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

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

June 8, 2019

Lecture Outline

Introduction

- Graphs are irregular structures which naturally account for data integrity
- Multisensor and multinode measurements, likely recorded on irregular or ad-hoc grids
- In classic graph signal theory, the graphs are typically given (e.g., in various computer, social, road, transportation, and power networks)
- The first step in graph signal processing is to employ background knowledge of signal generating mechanisms in order to define the graph as a signal domain
- Graph signals benefit from the ability of graphs to incorporate signal similarity awareness

Relevance

A number of challenges in graph signal processing:

- The data sensing points (graph vertices) are usually well defined in advance
- Vertex connectivity (graph edges) is not often available.
- 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.
- The definition of an appropriate graph structure is a prerequisite for physically meaningful and computationally efficient graph signal processing applications.

History of Graph Theoretic Application

- The beginning of graph theory applications in electrical engineering dates back to the mid-XIX century and the definition of Kirchoff's laws.
- Graph models have since become a de facto standard for data analysis across the science and engineering areas, including chemistry, operational research, social networks, and computer sciences
- Graph theory as an optimization tool can be attributed to the seminal book by Nicos Christofides of Imperial College London, published in 1975
- 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 1970s.
- Current developments in graph theory owe their prominence to the emergence of modern data sources, such as large-scale and social networks

An Illustrative Example

Consider a multi-sensor setup for measuring a temperature field in a known geographical region. 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.

An Illustrative Example

- Classical signal processing requires an arrangement of the quintessentially spatial temperature samples into a linear structure
- “Lexicographic" ordering is not amenable to exploiting the spatial information related to the actual sensor arrangement, dictated by the terrain.
- This exemplifies that even a most routine temperature measurement setup requires a more complex estimation structure than the simple linear one corresponding to the classical signal processing framework
- To introduce a “situation-aware" noise reduction scheme for the temperature field, we proceed to explore a graph-theoretic framework to this problem, starting from a local signal average operator.

Local Signal Average

- In classical Signal Processing this can be achieved through a moving average operator, through averaging across the neighboring neighboring nodes, in the linear graph.
- Since the sensor network measures a set of related temperatures from irregularly spaced sensors, an effective estimation strategy should include domain knowledge.
- For example, for the sensing points $n = 20$ and $n = 37$, the “domain knowledge aware" local estimation takes the form

$$y(20) = x(20) + x(19) + x(22) + x(23) \tag{1}$$

$$y(37) = x(37) + x(32) + x(33) + x(35) + x(61). \tag{2}$$

- The full set of relations among the sensing points can be arranged into the matrix form

$$\mathbf{y} = \mathbf{x} + \mathbf{A}\mathbf{x}, \tag{3}$$

Adjacency Matrix

- The matrix $\mathbf{A}$ is the **connectivity or adjacency matrix** of a graph. It indicates structure of the neighboring sensing locations that should be involved in each $y(n)$.
- The elements of matrix $\mathbf{A}$ are either 1 (if the corresponding vertices are related) or 0 (if they are not related).

Graph Processing Framework

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**,
- The vertex-to-vertex lines indicating the connectivity among the sensing points are called the **graph edges**,
- The vertices and edges form **a graph**, a new and very structurally rich signal domain,
- The measured temperatures are now interpreted as **signal samples on graph**,
- Similar to traditional signal processing, this new **graph signal** may have many realizations on the same graph and may include noise,
- Through relation (3), we have therefore introduced a simple **graph system** for physically and spatially aware signal averaging (a linear first-order graph system).

Graph Signal

From a multi-sensor measurement to a graph signal

Weighted Graph

- To emphasize our trust in a particular sensor (i.e., to model sensor relevance), a weighting scheme may be imposed on the edges (connectivity) between the sensing points,

$$y(n) = x(n) + \sum_{m \neq n} W_{nm} x(m). \tag{4}$$

- We have now arrived at a weighted graph, whereby each edge has an associated weight, W_{nm},
- A matrix form of a weighted cumulative graph signal

$$\mathbf{y} = \mathbf{x} + \mathbf{W}\mathbf{x}. \tag{5}$$

- The weighting coefficients within the estimate for each $y(n)$ should sum up to unity.

