Signal Processing on Graphs – Classification of Cancer Types

Weiyu Huang, Santiago Segarra, and Alejandro Ribeiro

April 19, 2016

In last week, we studied graph signal processing, which defines the concept of frequency and graph Fourier transform for signals supported on graphs. We observed that the traditional finite discrete time signal processing can be viewed as a particular case of the graph signal processing when the graph considered is a directed cycle. In the lab this week, we are going to complete our journey in ESE 224 with an application of graph signal processing to improve the classification of cancer types through the use of genetic networks.

1 Review of graph theory and graph filters

Formally, a graph is a triplet $G = (\mathcal{V}, \mathcal{E}, W)$ where $\mathcal{V} = \{1, 2, \ldots, N\}$ 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

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

In unweighted graphs, A_{nm} 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

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

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

$$\tilde{\mathbf{x}}(k) = \langle \mathbf{x}, \mathbf{v}_k \rangle = \sum_{n=1}^{N} \mathbf{x}(n)\mathbf{v}_k^*(n). \tag{4}$$

Equation (4) can be rewritten in matrix form to obtain

$$\tilde{\mathbf{x}} = \mathbf{V}^H\mathbf{x}. \tag{5}$$

Since the columns of $\mathbf{V}$ are the eigenvectors $\mathbf{v}_i$ of $\mathbf{S}$, $\tilde{\mathbf{x}}(k) = \mathbf{v}_k^H\mathbf{x}$ 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 $\mathbf{V}^H = \mathbf{F}$, 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

$$TV_G(\mathbf{x}) = \sum_{n=1}^{N} \sum_{m \in \mathcal{N}(n)} (x_n - x_m)^2 w_{mn} = \mathbf{x}^T\mathbf{L}\mathbf{x}. \tag{6}$$

As can be seen from (6) 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

$$\mathbf{x}(n) = \sum_{k=0}^{N-1} \tilde{\mathbf{x}}(k) v_k(n), \tag{7}$$

which can be rewritten in matrix form to obtain

$$\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}}. \tag{8}$$

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.

Given a graph signal $\mathbf{x}$ with its GFT $\tilde{\mathbf{x}}$, we can filter the graph signal by passing it through a filter H. In frequency domain, the filtered GFT is the multiplication of the original GFT with the frequency response of the filter, i.e.

$$\hat{\tilde{\mathbf{x}}} = H\tilde{\mathbf{x}}. \tag{9}$$

And the filtered signal in the graph domain $\hat{\mathbf{x}}$ can be recovered by performing inverse GFT onto the filtered GFT,

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

2 Genetic network

Let us begin by analyzing the genetic network describing gene-to-gene interactions. The network is downloaded from the *NCI Nature database* (link: http://pid.nci.nih.gov/download.shtml). The network consists of 2458 genes and loosely speaking, two genes are connected if the proteins encoded by them participate in the same metabolism process.

2.1 Understanding the data. Load the file *geneNetwork_rawPCNCI.mat*. You will see the gene network $\mathbf{A} \in \mathbb{R}^{2458\times 2458}$. This is our adjacency matrix for the following analysis. Does this graph contains self-loops? Is the graph directed or undirected, weighted or unweighted? For graphs possessing this many nodes, it would be very useful to visualize the graph to get a sense of how does the graph looks like. Plot the graph using the *spy* command in MATLAB.

Construct $\mathbf{L}$ the Laplacian of the loaded adjacency matrix $\mathbf{A}$. Suppose we define another adjacency matrix $\tilde{\mathbf{A}}$ with all self-loops removed, is $\tilde{\mathbf{L}}$ the Laplacian of $\tilde{\mathbf{A}}$ different from $\mathbf{L}$? Explain why.

2.2 Total variations. For the graph-shift $\mathbf{S} = \mathbf{L}$, perform eigendecomposition and find its eigenvectors. Compute the total variation $TV_G(\mathbf{v}_k)$ for each of its eigenvectors $\mathbf{v}_k$ using (6). Plot the total variations for all eigenvectors versus their corresponding eigenvalues. What can you say about this? Recall that the total variation of a signal is defined to quantify how much a signal oscillates within a graph, what does your finding imply about the ordering of frequencies, i.e. does eigenvectors associated with larger eigenvalues oscillate more rapidly or more slowly?

