Signal Processing on Graphs

Santiago Segarra, Weiyu Huang, and Alejandro Ribeiro

April 5, 2021

In previous weeks, we have focused our attention on discrete time signal processing, image processing, and principal component analysis (PCA). These three seemingly unrelated areas can be thought of as the study of signals on particular graphs: a directed cycle, a lattice and a covariance graph. Thus, the theory of signal processing for graphs can be conceived as a unifying theory which develops tools for more general graph domains and, when particularized for the mentioned graphs, recovers some of the existing results.

1 Intro to graph theory

Formally, a graph is a triplet $G = (\mathcal{V}, \mathcal{E}, W)$ where $\mathcal{V} = \{1, 2, \dots, N\}$ is a finite set of N nodes or vertices, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges defined as order pairs (n, m) and $W : \mathcal{E} \to \mathbb{R}$ is a map from the set of edges to scalar values, w_{nm}. Weights w_{nm} represent the similarity or level of relationship from n to m. The adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ of a graph is defined as

$$A_{mn} = \begin{cases} w_{nm}, & \text{if } (n, m) \in \mathcal{E}; \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

In unweighted graphs, A_{mn} is either 1 (if nodes n and m are connected) or 0 (otherwise). For undirected graphs, the adjacency matrix is symmetric, i.e. $w_{nm} = w_{mn}$ for all nodes n and m. When this is not the case, we say that the graph is directed.

Graph signals are mappings $x : \mathcal{V} \to \mathbb{R}$ from the vertices of the graph into the real (or complex) numbers. Graph signals can be represented as vectors $\mathbf{x} \in \mathbb{R}^N$ where x_n stores the signal value at the nth vertex in $\mathcal{V}$. Notice that this assumes an indexing of the nodes, which coincides with the indexing used in the adjacency matrix.

The degree of a node is the sum of the weights of the edges incident to this node. Formally, the degree of node i, $\deg(i)$ is defined as

$$\deg(i) = \sum_{j \in \mathcal{N}(i)} w_{ij}, \tag{2}$$

where $\mathcal{N}(i)$ stands for the neighborhood of node i, i.e., all other nodes connected to node i. The degree matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$ is a diagonal matrix such that $D_{ii} = \deg(i)$. In directed graphs, each node has an out-degree (sum of the weights of all edges out of the node) and an in-degree (sum the weights of all edges into the node).

Given a graph G with adjacency matrix $\mathbf{A}$ and degree matrix $\mathbf{D}$, we define the Laplacian matrix $\mathbf{L} \in \mathbb{R}^{N \times N}$ as

$$\mathbf{L} = \mathbf{D} - \mathbf{A}. \tag{3}$$

Equivalently, $\mathbf{L}$ can be defined elementwise as

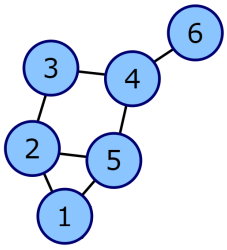
$$L_{ij} = \begin{cases} \deg(i), & \text{if } i = j; \\ -w_{ji}, & \text{if } (j,i) \in \mathcal{E}; \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The Laplacian acts as a difference operator on graph signals. To see why this is true, consider a graph signal $\mathbf{x}$ on graph G and define the new signal $\mathbf{y} = \mathbf{L}\mathbf{x}$ where each element y_i is computed as

$$y_i = [\mathbf{L}\mathbf{x}]_i = \sum_{j \in \mathcal{N}(i)} w_{ji}(x_i - x_j). \tag{5}$$

Notice that the element y_i measures the difference between the value of the signal $\mathbf{x}$ at node i and at its neighborhood. The Laplacian has very specific spectral properties. In particular, the Laplacian of any undirected graph is positive semi-definite (all its eigenvalues are nonnegative) and has an eigenvalue of 0. Moreover, the multiplicity of the 0 eigenvalue is equal to the number of connected components in the graph.