Degree Matrix and Laplacian

- A normalized form of (5)

$$\mathbf{y} = \frac{1}{2}(\mathbf{x} + \mathbf{D}^{-1}\mathbf{W}\mathbf{x}), \tag{6}$$

- The diagonal normalization matrix, $\mathbf{D}$, is called **the degree matrix**, are $D_{nn} = \sum_m W_{nm}$.
- An important operator for graph signal processing is the **graph Laplacian**, $\mathbf{L}$, which is defined as

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

is a combination of the degree matrix and weighting matrix.

Graph Specification

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
- The “connectivity strength" weighting 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/weighting 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/weighting matrix is not symmetric and such graphs are called **directed graphs**.

Graph Topology (Edges and Weights) I

There are three possible classes of problems which dictate the definition of graph edges:

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

$$W_{mn} = e^{-r_{mn}^2/\alpha} \text{ or } W_{mn} = e^{-r_{mn}/\alpha}$$

if $r_{mn} < \tau$ and $W_{mn} = 0$ for $r_{mn} \geq \tau$. This form has been used in the graph in Fig. 2, whereby the altitude difference, h_{mn}, was accounted for as $W_{mn} = e^{-r_{mn}/\alpha} e^{-h_{mn}/\beta}$.

Graph Topology (Edges and Weights) II

- **Physically well defined relations among the sensing positions:** Examples include electric circuits, linear heat transfer systems, spring-mass systems, and various forms of networks like social, computer or power networks. In these cases, the edge weights are defined as a part of problem definition.
- **Data similarity dictates the underlying graph topology:** This scenario is the most common in image and biomedical signal processing. Various approaches can be used to define data similarity, including the correlation matrix between the signals at various sensors or the corresponding inverse covariance (precision) matrix. Learning a graph (its edges) based on the set of the available data is an interesting and currently extensively studied research area.

System on a Graph - Graph Shift

- **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 signal shift operator can then be compactly defined using the graph adjacency matrix as

$$\mathbf{x}_{shifted} = \mathbf{A}\mathbf{x}.$$

The energy of the shifted signal is not the same as the energy of the original signal (graph shit is not isometric).

System on a Graph I

- A system on a graph can be implemented as a linear combination of a graph signal and its graph shifted versions

$$\mathbf{y} = h_0\mathbf{W}^0\mathbf{x} + h_1\mathbf{W}^1\mathbf{x} + \cdots + h_{M-1}\mathbf{W}^{M-1}\mathbf{x} = \sum_{m=0}^{M-1} h_m\mathbf{W}^m\mathbf{x}, \quad (7)$$

  where, by definition $\mathbf{W}^0 = \mathbf{I}$, while $h_0, h_1, \ldots, h_{M-1}$ are the system coefficients to be found.

- Corresponding classic system is a standard FIR filter,

$$y(n) = h_0x(n) + h_1x(n-1) + \cdots + h_{M-1}x(n-M+1). \quad (8)$$

System on a Graph II

- The Laplacian operator applied on a signal, $\mathbf{Lx}$, can be considered as a combination of the scaled original signal, $\mathbf{Dx}$, and its weighted shifted version, $\mathbf{Wx}$, since $\mathbf{Lx} = \mathbf{Dx} - \mathbf{Wx}$.
- A system defined using the Laplacian

$$\mathbf{y} = \mathbf{L}^0\mathbf{x} + h_1\mathbf{L}^1\mathbf{x} + \cdots + h_{M-1}\mathbf{L}^{M-1}\mathbf{x} \tag{9}$$

  therefore allows us to always produce an unbiased estimate of a constant c, that is, if $\mathbf{x} = \mathbf{c}$ then $\mathbf{y} = \mathbf{c}$.
- simple first order system based on the graph Laplacian can be written as

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

Properties of a System on a Graph

- A system on a graph is conveniently defined by the “graph transfer function", $H(\mathbf{W})$, as

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

- For an unweighted graph, $\mathbf{W} = \mathbf{A}$
- We can also use the Laplacian matrix, $\mathbf{L} = \mathbf{D} - \mathbf{W}$.
- Linear, if

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

- Shift invariant, if

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

- The following system on a graph is linear and shift invariant,

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

Graph Fourier Transform

Spectral representations of graph signals employ either the adjacency/weighting matrix or the graph Laplacian eigenvalue decomposition. For the latter case we have