3 Genetic profiles

In this section, we are going to study the genetic profiles of 240 patients diagnosed with different subtypes of ovarian cancer. We will see that by interpreting these genetic profiles as graph signals defined on the genetic networks, we will be able to clearly distinguish patients from different subtypes.

Load the file *signal_mutation.mat*. You will see the aggregated graph signals $\mathbf{X} \in \mathbb{R}^{240\times 2458}$. The i-th row of this matrix represents the genetic profile $\mathbf{x}_i$ for the i-th patient. The n-th gene for this patient is mutated if the n-th entry of $\mathbf{x}_i$ is 1 and is not mutated (normal) if the n-th entry is 0.

Patients diagnosed with the same disease may exhibit different phenotypes, i.e. different variations of the same disease. The most effective therapies for different phenotypes may differ a lot and, for this reason, it is very beneficial if we can distinguish phenotypes based on the genetic profiles. Load the file *histology_subtype.mat* and you will see a vector $\mathbf{y} \in \mathbb{R}^{240}$. The i-th element being 1 implies the i-th patient has serous subtype ovarian cancer and being 2 represents endometrioid subtype ovarian cancer. Our goal is to better differentiate between patients with these two subtypes based on graph Fourier analysis.

3.1 Distinguishing power. For the specific graph-shift operator being the Laplacian $\mathbf{S} = \mathbf{L} = \mathbf{V}\Lambda\mathbf{V}^H$, we want to find the oscillation modes $\mathbf{v}_k$ such that the corresponding Graph Fourier Transform $\tilde{\mathbf{x}}(k)$ are the most different between patients with serous subtype and patients with endometrioid subtype. There are many ways to do this, and we consider the following simple heuristic. First compute the GFTs $\tilde{\mathbf{x}}_i = \mathbf{V}^H\mathbf{x}_i$ for all patients. Then for each k, define the distinguishing power of the oscillation mode $\mathbf{v}_k$ as

$$DP(\mathbf{v}_k) = \left| \frac{\sum_{i:y_i=1} \tilde{\mathbf{x}}_i(k)}{\sum_i \mathbf{1}\{y_i = 1\}} - \frac{\sum_{i:y_i=2} \tilde{\mathbf{x}}_i(k)}{\sum_i \mathbf{1}\{y_i = 2\}} \right| / \sum_i |\tilde{\mathbf{x}}_i(k)|, \tag{11}$$

where $\mathbf{1}$ is defined as the indicator function with

$$\mathbf{1}\{A\} = \begin{cases} 1, & \text{if } A \text{ is true;} \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

I.e., $DP(\mathbf{v}_k)$ computes the normalized difference between the mean GFT coefficient for $\mathbf{v}_k$ among patients with serous subtype and the mean GFT coefficient among patients with endometrioid subtype. Generate a plot of $DP(\mathbf{v}_k)$ versus k for all frequency indexes k.

3.2 Interpretation. To have a better sense about the distribution of distinguishing powers, generate a boxplot of $DP(\mathbf{v}_k)$ for all k using the command *boxplot*. In the boxplot, the central mark represents the median, the edges of the box are the 25th and 75th percentiles, and the whiskers extend to the most extreme data points that MATLAB considers not to be outliers. The data points MATLAB considers as outliers are plotted individually. Combining this analysis with the plot you generated, what can you say about the distribution of distinguishing power? Oscillation modes with high DP contain distinguishing features of the two subtypes and oscillation modes with low DP hardly contain useful information. In the following section, we will design a graph filter in order to improve classification accuracy of cancer subtypes. Following the study of ESE224 this semester, what's your best guess about the characteristics of reasonable graph filters?

4 Improving classifications using graph filter

We have applied k-Nearest Neighbors (k-NN) classifications a couple of times so far in the class and we will utilize it again in the cancer subtype classification. We will describe the particular k-NN algorithm we

are going to use in this problem and introduce a few vectorized operation functions in Section 4.1 to accelerate the running time for the k-NN algorithm. Then we will design a graph filter to improve classification accuracy for cancer subtypes in Section 4.2.

