Discrete Signal Processing on Graphs

Group 33
Tanniru Abhinav Siddharth
2020112007
Vayur Shanbhag
2020112027

1 Abstract

We have access to large amounts of data everyday, our challenge is to create useful information from various types of measurements or data. The data indexed by the nodes of a graph gives us a signal. Discrete signal processing provides us an effective and efficient method to analyse and transform various well ordered time or image signals.

We intend to extend this to DSP on graphs by drawing parallels from already known concepts like filters, spectral representation, convolution, Fourier transform, impulse response and frequency response. We will also illustrate an example on data compression.

2 Signals on Graphs

We refer to data indexed by nodes of a graph as a graph signal and our approach as DSP_G. We represent a graph $G = (V, A)$, where node set $V = \{0, \cdots, N-1\}$ and A is an weighted adjacency matrix of the graph, denotes a directed edge of weight A_{ij} going from v_j to v_i. ($A_{ij} \in \mathbb{C}$)

We define a graph signal as a map from the set of nodes into the set of complex numbers:

$$s : v \to \mathbb{C}$$

$$v_n \mapsto s_n \tag{1}$$

For simplicity we can write the graph signals as vectors, $s = (s_0 \cdots s_{N-1})^T$. The set of nodes connected to v_n is called the neighborhood of v_n and denoted by: $N_n = \{m : A_{n,m} \neq 0\}$.

We will be dealing with directed graphs, hence the graph Laplacian definition can not be applied here.

2.1 Graph Shift

In classic DSP, we represent signals and build filters as polynomials of z^{-1}, we have the shift/delay filter which delays the input signal by one sample.

$$h_{\text{shift}}(z) = z^{-1}$$

Also, in DSP, series combination of filters is commutative, filtering a delayed signal and delaying a filtered signal must be the same:

$$z^{-1}.h(z) = h(z).z^{-1}$$

We now extend these concepts to DSP$_G$, we call the the delay operation as the graph shift as replacing the sample with weighted linear combination of the sample signals of the neighbours.

$$s_o = \sum_{m=0}^{N-1} A_{n,m} s_m = \sum_{m \in N_n} A_{n,m} s_m \tag{2}$$

2.2 Graph filters

Any system $H \in \mathbb{C}^{N \times N}$, a graph filter, for an input s producing output Hs represents a linear system.

Graph filters must have the following properties as discussed in the previous section:

- Linearity: $H(as_1 + bs_2) = aHs_1 + bHs_2$
- Shift Invariance: $A(Hs) = H(As)$

If A is the graph adjacency matrix, then a graph filter H is linear and shift-invariant **iff** H is a polynomial in the graph shift A:

$$H = h(A) = h_0 I + h_1 A + \cdots + h_L A^L \tag{3}$$

$$h(x) = h_0 + h_1 x + \cdots + h_L x^L \quad h_l \in \mathbb{C}$$

2.3 Properties of Graph Filters

See references for detailed proofs. [1]

- All linear, shift-invariant filters on a graph $G = (V, A)$ form a vector space:

$$F = \left\{ H : H = \sum_{l=0}^{N_A - 1} h_l A^l \quad h_l \in \mathbb{C} \right\} \tag{4}$$

- A graph filter $H = h(A) \in F$ is invertible **iff** polynomial $h(x)$ satisfies $h(\lambda_m) \neq 0$ for all distinct eigenvalues $\lambda_1, \cdots, \lambda_{M-1}$ of A.

  Then, there is a unique polynomial $g(x)$ of degree $deg(g(x)) < N_A$ that satisfies (N_A is the degree of minimal polynomial of $m_A(x)$):

$$h(A)^{-1} = g(A) \in F \tag{5}$$

- The filter taps (or coefficients) $h_0, \cdots, h_{N_A} - 1$ of the filter $h(A)$ uniquely determine its impulse response u (output for the unit impulse input $\delta = (1, 0, \cdots, 0)^T$).

  Conversely, the impulse response uniquely determines the filter taps, provided $rank\hat{A} = N_A$, $\hat{A} = (A^0\delta, \cdots, A^{N_A - 1}\delta)$.

