Graph Signal Processing

Rahul Singh

DATA SCIENCE READING GROUP
IOWA STATE UNIVERSITY

March 24, 2017
Outline

1. Graph Signal Processing Background

2. Graph Signal Processing Frameworks
   - Laplacian Based
   - Discrete Signal Processing on Graphs ($\text{DSP}_G$) Framework
     - Weight Matrix Based
     - Directed Laplacian Based

3. Spectral Graph Wavelets

4. Conclusions
Classical Signal Processing

Structure behind time-series
Classical Signal Processing

Structure behind time-series

Structure behind image
Classical Signal Processing

- Translation, filtering, convolution, modulation, Fourier transform, wavelets, sparse representations . . .

Structure behind time-series

Structure behind image
Graph Signal Processing
Graph Signal Processing

**Background**

**Introduction**

Graph Signal Processing

- Data Science Reading Group, ISU

March 24, 2017
Graph Signal Processing

- A graph signal
Graph Signal Processing Applications

Social Network

Power Grid Network

Bangalore Road Network

Biological Network
Difficulty in GSP

- Translation is simple in classical signal processing

What does it mean to translate the signal to 'vertex 50'?
Difficulty in GSP

- Translation is simple in classical signal processing

- What does it mean to translate the signal to ‘vertex 50’?

- Challenging in GSP
Background  

Notation

A graph signal \( f \)

Graph \( G = (V, W) \)

Weight matrix \( W = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \)

Degree matrix \( D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} \)

Laplacian matrix

\[
L = D - W = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{bmatrix}
\]
### Existing Graph Signal Processing (GSP) Frameworks

<table>
<thead>
<tr>
<th>Shift Operator</th>
<th>GSP based on Laplacian(^1)</th>
<th>Discrete Signal Processing on Graphs (DSP(_G)) Framework(^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Not defined</td>
<td>The weight matrix</td>
</tr>
<tr>
<td>LSI Filters</td>
<td>Not applicable</td>
<td>Applicable</td>
</tr>
<tr>
<td>Applicability</td>
<td>Only undirected graphs</td>
<td>Directed graphs</td>
</tr>
<tr>
<td>Frequencies</td>
<td>Eigenvalues of the Laplacian matrix</td>
<td>Eigenvalues of the weight matrix</td>
</tr>
<tr>
<td>Harmonics</td>
<td>Eigenvectors of the Laplacian matrix</td>
<td>Eigenvectors of the weight matrix</td>
</tr>
<tr>
<td>Frequency Ordering</td>
<td>Laplacian quadratic form</td>
<td>Total variation</td>
</tr>
<tr>
<td>Multiscale Analysis</td>
<td>Exists (SGWT)</td>
<td>Does not exist</td>
</tr>
</tbody>
</table>


# Classical vs Graph Signal Processing (Laplacian Based)

<table>
<thead>
<tr>
<th>Operator/ Transform</th>
<th>Classical Signal Processing</th>
<th>Graph Signal Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fourier Transform</strong></td>
<td>$\hat{x}(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt$</td>
<td>$\hat{f}(\lambda_\ell) = \sum_{n=1}^{N} f(n)u_\ell^*(n)$</td>
</tr>
<tr>
<td></td>
<td>Frequency: $\omega$ can take any value</td>
<td>Frequency: Eigenvalues of the graph Laplacian ($\lambda_\ell$)</td>
</tr>
<tr>
<td></td>
<td>Fourier basis: Complex exponentials $e^{j\omega t}$</td>
<td>Fourier basis: Eigenvectors of the graph Laplacian ($u_\ell$)</td>
</tr>
<tr>
<td><strong>Convolution</strong></td>
<td>In time domain: $x(t) \ast y(t) = \int_{-\infty}^{\infty} x(\tau)y(t-\tau) d\tau$</td>
<td>Defined through Graph Fourier Transform</td>
</tr>
<tr>
<td></td>
<td>In frequency domain: $x(t) \ast y(t) = \hat{x}(\omega)\hat{y}(\omega)$</td>
<td>$\hat{f} \ast \hat{g} = U(\hat{f} \cdot \hat{g})$</td>
</tr>
<tr>
<td><strong>Translation</strong></td>
<td>Can be defined using convolution</td>
<td>Defined through graph convolution</td>
</tr>
<tr>
<td></td>
<td>$T_\tau x(t) = x(t - \tau) = x(t) \ast \delta_\tau(t)$</td>
<td>$T_i f(n) = \sqrt{N}(f \ast \delta_i)(n)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$= \sqrt{N} \sum_{\ell=0}^{N-1} \hat{f}(\lambda_\ell)u^*<em>\ell(i)u</em>\ell(n)$</td>
</tr>
<tr>
<td><strong>Modulation</strong></td>
<td>Multiplication with the complex exponential</td>
<td>Multiplication with the eigenvector of the graph Laplacian</td>
</tr>
<tr>
<td></td>
<td>$M_\omega x(t) = e^{j\omega t}x(t)$</td>
<td>$M_k f(n) = \sqrt{N}u_k(n)f(n)$</td>
</tr>
</tbody>
</table>
Graph Fourier Transform

