast algorithm for vertex-frequency representations of signals on graphs,” *Signal Processing*, vol. 131, pp. 483 – 491, 2017.