4.1 k-NN for leave-one-out cross validation

We design the following procedure to test the improvement of filtered graph signals $\mathbf{X}_{\text{f}}$ from the original graph signals $\mathbf{X}$. In short, we perform leave-one-out cross validation for a k nearest neighbors (k-NN) classifier. More precisely, we compute the pairwise distance between any pairs of patients using the graph signals $\mathbf{X}$ or the filtered graph signals $\mathbf{X}_{\text{f}}$. Then for a particular patient, we look at the k nearest patients as given by the distance being evaluated and assign to this patient the most common cancer histology among the k nearest patients. We then compare the assigned histology with her real cancer histology and evaluate the accuracy of the classifier. Finally, we repeat this process for the 240 women considered and obtain a global classification accuracy.

Write a function that given a matrix of graph signals (either the original ones or the filtered ones), the vector representing subtypes of patients $\mathbf{y}$, and the number of neighbors to be considered k, compute the global classification accuracy. Run this function for the original graph signals and $k = 3, 5, 7$, report your accuracies.

Here are some tips to write a fast vectorized leave-one-out cross validation algorithm, given arbitrary graph signals $\mathbf{Z}$. In computing the pairwise distance between pairs of patients, we can use the commands *d = squareform(pdist(Z))*. The function *pdist* outputs a vector containing the distances between each pair of observations in $\mathbf{Z}$ and *squareform* converts this vector into the matrix form.

For leave-one-out cross validation, given the matrix representing the pairwise distance, to select the nearest neighbors we can do the following. Start with the command *[∼, nn] = sort(d, 1, 'ascend')*, whose second output is a matrix with the i-th column representing the patient indexes ordered from the most similar patient of the i-th patient to the most dissimilar one. Since the distance from a patient to herself is always 0, the most closest patient to any patient is herself. Considering this fact, the command *nn_label = y(nn(2:(k + 1), :))* gives us the labels of the k nearest neighbors for all patients. In *nn_label*, the i-th column represents the labels of the k closest patient of the i-th patient. Finally, taking a *mode* with the right dimension *mode(nn_label, 1)* yields the prediction results from the k-NN classifier. Comparing the results with the actual label $\mathbf{y}$ gives the accuracy.

4.2 Graph filters

We consider the following two graph filters to improve the classification accuracy results we computed in the previous section. The first graph filter keeps only the information conveyed in the oscillation mode that is most distinguishable for two subtypes. To be more specific, for the genetic profile $\mathbf{x}_i$ of each patient, its GFT $\tilde{\mathbf{x}}_i$ is fed into a graph filter H_1 with the following frequency response,

$$H_1(k) = \begin{cases} 1, & \text{if } k = \text{argmax}_k \, DP(\mathbf{v}_k); \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Denote the filtered GFT as $\hat{\tilde{\mathbf{x}}}_i$ and its inverse GFT as $\hat{\mathbf{x}}_i$. We can then form a filtered graph signal matrix with each of its row representing the filtered genetic profile for the i-th patient. Run the k-NN classifier you wrote in 4.1 with this filtered graph signal matrix and report your accuracies for $k = 3, 5, 7$. Compare the results with those obtained for the original graph signals. What's your observation?

We can consider a more general graph filter H_p with the following frequency response,

$$H_p(k) = \begin{cases} 1, & \text{if } DP(\mathbf{v}_k) \geq p\text{-th percentile of the distribution of } DP; \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

where p is any real number between 0 and 100. In words, this family of graph filters keep the information conveyed in oscillation modes that are distinguishable for two subtypes to some extent. The p here is chosen to select the number of oscillation modes. One common choice of p is 50, and the corresponding graph filter keeps all information conveyed in the 50% of the oscillation modes that have more distinguishing power and removes information conveyed in the remaining 50% modes that cannot differentiate two subtypes very well. Run the k-NN classifier you wrote in 4.1 with the graph signal fed into this graph filter with your choice of p (this choice is very robust so start with some guessing) such that the classification error is much smaller than the classification error using the original graph signals. Report your accuracies for $k = 3, 5, 7$ and at least 5 different p of your choices for a wide range.