- Analogy from classical signal processing
  - Classical Fourier basis: Complex exponentials
  - Complex exponentials are **Eigenfunctions** of 1-D Laplacian operator $\Delta$,

\[-\Delta(e^{j\omega t}) = -\frac{\partial^2}{\partial t^2} e^{j\omega t} = (\omega)^2 e^{j\omega t}.\]
Graph Fourier Transform

- Analogy from classical signal processing
  - Classical Fourier basis: Complex exponentials
  - Complex exponentials are **Eigenfunctions** of 1-D Laplacian operator $\Delta$,
    \[
    -\Delta(e^{j\omega t}) = -\frac{\partial^2}{\partial t^2} e^{j\omega t} = (\omega)^2 e^{j\omega t}.
    \]

- Graph Fourier Transform
  - Graph Fourier basis are **Eigenfunctions** of the Laplacian matrix (operator)
  - Graph Frequencies: Eigenvalues of the Laplacian matrix $L$
  - Graph Harmonics: Eigenvectors of the Laplacian matrix $L$

\[
L = U\Lambda U^T
\]

- GFT $\hat{f} = U^T f$, IGFT $f = U\hat{f}$
Graph Signal in Two Domains

A graph signal in vertex domain and spectral domain.
Laplacian Eigenvectors as GFT Basis
Graph Convolution

Laplacian Based

\[ L = \begin{bmatrix}
2 & -1 & -1 & 0 & 0 \\
-1 & 3 & -1 & -1 & 0 \\
-1 & -1 & 3 & -1 & 0 \\
0 & -1 & -1 & 3 & -1 \\
0 & 0 & 0 & 0 & -1 & 1
\end{bmatrix} \]

Eigenvalues: 0, 0.8299, 2.6889, 4, 4.4812

\[ U = \begin{bmatrix}
0.4472 & 0.4375 & 0.7031 & 0 & 0.3380 \\
0.4472 & 0.2560 & -0.2422 & 0.7071 & -0.4193 \\
0.4472 & 0.2560 & -0.2422 & -0.7071 & -0.4193 \\
0.4472 & -0.1380 & -0.5362 & 0 & 0.7024 \\
0.4472 & -0.8115 & 0.3175 & 0 & -0.2018
\end{bmatrix} \]
Graph Convolution (cont’d. . . )

\[ f = [3, 4, 6, 3, 1]^T \]

\[ g = [4, 2, 4, 2, 2]^T \]

\[ h = f \ast g = \text{IGFT}(\hat{f} \cdot \hat{g}) \]

\[ h = [21.92, 23.92, 21.08, 21.72, 17.80]^T \]
Graph Translation

Translation to node $i$:

$$T_i(f) = \sqrt{N} (f \ast \delta_i) = \sqrt{N} \text{IGFT}(\hat{f} \cdot U^T(:, i))$$

**Examples:**

$$f = [3, 4, 6, 3, 1]^T$$

$$T_1 f = [2.44, 5.08, 5.08, 3.72, 0.69]^T$$

$$T_2 f = [5.08, 1.50, 4.66, 3.56, 2.21]^T$$

$$T_3 f = [3.72, 3.56, 3.56, 1.08, 5.08]^T$$
Graph Harmonics are eigenfunctions of LSI filters. Total Variation is used for frequency ordering.

