A Windowed Digraph Fourier Transform
Rasoul Shafipour1, Ali Khodabakhsh2, and Gonzalo Mateos1
1Dept. of Electrical and Computer Engineering, University of Rochester, Rochester, NY, USA
2Dept. of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA

Problem Statement
- Windowed (localized) graph signal $x(t)$ for node $i$: $x(t) := x \circ \phi_i = \text{diag}(x)\phi_i$, where $\phi_i \in \mathbb{R}^N$ is a windowing signal around $i \in V$.
- Akin to STFT, let the Windowed DGFT (WDGFT) be $X = \{x^{(1)}[i], \cdots, x^{(N)}[i]\} = U^T U \text{diag}(x) \Phi$, where $\Phi = [\phi_1, \cdots, \phi_N] = [\phi_i] \in \mathbb{R}^{N \times N}$.
- Q: How to learn $\Phi = [\phi_1, \cdots, \phi_N]$ capturing vertex-frequency energy content of graph signals?

Spectral vs. Spatial Resolution
Proposition 1: If $w_1 = 0$, then the optimal value of (P2) is zero and is achieved by $\tau_i = 0$, i.e., $f_i(0) = 0$.
$\Rightarrow w_1 = 0$ leads to constant all ones window $\Rightarrow$ WDGFT $\Rightarrow$ No resolution in the vertex domain.

Proposition 2: Let $w_1 < w_2$ be two parameters for penalizing the DC component of the window. If $\tau$ is a local minima for (P2) with $w_1$, then the corresponding optimization problem for $w_2$ has a local (or asymptotic) minima greater than $\tau$.
$\Rightarrow$ Tradeoff between smoothness and locality of windows $\Rightarrow w_1$ can be tuned to achieve a desired resolution.

Numerical Results
Figure 1: Undirected graph via stochastic block model ($N = 60$, $p_1 = 0.5$, $p_2 = 0.05$, and 3 communities)
- Run Dijkstra algorithm to find a proximity matrix $D$.
- Solve $N$ independent subproblems (P2); see Fig. 3.
- Construct signal $x$ in Fig. 1 by adding $u_{15}$ restricted to the first 20 nodes, $u_{20}$ restricted to the middle 20 nodes, and $u_{30}$ restricted to the last 20 nodes.
- Construct $x$ in the directed case, by adding $u_{10}$ restricted to 24 highly connected nodes and $u_{20}$ restricted to the rest.
- Obtain spectrograms $|X| = |U^T \text{diag}(x)\Phi|$.
- For the undirected graph, compare with [2].

Figure 2: Directed structural brain networks ($N = 47$ and 505 edges, among which 121 links are directed)
- Run Dijkstra algorithm to find a proximity matrix $D$.
- Solve $N$ independent subproblems (P2); see Fig. 3.
- Construct signal $x$ in Fig. 1 by adding $u_{15}$ restricted to the first 20 nodes, $u_{20}$ restricted to the middle 20 nodes, and $u_{30}$ restricted to the last 20 nodes.
- Construct $x$ in the directed case, by adding $u_{10}$ restricted to 24 highly connected nodes and $u_{20}$ restricted to the rest.
- Obtain spectrograms $|X| = |U^T \text{diag}(x)\Phi|$.
- For the undirected graph, compare with [2].

Figure 3: Examples of $f_i(\tau)$ in (P2) for different vertices.
- $f_i(\tau)$ is a local minima for $\tau_i$.
- Solve subproblems in parallel via gradient descent $\Rightarrow g = df_i/\partial \tau_i = -\{d_i \circ \phi_i\}^T U^T U \text{diag}(x)\phi_i$, where $W = \text{diag}(w_1, \cdots, w_N)$ and $d_i = D(:, i)$.
- Follow the update rule $\tau_i^{t+1} = \tau_i^{t} - \eta g(\tau_i)$ $\Rightarrow \{\tau_i\}$ converges to a stationary point of $f_i(\tau)$.

Figure 4: Spectrograms for both undirected and directed examples. (a) Proposed windowed digraph Fourier transform for the graph in Fig. 1 and a signal constructed by three different basis vectors using DGFT (b) Method in [2] for the same graph and a signal constructed by three different eigenvectors of the Laplacian matrix (c) Proposed method for the directed brain graph.

References

Contact Information
Web: www.ece.rochester.edu/~gmateosb/
Email: gmateosb@ece.rochester.edu