Given an arbitrary graph $G = (\mathcal{V}, \mathcal{E}, W)$, a graph-shift operator $\mathbf{S} \in \mathbb{R}^{N \times N}$ is a matrix satisfying $S_{ij} = 0$ for $i \neq j$ and $(i,j) \notin \mathcal{E}$. That is, $\mathbf{S}$ can take nonzero values in the edges of G or in its diagonal. Some common choices for $\mathbf{S}$ include the adjacency matrix $\mathbf{A}$ and the Laplacian $\mathbf{L}$. We consider normal graph-shift operators, i.e. operators that can be written as $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H$ where the columns of $\mathbf{V}$ are the eigenvectors of $\mathbf{S}$ and $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues of $\mathbf{S}$. See Fig. 1 for an example graph.



$$\mathbf{S} = \begin{pmatrix} S_{11} & S_{12} & 0 & 0 & S_{15} & 0 \\ S_{21} & S_{22} & S_{23} & 0 & S_{25} & 0 \\ 0 & S_{23} & S_{33} & S_{34} & 0 & 0 \\ 0 & 0 & S_{43} & S_{44} & S_{45} & S_{46} \\ S_{51} & S_{52} & 0 & S_{54} & S_{55} & 0 \\ 0 & 0 & 0 & S_{64} & 0 & S_{66} \end{pmatrix}$$

Figure 1. Example of a six node graph and its corresponding graph shift operator. Observe that the nodes are labeled, and that the graph shift operator is nonzero only where there is an edge, or in the diagonal.

For a given graph-shift operator $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H$, the Graph Fourier Transform (GFT) of $\mathbf{x}$ is defined as

$$\tilde{x}(k) = \langle \mathbf{x}, \mathbf{v}_k \rangle = \sum_{n=1}^{N} \mathbf{x}(n)\mathbf{v}_k^*(n). \tag{6}$$

Equation (6) can be rewritten in matrix form to obtain

$$\tilde{\mathbf{x}} = \mathbf{V}^H\mathbf{x}. \tag{7}$$

Since the columns of $\mathbf{V}$ are the eigenvectors $\mathbf{v}_k$ of $\mathbf{S}$, $\tilde{x}(k) = \mathbf{v}_k^H\mathbf{x}$ is the inner product between $\mathbf{v}_k$ and $\mathbf{x}$. We think of the eigenvectors $\mathbf{v}_k$ as oscillation modes associated to the eigenvalues in the same way that, in discrete time signal processing, different complex exponentials are associated to frequency values. In particular, GFT is equivalent to DFT when $\mathbf{V} = \mathbf{F}$, i.e. $\mathbf{v}_k = \mathbf{e}_{kN}$, the complex exponential vector.

In order to measure how much a signal oscillates within a graph, the concept of total variation can be extended from traditional signal processing. Classically, the total variation of a signal is defined as the sum of squared differences in consecutive signal samples, $\sum_n (x_n - x_{n-1})^2$. This concept can be extended to graphs where the notion of neighborhood replaces that of consecutive nodes to obtain

$$TV_G(\mathbf{x}) = \sum_{n=1}^{N} \sum_{m \in \mathcal{N}(n)} (x_n - x_m)^2 w_{mn} = \mathbf{x}^T\mathbf{L}\mathbf{x}. \tag{8}$$

As can be seen from (8) the total variation of a signal in a graph can be written as a quadratic form that depends on the Laplacian of that graph.

Total variation allows us to interpret the ordering of the eigenvalues of the Laplacian in terms of frequencies, i.e., larger eigenvalues correspond to higher frequencies (larger total variation). The eigenvectors associated with large eigenvalues oscillate rapidly whereas the eigenvectors associated with small eigenvalues vary slowly.

The inverse graph Fourier transform (iGFT) of a graph signal $\tilde{\mathbf{x}} \in \mathbb{R}^N$ is given by

$$x(n) = \sum_{k=0}^{N-1} \tilde{x}(k) v_k(n), \tag{9}$$

which can be rewritten in matrix form to obtain

$$\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}}. \tag{10}$$

The orthonormality of $\mathbf{V}$ ensures that, indeed, the GFT and iGFT are inverse operations. Orthonormality also allows the extension of other classical results to the graph domain, e.g., Parseval's theorem.

2 Connection with traditional signal processing

Let us begin by analyzing the connection between graph signal processing and traditional finite discrete time signal processing. The latter is a particular case of the former when the graph considered is a directed cycle, as we will discuss in this section. Recall that the adjacency matrix of a directed cycle is

$$\mathbf{A}_{dc} = \begin{bmatrix} & & & & 1 \\ 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \end{bmatrix}, \tag{11}$$

where the unspecified entries are zeros.

2.1 Generate a directed cyclic graph. Write a Python Class that takes as an input the number N of nodes in a graph and generates the adjacency matrix of a directed cyclic graph as described in (11). Show your output (through *matplotlib.pyplot.imshow* for example) for $N = 20$.