Concepts in the DSP\textsubscript{G} framework
DSP\textsubscript{G} Framework (Cont’d...)

- **Shift operator**
  - Weight matrix \( W \) of the graph

- **Shifted graph signal** \( \tilde{f} = Wf \)

- **Example:** shifting discrete-time signal (one unit right)

\[
\begin{align*}
x & = [9, 7, 5, 0, 6]^T \\
\tilde{x} & = Wx = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 9 \\ 7 \\ 5 \\ 0 \\ 6 \end{bmatrix} = \begin{bmatrix} 6 \\ 9 \\ 7 \\ 5 \\ 0 \end{bmatrix}
\end{align*}
\]

- **Linear Shift Invariant (LSI) filters**
  - \( H(Wf_{in}) = W(Hf_{in}) \)
  - Polynomials in \( W \)

\[
H = h(W) = \sum_{m=0}^{M-1} h_m W^m
\]

\[
H = h_0 I + h_1 W + \ldots + h_{M-1} W^{M-1}
\]
DSPG Framework (Cont’d...)

- Analogy from classical signal processing
  - Classical Fourier basis: Complex exponentials
  - Complex exponentials are **Eigenfunctions** of Linear Time Invariant (LTI) filters

- Graph Fourier Transform
  - Graph Fourier basis are **Eigenfunctions** of Linear Shift Invariant (LSI) graph filters
  - Graph Frequencies: Eigenvalues of the weight matrix $W$
  - Graph Harmonics: Eigenvectors of the weight matrix $W$
  - $W = V\Sigma V^{-1}$
  - GFT $\hat{f} = V^{-1}f$, IGFT $f = V^{-1}\hat{f}$
**DSP\(_G\) Framework (Cont’d.)**

- **Total Variation in classical signal processing**
  
  \[
  \text{TV}(x) = \sum_n x[n] - x[n - 1] = \|x - \tilde{x}\|_1, \text{ where } \tilde{x}[n] = x[n - 1]
  \]

- **Analogy from classical signal processing**

- **Total Variation on graphs**
  
  \[
  \text{TV}_G(f) = \|f - \tilde{f}\|_1 = \|f - Wf\|_1
  \]

- **Frequency ordering:** Based on Total Variation

- **Eigenvalue with largest magnitude:** Lowest frequency
Problems in Weight Matrix based DSP<sub>G</sub>

- Weight matrix based DSP<sub>G</sub>
  - Does not provide “natural” frequency ordering
  - Even a constant signal has high frequency components

Constant graph signal

Graph frequencies:
-1.62, -1.47, -0.46, 0.62, 2.94

\[
\hat{f} = \begin{bmatrix}
0 \\
0.36 \\
0.16 \\
0 \\
2.20
\end{bmatrix}
\]
Graph Fourier Transform based on Directed Laplacian
Graph Fourier Transform based on Directed Laplacian

- Redefines Graph Fourier Transform under $\text{DSP}_G$
  - Shift operator: Derived from directed Laplacian
  - Linear Shift Invariant filters: Polynomials in the directed Laplacian
  - Graph frequencies: Eigenvalues of the directed Laplacian
  - Graph harmonics: Eigenvectors of the directed Laplacian

- “Natural” frequency ordering

- Better intuition of frequency as compared to the weight matrix based approach

- Coincides with the Laplacian based approach for undirected graphs

---

Rahul Singh, Abhishek Chakraborty, and BS Manoj. “Graph Fourier transform based on directed Laplacian”. In: Signal Processing and Communications (SPCOM), 2016 International Conference on. IEEE. 2016, pp. 1–5.
Directed Laplacian Matrix

- Basic matrices of a directed graph
  - Weight matrix: $W$
    - $w_{ij}$ is the weight of the directed edge from node $j$ to node $i$
  - In-degree matrix: $D_{in} = \text{diag}(\{d_{in}^i\}_{i=1,2,\ldots,N})$, $d_{in}^i = \sum_{j=1}^{N} w_{ij}$
  - Out-degree matrix: $D_{out} = \text{diag}(\{d_{out}^i\}_{i=1,2,\ldots,N})$, $d_{out}^i = \sum_{i=1}^{N} w_{ij}$