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

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

- The eigenvectors, $\mathbf{u}_k$, are used as a set of bases for spectral segmentation of graphs.
- The graph Fourier transform, $\mathbf{X}$, of a graph signal, $\mathbf{x}$, is defined as

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

- The element $X(k)$ of $\mathbf{X}$ is a projection of $\mathbf{x}$ onto $\mathbf{u}_k$,

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

Inverse Graph Fourier Transform

The inverse graph Fourier transform is obtained as

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

or

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

In analogy to the classic 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 $\mathbf{u}_k = [1, e^{j2\pi k/N}, \ldots, e^{j\pi(N-1)k/N}]^T/\sqrt{N}$, the graph Fourier transform 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.

Classical Fourier Analysis as a Special Case

- Classic spectral analysis can considered as a special case of graph signal spectral analysis, with the adjacency matrix defined on an unweighted circular directed graph (a line graph with the circularly connected last and first vertex), when $\mathbf{u}_k = [1, e^{j2\pi k/N}, \ldots, e^{j\pi(N-1)k/N}]^T/\sqrt{N}$.
- 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 previous vector elements $u_k(n)$ and eigenvalues λ_k are the solutions of this difference equation.

Graph Segmentation Based on the Eigenvectors

- The Laplacian quadratic form of an eigenvector (its smoothness index) is equal to the corresponding eigenvalue,

$$\mathbf{u}_k^T(\mathbf{L}\mathbf{u}_k) = \mathbf{u}_k^T(\lambda_k \mathbf{u}_k) = \lambda_k .$$

- The eigenvector corresponding to $\lambda_1 = 0$ is a constant (maximally smooth for any vertex ordering).
- Spectral similarity of vertices is defined using eigenvectors, if the eigenvector elements $u_k(n)$, $k = 1, 2, \ldots, P$ are assigned to the vertex n. If $\mathbf{u}_1$ is omitted, then a $(P-1)$-dimensional spectral vector becomes $\mathbf{q}_n = [u_2(n), \ldots, u_P(n)]^T$.
- The spectral similarity between vertices n and m is defined as the two-norm $\|\mathbf{q}_n - \mathbf{q}_m\|_2$.

Graph Segmentation

- Keep the original vertex positions and color them according to the spectral vectors $\mathbf{q}_n$.
- Coloring is performed using the eigenvector elements $u_2(n)$, $u_3(n)$, and $u_4(n)$ as color coordinates for the vertex n.

  - Graph segmentation, by grouping vertices with similar colors.
  - The graph segmentation is a signal-independent operation. It roughly indicates the data connectivity between sensor data values on this graph, and suggests that the data processing will predominantly be localized within these regions.

Spectral Domain of System on Graphs

- A system on a graph, defined by its Laplacian matrix,

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

- Employing the graph spectral representation, $\mathbf{L} = \mathbf{U}\Lambda\mathbf{U}^{-1}$,

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

  where

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

  is the **transfer function of the graph system**.

- With $\mathbf{U}^{-1}\mathbf{y} = H(\Lambda)\mathbf{U}^{-1}\mathbf{x}$, the input and output signal graph Fourier transforms are related by

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

Spectral Domain Filtering

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

1. Calculate the GDFT of the input graph signal $\mathbf{X} = \mathbf{U}^{-1}\mathbf{x}$,
2. Multiply the GDFT of the input graph signal with transfer function $G(\Lambda)$ to obtain $\mathbf{Y} = G(\Lambda)\mathbf{X}$, and
3. Calculate the output graph signal as the inverse graph Fourier transform 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

Vertex Domain Filtering

Physically, the minimum of $\mathbf{x}\mathbf{L}\mathbf{x}^T$ implies the smoothest possible signal and to arrive at this solution we may employ steepest descent.

- The signal value at an iteration p is adjusted in the opposite direction of the gradient, $\partial E_x/\partial \mathbf{x}^T = 2\mathbf{L}\mathbf{x}$
- This yields the iterative procedure

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

- The signal $\mathbf{x}_{p+1}$ is as an output of the first order system.
- The minimum of the quadratic form $\mathbf{x}\mathbf{L}\mathbf{x}^T$ corresponds to a constant signal. To avoid obtaining only constant steady state, the above iteration process can be used in alternation with

$$\mathbf{x}_{p+2} = (\mathbf{I} + \beta \mathbf{L})\mathbf{x}_{p+1}$$

- This two-step iterative processes is known as **Taubin's $\alpha - \beta$ algorithm.**

Vertex Domain Filtering Results

For appropriate values of α and β, this system can give a good and very simple approximation of a graph low-pass filter.
The original noisy signal 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 $SNR_0 = 14.2$ dB to 26.8 dB.