3 Fourier Transform on Graphs

3.1 Spectral Decomposition

Spectral decomposition refers to the identification of subspaces $S_0, \cdots, S_{K-1}$ of signal space S that are invariant to filtering, so that for any $s_k \in S_k$ and filter $h(A) \in F$, the output $\hat{(s)}_k = h(A)s_k$ lies in the same subspace as S_k. Then, $s \in S$ can be represented as:

$$s = s_0 + s_1 + \cdots + s_{K-1} \qquad s_k \in S_k \tag{6}$$

The above decomposition can be uniquely determined **iff**:

- $S_k \cap S_m = \phi \qquad k \neq m.$
- $\dim S_0 + \cdots + \dim S_{K-1} = N.$
- Each S_k can't be decomposed into smaller subspaces. Hence, S can be written as:

$$S = S_0 \oplus S_1 \oplus \cdots \oplus S_{K-1}$$

We can write $A = VJV^{-1}$, as A need not always be diagonizable, J is the Jordan normal form and V is the matrix of generalised eigenvectors. Let $S_{m,d} = span\{v_{m,d,0}, \cdots, v_{m,d,R_{m,d}-1}\}$be a vector subspace of S spanned by the d^{th} Jordan chain of λ_m. Then the spectral decomposition of the signal space S, is given by:

$$S = \bigoplus_{m=1}^{M-1} \bigoplus_{d=0}^{D_M - 1} S_{m,d} \tag{7}$$

3.2 Graph Fourier Transform

We can get s from the generalised eigenvector bases of the subspaces $S_{m,d}$, using the generalised eigenvector basis V:

$$s = V\hat{s} \tag{8}$$

The vector of expansion coefficients is given by:

$$\hat{s} = V^{-1}s \tag{9}$$

This equation is the graph Fourier transform and the union of the bases of all the spectral components $S_{m,d}$ is called the graph Fourier basis. Hence, equation (8) is called the inverse graph Fourier transform.

The Fourier transform matrix is given by:

$$F = V^{-1} \tag{10}$$

Similar to DSP, in DSP_G, we call the coefficients $\hat{s}_n$ as the spectrum of the input signal s.

3.3 Convolution

h(J) represents the frequency response of any given filter h(A). Hence,

$$\begin{aligned} s_F &= h(A)s \\ &= F^{-1}h(J)Fs \\ &= F^{-1}h(J)\hat{s} \end{aligned}$$

$$\Rightarrow Fs_F = h(J)\hat{s} \tag{11}$$

In other words, the above equation says the product of frequecy response of a filter and the spectrum of a signal is equal to the graph Fourier transform of the filtered signal.

Note the similarity to the convolution theorem from classical DSP.

3.4 Graphs with diagonizable adjacency matrix

Since diagonizable adjacency matrices would have N linearly independent eigenvectors, we would be able to simplify the frequecy response h(J) of a filter h(A) to $h(\lambda_m)$, λ_m are the eigenvalues of A.

Graphs with symmetric adjacency matrices, undirected graphs, are always diagonizable, they are also orthogonal ($X.X^{-1} = I$), hence $F^{-1} = F^H$, this assists us while dealing with real world problems.

3.5 Correspondence of DSP and DSP on graphs

A finite discrete time series we use in DSP can be represented using a graph as:

It would have the adjacency matrix as:

$$A = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 \end{bmatrix}$$

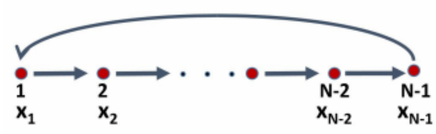


Figure 1: Time series

Hence, time-shift can be written as

$$s_0 = As$$

It's eigen decomposition(Jordan decomposition) is, A= C_N:

$$C_N = \frac{1}{N}\mathrm{DFT}_N^{-1}\begin{pmatrix} e^{-j\frac{2\pi.0}{N}} & & \\ & \ddots & \\ & & e^{-j\frac{2\pi.(N-1)}{N}} \end{pmatrix}\mathrm{DFT}_N$$

where DFT_N is the discrete Fourier transform matrix. Hence, the graph Fourier transform F= DFT_N.

For any general filter, $h(C_N) = \sum_{l=0}^{N-1} h_l C_N^l$, the coefficients of the output $\hat{s} = h(C_N)s$ are:

$$\begin{aligned} \hat{s} &= h_n s_0 + \cdots + h_0 s_n + h_{N-1}s_{n+1} + \cdots + h_{N+1}s_{N-1} \\ &= \sum_{k=0}^{N-1} s_k h_{n-k \, mod \, N} \end{aligned}$$

which is just circular convolution.