- Directed Laplacian matrix $L = D_{in} - W$
  - Sum of each row is zero
  - $\lambda = 0$ is surely an eigenvalue

A directed graph

\[
W = \begin{bmatrix}
0 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
\end{bmatrix}
\quad
D_{in} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 3 & 0 & 0 \\
0 & 0 & 2 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\quad
L = \begin{bmatrix}
1 & 0 & 0 & -1 \\
-1 & 3 & -1 & -1 \\
-1 & -1 & 2 & 0 \\
-1 & 0 & 0 & 1 \\
\end{bmatrix}
\]

Weight matrix \quad In-degree matrix \quad Directed Laplacian matrix
Graph Fourier Transform based on Directed Laplacian Shift Operator

Shift Operator (Proposed)

A directed cyclic (ring) graph

L =
\[
\begin{bmatrix}
1 & 0 & 0 & 0 & -1 \\
-1 & 1 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & -1 & 1 \\
\end{bmatrix}
\]

A signal \( x = [9 7 1 0 6]^T \); shifted by one unit to the right \( \tilde{x} = [6 9 7 1 0]^T \)

\[
\tilde{x} = Sx = (I - L)x = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
\end{bmatrix} \begin{bmatrix}
9 \\
7 \\
1 \\
0 \\
6 \\
\end{bmatrix} = \begin{bmatrix}
6 \\
9 \\
7 \\
1 \\
0 \\
\end{bmatrix}
\]

\( S = (I - L) \) is the shift operator

Shifted graph signal: \( \tilde{f} = Sf = (I - L)f \)
A linear graph filter

A graph filter $H$ is LSI if the following conditions are satisfied.

1. Geometric multiplicity of each distinct eigenvalue of the graph Laplacian is one.

2. The graph filter $H$ is a polynomial in $L$, i.e., if $H$ can be written as

$$H = h(L) = h_0 I + h_1 L + \ldots + h_m L^m$$

where, $h_0, h_1, \ldots, h_m \in \mathbb{C}$ are called filter taps.
Graph Fourier Transform based on Directed Laplacian

- Jordan decomposition of the directed Laplacian: \( L = V JV^{-1} \)
- **Graph Fourier basis**: Columns of \( V \) (Jordan Eigenvectors of \( L \))
- **Graph frequencies**: Eigenvalues of \( L \) (diagonal entries of Jordan blocks in \( J \))

GFT: \( \hat{f} = V^{-1} f \) and IGFT: \( f = V \hat{f} \)

**Frequency Ordering**: based on Total Variation

- Total Variation: \( TV_G(f) = \| f - Sf \|_1 = \| f - (I - L)f \|_1 \)

\[ TV_G(f) = \| Lf \|_1 \]

**Theorem**

\( TV \) of an eigenvector \( v_r \) is proportional to the absolute value of the corresponding eigenvalue

\[ TV(v_r) \propto |\lambda_r| \]
Frequency Ordering

Arbitrary graph

Graph with positive edge weights

Undirected graph with real edge weights.

Undirected graph with real and non-negative edge weights.
Example

- Graph signal $\mathbf{f} = [0.1189 \ 0.3801 \ 0.8128 \ 0.2441 \ 0.8844]^T$ defined on the directed graph

A weighted directed graph

Spectrum of the signal $\mathbf{f} = [0.1189 \ 0.3801 \ 0.8128 \ 0.2441 \ 0.8844]^T$
Example

- Graph signal $\mathbf{f} = [0.1189 \ 0.3801 \ 0.8128 \ 0.2441 \ 0.8844]^T$ defined on the directed graph

A weighted directed graph

Spectrum of the signal $\mathbf{f} = [0.1189 \ 0.3801 \ 0.8128 \ 0.2441 \ 0.8844]^T$
Example: Zero Frequency

- Eigenvector corresponding to $\lambda_0$ is $v_0 = \frac{1}{\sqrt{N}}[1, 1, \ldots, 1]^T$
- TV of $v_0$ is zero
- For a constant graph signal $f = [k, k, \ldots]^T$, GFT is $\hat{f} = [(k\sqrt{N}), 0, \ldots]^T$
- Only zero frequency component