Optimal Denoising

Consider a measurement which is composed of a slow-varying desired signal, $\mathbf{s}$, and a superimposed fast changing disturbance, $\boldsymbol{\varepsilon}$, to give

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

The aim is to design a graph filter for disturbance suppression (denoising), the output of which is denoted by $\mathbf{y}$.

- The optimal denoising task can then be defined through a minimization of the cost function

$$J = \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 + \alpha \mathbf{y}^T \mathbf{L} \mathbf{y}. \tag{21}$$

- The minimization of the first term, $\frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2$, enforces the output signal, $\mathbf{y}$, to be as close as possible, in terms of the minimum residual disturbance power, to the available observations, $\mathbf{x}$.
- The second term, $\mathbf{y}^T \mathbf{L} \mathbf{y}$, represents a measure of smoothness of the graph filter output, $\mathbf{y}$.

Solution for the Optimal Denoising

- The solution to this minimization problem follows from

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

- It results in a denoiser

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

- The Laplacian spectral domain form of this relation is

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

- The corresponding graph filter transfer function

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

- Using $\alpha = 4$, the obtained output signal-to-noise ratio for the previous graph signa was $SNR = 26$ dB, a 11.8 dB improvement over the original $SNR_0 = 14.2$ dB.

Graph Topology Based on Signal Similarity

In some application signal values themselves (or appropriate correlations) are used as an indicator of signal similarity.

- In image processing, for the image intensity values at pixels indexed by n and m, denoted by $x(n)$ and $x(m)$, the difference of intensities is defined using an exponential kernel, as

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

- The corresponding weights may be defined as

$$W_{nm} = e^{-(x(n)-x(m))^2/\tau^2}$$

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

Image Graph Filtering

Consider the problem of denoising a 50×50 pixel, 8-bit grayscale, image. The vertices of the graph are the pixel locations. The edge weights for the graph representation of this image are calculated with $\kappa = \sqrt{2}$ and $\tau = 20$. Low-pass filtering is performed on the corresponding image graph using iterative filtering (Taubin's algorithm) over 200 iterations, with $\alpha = 0.1$ and $\beta = 0.15$.

Originaln, noise corrupted, and filtered image using Taubin's

What We Have Learned

- Natural signals (speech, biomedical, video) reside over irregular domains and are, unlike the signals in communications, not adequately processed using, e.g., standard harmonic analyses.
- 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.
- 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.

Refrences I

- N. Christofides, “Graph theory: An algorithmic approach", Academic Press, 1975.
- F. Afrati and A. G. Constantinides, “The use of graph theory in binary block code construction", in Proceedings of the International Conference on Digital Signal Processing, pp. 228-233, Florence, Italy, 310 August - 2 September, 1978
- 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 Proceedings F-Communications, Radar and Signal Processing, vol. 133, no. 2, pp. 146-152, 1986.
- S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, “Discrete signal processing on graphs: Sampling theory," *IEEE Trans. on Signal Processing*, vol. 63, no. 24, pp. 6510-6523, Dec.15, 2015.

Refrences II

- A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs," *IEEE. Trans. on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.
- A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs: Frequency analysis," *IEEE. Trans. on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, Jun. 2014.
- 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 Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

Refrences III

- L. Stanković, M. Daković, and E. Sejdić, "Vertex-frequency analysis: A way to localize graph spectral components," *IEEE Signal Processing Mag.*, Vol.34, July 2017.
- S. Saito, H. Suzuki, and D. P. Mandic, "Hypergraph p-Laplacian: A differential geometry view", Proceedings of the the Thirty Second AAAI Conference on Artificial Intelligence (AAAI-18), pp. 3984-3991, 2018.
- L. Stanković and E. Sejdić, "*Vertex-frequency analysis of graph signals,*", Springer Nature, 2019.
- M.H. Hayes, "*Statistical digital signal processing and modeling*," John Wiley & Sons, 2009.