Graphs similar to the above figure but with undirected edges and different boundary conditions give various discrete cosine and sine transforms which are extensively used in signal processing and data compression which have been fundamental to the digital revolution.

4 Applications

- Linear Prediction:

  Here, we can approximate the signal based on the previous values it has taken with the help of a adaptive prediction filter. A residual is created using the forward prediction filter, and then the signal is synthesized using the backward synthesis filter.

  In a graph signal, we can approximate the signal based on its neighbors and obtain a residual signal. Then we can synthesize the signal with an L-tap filter to obtain the approximated signal.

- Data Classification:

  Data classification is mainly used in machine learning, but this can also be represented as a filter on a graph. Given a directed graph with some nodes "known" or classified, we can then follow the edges of the graph to classify all the "unknown" nodes, and hence classify the data.

- Customer behavior prediction:

  Using the prediction filter, we can approximate the customer pattern in the future and hence predict their behavior in the future. To calculate the behavior in the Kth month, we look at the data of the past K-1 months and design a predictive filter. Hence, we can predict the behavior. As K increases, the accuracy of the model will also increase.

5 Application of interest

5.1 Objective

There are a few applications of graph signal processing, and we will be focusing on signal compression. Our goal is to approximate the given signal such that less space is required for storage and transmission. We also intend to find the error in the compression scheme as we take more and more space to store the signal.

5.2 Theory

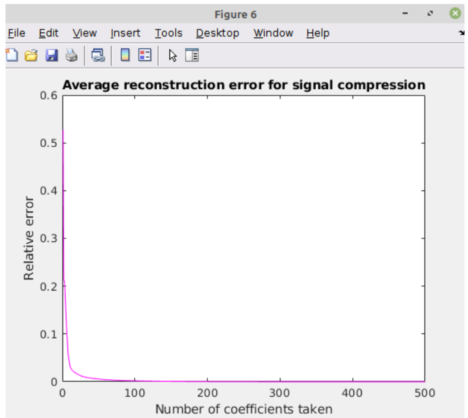
Compression is done by first representing the signal in an orthonormal basis. An assumption is made that only a few eigenvectors that have the highest eigenvalues contains most of the information. We then only use and store and use these few values. Image compression standards like JPEG and JPEG 2000 also use such methods.

Specifically in Graph signals, we know that graph signals are shift invariant. hence we can represent As = s (approximately). This implies that we can write s as sum of eigenvectors of A. Take only the C eigenvectors with the largest eigenvalues, we get

$$S_t = F^{-)}[S_0 + S_1 + ... + S_{C-1} + 0 + 0...]$$

5.3 Implementation

We do this by first expanding the signal into an orthonormal basis (graph fourier transform) and then taking the assumption that most of the signal will be present in only some spectral coefficients. Our matrix A can be approximated based on the C largest spectral coeffients. For this we use the "gspbox" toolbox to compute the graph fourier transform and the inverse graph fourier transform. We first take an arbitrary graph signal and then compute its dft. then we just consider the C largest coefficients and make the rest go to zero and calculate



the igft of the new fourier coefficients. Then we calculate the relative error for different C and plot it.

6 Conclusion

Graphs have wide range of uses, from social networks, machine learning, sensor nets or image processing, the above concepts allow us to extract useful information from datasets.

We have developed a new DSP_G for data represented by graphs. We have seen how DSP on graphs extends natural concepts such as filters, spectral representation, Fourier transform and frequency response.

Finally, we have seen specific applications of graph signal processing methods. We looked in detail into the application of signal processing and successfully compressed a given signal and then reconstructed. We also found out that the relative error upon reconstruction decreases as we increase number of eigenvectors in our representation of the signal

References

[1] Sandryhaila, Aliaksei, and José MF Moura. "Discrete signal processing on graphs." IEEE transactions on signal processing 61.7 (2013): 1644-1656.

[2] Ortega, Antonio, et al. "Graph signal processing: Overview, challenges, and applications." Proceedings of the IEEE 106.5 (2018): 808-828.

[3] Stanković, Ljubiša, Miloš Daković, and Ervin Sejdić. "Introduction to graph signal processing." Vertex-Frequency Analysis of Graph Signals. Springer, Cham, 2019. 3-108.

[4] Newman, Mark. Networks. Oxford university press, 2018.