A weighted directed graph

Spectrum of the constant signal $f = [1 1 1 1 1]^T$
Example: Zero Frequency

- Eigenvector corresponding to $\lambda_0$ is $v_0 = \frac{1}{\sqrt{N}}[1, 1, \ldots, 1]^T$
- TV of $v_0$ is zero
- For a constant graph signal $f = [k, k, \ldots]^T$, GFT is $\hat{f} = [(k\sqrt{N}), 0, \ldots]^T$
- Only zero frequency component

The weight matrix based approach of GFT fails to give this basic intuition

A weighted directed graph

Spectrum of the constant signal $f = [1 \ 1 \ 1 \ 1 \ 1]^T$
## Comparison of the GSP Frameworks

<table>
<thead>
<tr>
<th></th>
<th>GSP based on Laplacian</th>
<th>DSP\textsubscript{G} Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GSP based on Laplacian</td>
<td>DSP\textsubscript{G} Framework</td>
</tr>
<tr>
<td></td>
<td>Based on Weight Matrix</td>
<td>Based on Directed Laplacian</td>
</tr>
<tr>
<td><strong>Shift Operator</strong></td>
<td>Not defined</td>
<td>Derived from directed Laplacian ((\mathbf{I} - \mathbf{L}))</td>
</tr>
<tr>
<td><strong>LSI Filters</strong></td>
<td>Not applicable</td>
<td>Applicable</td>
</tr>
<tr>
<td></td>
<td>Only undirected graphs</td>
<td>Directed graphs</td>
</tr>
<tr>
<td></td>
<td>Directed graphs</td>
<td>Directed graphs</td>
</tr>
<tr>
<td><strong>Frequencies</strong></td>
<td>Eigenvalues of the Laplacian (real)</td>
<td>Eigenvalues of the weight matrix</td>
</tr>
<tr>
<td></td>
<td>Eigenvalues of the Laplacian (real)</td>
<td>Eigenvalues of the directed Laplacian</td>
</tr>
<tr>
<td><strong>Harmonics</strong></td>
<td>Eigenvectors of the Laplacian matrix (real)</td>
<td>Eigenvectors of the weight matrix</td>
</tr>
<tr>
<td></td>
<td>Eigenvectors of the Laplacian matrix (real)</td>
<td>Eigenvectors of the directed Laplacian</td>
</tr>
<tr>
<td><strong>Frequency Ordering</strong></td>
<td>Laplacian quadratic form (natural)</td>
<td>Total variation (not natural)</td>
</tr>
<tr>
<td></td>
<td>Total variation (not natural)</td>
<td>Total variation (natural)</td>
</tr>
<tr>
<td><strong>Multiscale Analysis</strong></td>
<td>Exists (Spectral Graph Wavelet Transform)</td>
<td>Does not exist</td>
</tr>
<tr>
<td></td>
<td>Does not exist</td>
<td>Possible</td>
</tr>
</tbody>
</table>
Spectral Graph Wavelet Transform

- Classical wavelets
  - Wavelets at different scales and locations are constructed by scaling and translating a single "mother" wavelet $\psi$
    \[
    \psi_{s,a}(x) = \frac{1}{s} \psi \left( \frac{x-a}{s} \right)
    \]
  - Scaling in Fourier domain
    \[
    \psi_{s,a}(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{j\omega x} \hat{\psi}(s\omega) e^{-j\omega a} d\omega
    \]
  - Scaling $\psi$ by $1/s$ corresponds to scaling $\hat{\psi}$ with $s$
  - Term $e^{-j\omega a}$ comes from localization of the wavelet at location $a$

- Spectral Graph Wavelets
  - Graph wavelet at scale $t$ and centered at node $n$
    \[
    \psi_{t,n}(m) = \sum_{\ell=0}^{N-1} u_\ell(m) g(t\lambda_\ell) u^*_\ell(n)
    \]
  - Frequency $\omega$ is replaced with eigenvalues of graph Laplacian $\lambda_\ell$
  - Translating to node $n$ corresponds to multiplication by $u^*_\ell(n)$
  - $g$ acts as a scaled bandpass filter, replacing $\hat{\psi}$
Wavelet Generating Kernel

\[ \psi_{t,n}(m) = \sum_{\ell=0}^{N-1} u_{\ell}(m) g(t\lambda_{\ell}) u^{\ast}_{\ell}(n) \]