2.2 Compare your graph basis with the Fourier basis. Fix $N = 20$ and consider the graph-shift operator $\mathbf{S}_1 = \mathbf{A}_{dc}$ equal to the adjacency matrix of the directed cycle. Consider the discrete Fourier basis $\mathbf{F}$ and show the result of the following computation

$$\mathbf{F}^H \mathbf{S}_1 \mathbf{F} = \mathbf{\Lambda}_1. \tag{12}$$

What is the structure of matrix $\mathbf{\Lambda}_1$? What does this tell you about the columns of $\mathbf{F}^H$? Confirm your answer by showing that the first column of $\mathbf{F}^H$ indeed satisfies the property stated. What does this tell you about the relation between DFT and GFT for this particular graph-shift operator?

Consider now a symmetric graph with adjacency matrix $\mathbf{A}_{sc} = \mathbf{A}_{dc} + \mathbf{A}_{dc}^T$ and pick as a graph-shift operator its Laplacian $\mathbf{S}_2 = \mathbf{L}_{sc}$. Repeat the computation in (12) for $\mathbf{S}_2$. What is the relation between DFT and GFT for this new operator $\mathbf{S}_2$?

3 The graph frequency domain

In this section we are going to analyze the frequency representation of different graph signals defined on a graph.

3.1 Compute the Graph Fourier Transform of a graph signal. Write a class in Python that takes as an input a graph shift $\mathbf{S} \in \mathbb{R}^{N \times N}$ and a signal $\mathbf{x} \in \mathbb{R}^N$ defined on the graph and generates as output $\tilde{\mathbf{x}} \in \mathbb{R}^N$, the graph Fourier representation of $\mathbf{x}$. Make sure to order the eigenvectors of $\mathbf{S}$ in increasing order of absolute value of the associated eigenvalues.

3.2 Understanding the data. Load the file *graph_sp_data.pkl*. You will see the adjacency matrix of a graph $\mathbf{A} \in \mathbb{R}^{50 \times 50}$, and four graph signals called $\mathbf{x}_i \in \mathbb{R}^{50}$ for $i = 1, 2, 3$ and $\mathbf{y} \in \mathbb{R}^{50}$. How many connected components does the graph have? Plot signals $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$. Can you tell which one 'varies faster' in the graph domain?

3.3 Finding the frequency representation of signals. For the remainder of the lab practice, we define as graph-shift $\mathbf{S} = \mathbf{L}$ the Laplacian of the loaded graph with adjacency matrix $\mathbf{A}$. Using your function in Section 3.1, plot $\tilde{\mathbf{x}}_1$, $\tilde{\mathbf{x}}_2$, $\tilde{\mathbf{x}}_3$, i.e. the frequency representations of $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$, respectively. By looking at the plots, can you tell which signal varies slower and which one varies faster in the graph domain?

3.4 Quantifying the variation of signals. Use total variation through the Laplacian quadratic form in (8) to quantify the variation of a signal in a graph. Do these results confirm your intuition from Section 3.3?

3.5 Compute the inverse Graph Fourier Transform of a graph signal. Write a class in Python that takes as an input a graph shift $\mathbf{S} \in \mathbb{R}^{N \times N}$ and the frequency coefficients $\tilde{\mathbf{x}} \in \mathbb{R}^N$ of a graph signal and outputs $\mathbf{x} \in \mathbb{R}^N$, the original signal. Make sure to order the eigenvectors of $\mathbf{S}$ in increasing order of absolute value of the associated eigenvalues.

3.6 Reconstruction and Parseval's theorem. If you need to compress $\mathbf{x}_1$ by only keeping $K = 5$ frequency coefficients, which ones would you keep? Perform the reconstruction and plot the original signal and the reconstructed one. Also, compute the energy of the error. Can you compute the energy of the reconstruction error without actually performing the reconstruction? What quality of the GFT allows you to do this?

3.7 Denoising a graph signal. Assume that graph signal $\mathbf{y}$ loaded from the file *graph_sp_data.pkl* is in fact composed of a graph signal $\mathbf{z}$ of bandwidth 3 contaminated with white noise. Your objective is to recover $\mathbf{z}$ by keeping the correct frequency coefficients. Plot $\tilde{\mathbf{y}}$, $\mathbf{y}$ and your recovered signal $\mathbf{z}$.