- \( g \) acts as a scaled bandpass filter: \textbf{wavelet generating kernel}

\[
g(x; \alpha, \beta, x_1, x_2) = \begin{cases} 
  x_1^{-\alpha}x^{\alpha} & \text{for } x < x_1 \\
  p(x) & \text{for } x_1 \leq x \leq x_1 \\
  x_2^{\beta}x^{-\beta} & \text{for } x > x_2, 
\end{cases}
\]

- \( x \) is the distance from origin (zero frequency)
- \( \alpha \) and \( \beta \) are integer parameters of \( g \)
- \( x_1 \) and \( x_2 \) determine transition regions
- \( p(x) \) is a cubic spline that ensures continuity in \( g \)

- A possible choice of these parameters: \( \alpha = \beta = 2, \ x_1 = 1, \ x_2 = 2, \) and \( p(x) = -5 + 11x - 6x^2 + x^3 \)
Wavelet Generating Kernel cont’d...

\[ \psi_{t,n}(m) = \sum_{\ell=0}^{N-1} u_{\ell}(m) g(t\lambda_{\ell}) u_{\ell}^*(n) \]

- Scale \( t \) is a continuous variable

- For practical purposes, choose \( J \) number of logarithmically equally spaced scales \( t_1, \ldots, t_J \)
  - \( t_{\text{min}} = |\lambda_{\text{max}}|/K, \lambda_{\text{max}} \) is the eigenvalue of \( L \) with largest magnitude and \( K \) is a design parameter
  - \( t_J = x_2/|\lambda_{\text{max}}| \) and \( t_1 = x_2/t_{\text{min}} \)
Wavelet Generating Kernel cont’d...

- $|\lambda_{max}| = 10$, $\alpha = \beta = 2$, $x_1 = 1$, $x_2 = 2$, $K = 20$, and $J = 4$

- Kernel at $t_1 = 4.0000$.

- Kernel at $t_2 = 1.4736$.

- Kernel at $t_3 = 0.5429$.

- Kernel at $t_4 = 0.2$.

- As $t$ increases, the kernel becomes increasingly confined to low frequencies.
SGWT Examples

- Wavelets at different scales

- Large scale $\equiv$ Low frequency $\equiv$ Spread of the wavelet over the graph is high
SGWT Examples

- Wavelets at different scales

- Large scale $\equiv$ Low frequency $\equiv$ Spread of the wavelet over the graph is high
SGWT Examples

$\psi(t)$

$t = 5.0742$

t = 1.8693$

t = 0.6887$

t = 0.2537$
Matrix Form of SGWT

- Wavelet basis at scale $t = \text{collection of } N \text{ number of wavelets (each wavelet centered at a particular node of the graph)}$

\[
\Psi_t = [\psi_{t,1} | \psi_{t,2} | \ldots | \psi_{t,N}]
\]

\[
= U \begin{bmatrix}
  g(t\lambda_0) \\
  g(t\lambda_1) \\
  \vdots \\
  g(t\lambda_{N-1})
\end{bmatrix} U^T
\]

\[
= U G_t U^T
\]

Column of $U$ are eigenvectors of $L$

$g(t\lambda_\ell)$ is the sampled value of $g(t\lambda)$ at frequency $\lambda_\ell$

- Wavelet coefficient at scale $t$ and centered at node $n$ of a graph signal $f$

\[
W_f(t, n) = \langle \psi_{t,n}, f \rangle = \psi_{t,n}^T f
\]
Conclusions

- Introduction to Graph Signal Processing (GSP)
- Graph Signal Processing Frameworks
  - Laplacian Based
  - Discrete Signal Processing on Graphs ($\text{DSP}_G$) Framework
    - Weight Matrix Based
    - Directed Laplacian Based
- Spectral Graph Wavelet Transform (SGWT)
- Research opportunities are plenty
References


Thank You.

rasingh@iastate.edu